Another Look at XCB

Debrup Chakraborty, Vicente Hernandez-Jimenez, Palash Sarkar

CINVESTAV, Mexico City
and
Indian Statistical Institute, Kolkata

Asiacrypt 2013 Rump Session Presentation

IEEE Std 1619.2™-2010

IEEE Standard for Wide-Block Encryption for Shared Storage Media

Sponsor:

Information Assurance Standards Committee

and

Storage Systems Standards Committee

of the

IEEE Computer Society

Approved 30 September 2010

IEEE-SA Standards Board

Approved 5 May 2011

American National Standards Institute

Specifies

"encryption modes ... oriented toward random access storage devices"

IEEE Std 1619.2™-2010

Approval to set-up P1619.2: 02 November 2006.

- Initial group: 4 persons with Jim Hughes as the Chair.
- Final group: 30 members with Matthew V. Ball as the Chair.
- Important technical contributions: Hal Finney, Brian Gladman, Shai Halevi and David McGrew.
- Voting by 49 members of the individual balloting committee.

Approval of the standard: 30 September 2010.
Current status: Active Standard.
Available from:

- IEEE Explore Digital Library
- Purchase: $111 (pdf), $133 (print), $167 (pdf+print).

`https://standards.ieee.org/findstds/standard/1619.2-2010.html`

IEEE Std 1619.2^{TM}-2010

Specifies two *encryption* algorithms.

- EME2-AES.
- XCB-AES.

This talk is about XCB with an aside on EME2-AES.

XCB-AES Encryption in IEEE Std 1619.2^{TM}-2010

- $H \leftarrow$ AES-Enc$(K, 0^{128})$; obtain K_e, K_d and K_c;
- $A \leftarrow P[m-128 : m-1]$;
- **B** $\leftarrow P[0 : m-$ **127**$]$; $C \leftarrow$ AES-Enc(K_e, A); $D \leftarrow C \oplus h_1(H, Z, B)$;
- **E** $\leftarrow B \oplus c(K_c, D, \#B)$; $F \leftarrow D \oplus h_2(H, Z, E)$;
- **G** $\leftarrow$ AES-Dec(K_d, F);
- **CT** $\leftarrow E|G$.

- m: # msg bits; $\#B = \#E = m - 126$, $\#G = 128$ and so $\#CT = m + 2$.

XCB-AES Encryption in IEEE Std 1619.2^{TM}-2010

- $H \leftarrow$ AES-Enc$(K, 0^{128})$; obtain K_e, K_d and K_c;
- $A \leftarrow P[m-128 : m-1]$;
- **B** $\leftarrow P[0 : m-$ **127**$]$; $C \leftarrow$ AES-Enc(K_e, A); $D \leftarrow C \oplus h_1(H, Z, B)$;
- **E** $\leftarrow B \oplus c(K_c, D, \#B)$; $F \leftarrow D \oplus h_2(H, Z, E)$;
- **G** $\leftarrow$ AES-Dec(K_d, F);
- **CT** $\leftarrow E|G$.

- m: # msg bits; $\#B = \#E = m - 126$, $\#G = 128$ and so $\#CT = m + 2$.
- Decryption is not the inverse of encryption; each application of encryption or decryption increases length by 2 bits.

XCB-AES Encryption in IEEE Std 1619.2™-2010

- $H \leftarrow$ AES-Enc$(K, 0^{128})$; obtain K_e, K_d and K_c;
- $A \leftarrow P[m-128 : m-1]$;
- **B** $\leftarrow P[0 : m-$ **127**$]$; $C \leftarrow$ AES-Enc(K_e, A); $D \leftarrow C \oplus h_1(H, Z, B)$;
- **E** $\leftarrow B \oplus c(K_c, D, \#B)$; $F \leftarrow D \oplus h_2(H, Z, E)$;
- **G** $\leftarrow$ AES-Dec(K_d, F);
- **CT** $\leftarrow E|G$.

- m: # msg bits; $\#B = \#E = m - 126$, $\#G = 128$ and so $\#CT = m + 2$.
- Decryption is not the inverse of encryption; each application of encryption or decryption increases length by 2 bits.
- *Serious* typo: we believe 127 should be 129 and then the description tallies with that given in the SAC 2007 paper.

Versions of XCB

XCBv1 (McGrew-Fluhrer): Cryptology ePrint Archive Report 2004/278, 2004.

- Was not accompanied by a proof of security.

Versions of XCB

XCBv1 (McGrew-Fluhrer): Cryptology ePrint Archive Report 2004/278, 2004.

- Was not accompanied by a proof of security.

XCBv2 (McGrew-Fluhrer): SAC 2007.

