Comments on XTS-AES

Moses Liskov* Kazuhiko Minematsu†

September 2, 2008

1 Introduction

This is a comment in response to the request for comment on XTS-AES, as specified in IEEE Std. 1619-2007 [7]. Overall, we believe that the XTS-AES algorithm, closely based on Rogaway's XEX mode [5] plus ciphertext stealing, is a good choice for the purpose of block-oriented data storage encryption, and the use of an algorithm of this type is well supported by research publications. We have two main criticisms of the publication. First, while XEX uses one key, the proposed XTS algorithm uses two keys; Key_1 is used to encipher the whitened plaintext, while Key_2 is used to compute the pre- and post-whitening values. We feel that only one key should be used, to serve both purposes. Second, the draft incompletely analyzes the security of XTS-AES; it needs correction and expansion in a couple of areas.

2 On the Use of Two Keys

The XTS-AES encryption procedure for a single block is described in the draft [7] as follows. "The key is parsed as a concatenation of two fields of equal size called Key_1 and Key_2, such that $Key = Key_1|Key_2$. The ciphertext shall then be computed by the following or an equivalent sequence of steps:

1. $T \leftarrow AESenc(Key_2, i) \otimes \alpha^j$
2. $PP \leftarrow P \oplus T$
3. $CC \leftarrow AESenc(Key_1, PP)$

*The College of William and Mary
†NEC Corporation

4. $C \leftarrow CC \oplus T$" [page 4, lines 36-42].

In contrast, the XEX algorithm, using a single α and similar notation, would use a key half the size, Key, following this sequence:

1. $T \leftarrow AESenc(Key, i) \otimes \alpha^j$
2. $PP \leftarrow P \oplus T$
3. $CC \leftarrow AESenc(Key, PP)$
4. $C \leftarrow CC \oplus T$

The advantages of using a single key are obvious: the key length required would be halved, yet the level of security would remain effectively the same. The draft does little to justify the choice to modify XEX so that two keys are used instead of one:

- On page 15, lines 26-45, the XEX transform is described, for informational purposes. The draft incorrectly claims that the XEX transform uses two keys just as XTS does. This appears to be a simple typographical mistake, but as it does appear to support the use of two keys, we feel it is necessary to address the point. This is not the case; see, for instance, the full version of Rogaway's paper [6], page 4:

  "**Definition 2 [XEX construction]** *Let* $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ *be a blockcipher, let* $\alpha_1, \ldots, \alpha_k \in \mathbb{F}_{2^n}^*$, *and let* $\mathbb{I}_1, \ldots, \mathbb{I}_k \subset \mathbb{Z}$. *Then* $\overline{E} = \mathsf{XEX}[E, \alpha_1^{\mathbb{I}_1} \cdots \alpha_k^{\mathbb{I}_k}]$ *is the tweakable blockcipher* $\overline{E} : \mathcal{K} \times (\{0,1\}^n \times \mathbb{I}_1 \times \cdots \times \mathbb{I}_k) \times \{0,1\}^n \to \{0,1\}^n$ *defined by* $\overline{E}_K^{Ni_1 \ldots i_k}(M) = E_K(M \oplus \Delta) \oplus \Delta$ *where* $\Delta = \alpha_1^{i_1} \alpha_2^{i_2} \cdots \alpha_k^{i_k} \mathsf{N}$ *and* $\mathsf{N} = E_K(N)$"

  Note that both $\overline{E}_K(M)$ and N are computed using E_K, that is, with the same key.

  The document, however, later acknowledges that the XEX construction described by Rogaway uses a single key (page 17, line 53). Thus, the section on page 15, lines 26-46 is incorrect and should be altered in future versions, and its current text does not justify the use of two keys.

- The document later specifically explains the motivation for the use of two keys. The described cipher modes "use separate keys for tweaking and encryption purposes. This separation is a specific example

of separation of key usage by purpose and is considered a good security design practice (see NIST Key Management Guidelines [8], part 1, section 5.2)." (page 18, lines 1-3). This step, we believe, is both unnecessary, and a misapplication of the security design practice described in the NIST guidelines.