- Accompanied by a 'proof' of security.
- The IEEE standard specialises **XCBv2** (of SAC 2007) in the following ways:
  - specifies the block cipher as AES;
  - specifies the message length to be a multiple of 8;
  - introduces a serious typo.

Covered by US patent number 7418100 dated August 26, 2008.

Distinguishing Attack on XCBv2

Consider **XCBv2** with AES and let

$$\begin{aligned} C^{(1)} &= \mathrm{XCBv2}_K^T(0^{2\times 128+8}); \\ C^{(2)} &= \mathrm{XCBv2}_K^T(0^{3\times 128}). \end{aligned}$$

The first 128 bits of $C^{(1)}$ and $C^{(2)}$ are equal!

Distinguishing Attack on XCBv2

Consider **XCBv2** with AES and let

$$\begin{aligned} C^{(1)} &= \mathrm{XCBv2}_K^T(0^{2\times 128+8}); \\ C^{(2)} &= \mathrm{XCBv2}_K^T(0^{3\times 128}). \end{aligned}$$

The first 128 bits of $C^{(1)}$ and $C^{(2)}$ are equal!

- For an n-bit block cipher, change 128 to n.
- 8 can be changed to any i with $1 \leq i \leq n-1$.
- The idea can be extended to obtain longer length distinguishing pairs of plaintexts.

Our Results on XCB

XCBv2:

- An easy distinguishing attack which also applies to the IEEE standard.
  - The attack does not apply if messages are restricted to be only full block messages.

Our Results on XCB

XCBv2:

- An easy distinguishing attack which also applies to the IEEE standard.
  - The attack does not apply if messages are restricted to be only full block messages.
- For full block messages:
  - The security proof in the SAC 2007 paper is incorrect. This is shown by counter-examples to the core collision analysis.
  - A new security proof is provided where the security bound is *significantly* weaker than what has been claimed.
  - The idea is motivated by Iwata et al, Crypto 2012 paper, though the counter-examples are different.

Our Results on XCB

XCBv2:

- An easy distinguishing attack which also applies to the IEEE standard.
  - The attack does not apply if messages are restricted to be only full block messages.
- For full block messages:
  - The security proof in the SAC 2007 paper is incorrect. This is shown by counter-examples to the core collision analysis.
  - A new security proof is provided where the security bound is *significantly* weaker than what has been claimed.
  - The idea is motivated by Iwata et al, Crypto 2012 paper, though the counter-examples are different.

XCBv1:

- The first security proof for this construction is provided; works for all length (i.e., not necessarily full block) messages.
- The security bound is similar to that of XCBv2.

XCBv1 versus XCBv2

Why move from XCBv1 to XCBv2?

XCBv2 “incorporates changes that make its security properties easier to analyze” (SAC 2007).

XCBv1 versus XCBv2

Why move from XCBv1 to XCBv2?

XCBv2 “incorporates changes that make its security properties easier to analyze” (SAC 2007).

On the contrary:

- The distinguishing attack on XCBv2 does not work on XCBv1. So, really XCBv2 incorporates changes that make it easier to attack.
- Proving security of XCBv1 is not much more difficult than proving security of XCBv2 (for full block messages).

XCBv1 versus XCBv2

Why move from XCBv1 to XCBv2?

XCBv2 “incorporates changes that make its security properties easier to analyze” (SAC 2007).

On the contrary:

- The distinguishing attack on XCBv2 does not work on XCBv1. So, really XCBv2 incorporates changes that make it easier to attack.
- Proving security of XCBv1 is not much more difficult than proving security of XCBv2 (for full block messages).

Concrete Security of XCB:

- XCBv2 (restricted to full-block messages) and XCBv1 have roughly the same security bound.
- In concrete terms, this security bound is about 2^{20} times weaker than what alernative schemes offer.

Future of XCB, IEEE XCB-AES?

Unknown!

Future of XCB, IEEE XCB-AES?

Unknown!

- Significantly better alternatives to XCB are known.
  - Better both in terms of security and efficiency.
  - No patent claims.
  - Some of these alternatives were already known when IEEE declared XCB as a standard!

Will anything change?

Aside on EME2-AES in IEEE Std 1619.2$^{\text{TM}}$-2010

EME2-AES Decryption Algorithm: Page 21, Table 4, Line 10:

$$CCC_j = \text{AES-Enc}(K_{\text{AES}}, L \oplus C_j)$$

should change to

$$CCC_j = \text{AES-Dec}(K_{\text{AES}}, L \oplus C_j).$$

Otherwise, decryption will not be the inverse of encryption!

Another serious(?) typo?

The above typo is courtesy of Cuauhtemoc Mancillas-López.

Details of the claims to be soon posted on eprint.

Thank you for your kind attention!