First, the use of two keys is unnecessary, because the XEX construction has been proven secure using a single key. So, design principles aside, we have a proof that the use of the same key for these two purposes in this case is secure. Indeed, the suggestion in the draft that two keys should be used amounts to a criticism of Rogaway's XEX mode, which has not appeared in the literature, does not address the explicit proofs given, and which does not have an explicit basis.

Second, the justification given in the draft erroneously conflates two notions of the word "purpose." In the sense of the NIST guidelines [8], the word purpose refers to the aim of the cryptographic application in which the key is used. The NIST guideline says, "In general, a single key **should** be used for only one purpose (e.g. encryption, authentication, key wrapping, random number generation, or digital signatures)." In the XTS construction, the key is being used for only one purpose: tweakable encryption. Tweakable encryption is not specifically mentioned in the NIST guideline, but tweakable encryption, like (non-tweakable) encryption, random number generation, authentication, et cetera, is meant to be an *atomic* cryptographic construct in practice. Because of this, the justification given in the draft seems to assert that two different uses of a value within an atomic cryptographic operation amount to two different purposes.

An analogy may be appropriate. When encrypting a long message with a block cipher in CBC mode, we run the block cipher in each round with the same key. While this constitutes multiple uses of the key itself, this is not an example of using the same key for multiple purposes; all these uses of the key are merely internal steps in an algorithm with an overall purpose: to encrypt the long message. XTS is similar in that it is an algorithm with an overall purpose: to tweakably encrypt a block of input.

Furthermore, the justifications for the NIST guideline design principle do not apply in this case. The justifications are:

1. "The use of the same key for two different cryptographic processes may weaken the security provided by one or both of the

processes."

This makes it even clearer that the principle isn't meant to apply here, as the two uses of the keys in XTS are not different processes. In addition, the construction is proven secure with a single key.

2. "Limiting the use of a key limits the damage that could be done if the key is compromised."

   The XTS construction is insecure if Key_2 is compromised, and has no analysis justifying its security if Key_1 is compromised. The compromise of *either* key invalidates our security analysis completely. The separation does not help. (If one of the keys were used for some other purpose besides XTS, then that purpose might not be affected if the other key were revealed. But this scenario would in itself violate the design principle.)

3. "Some uses of keys interfere with each other."

   This is not the case here.

Furthermore, the draft does not explicitly analyze the security of XTS; it relies instead of the analysis of XEX mode. XEX mode is only proven secure with one key. If two keys are to be used, the security of this choice should be explicitly addressed. We describe a better security analysis in the following section.

3 On Security Analysis

The draft gives a rather weak explanation of the security analysis for XTS-AES. Currently, the errors are:

1. The draft attempts to rely on the security analysis for XEX that cannot be directly applied when XEX is altered to use two keys,
2. The draft mis-cites Rogaway's analysis [6], which gives a bound of $9.5q^2/2^n$,[1]
3. There is better analysis possible, both if the use of two keys is kept, or if it is removed, and
4. There is insufficient analysis of the mode of operation, that is, of the use of sequential-tweak ECB mode with ciphertext stealing.

[1] It is possible that the authors confused Theorem 1 of [5] / Theorem 7 of [6], which gives a bound of $4.5q^2/2^n$ but for XE rather than XEX.

3.1 Security Proof of XEX

In the draft, XEX is explained as a provably secure scheme having $4.5q^2/2^n$ security bound, when n-bit block cipher is used and the adversary is allowed to ask at most q queries for both encryption and decryption, i.e., Chosen-Ciphertext Attack (CCA). However the bound shown by Rogaway's papers is larger: $9.5q^2/2^n$ (Theorem 2 of [5], Theorem 8 of [6]). The improvement to $4.5q^2/2^n$ is due to Minematsu [3], which is not mentioned in the draft. This improvement is obtained by the following analysis. If the standard is to be adjusted to use a single key, the justification should cite Minematsu's analysis for the security bound of $4.5q^2/2^n$.

3.1.1 Analysis

We first consider a general tweakable block cipher using two keyed permutations over $\mathcal{M} = \{0,1\}^n$, $E : \mathcal{K}_1 \times \mathcal{M} \to \mathcal{M}$ and $G : \mathcal{K}_2 \times \mathcal{M} \to \mathcal{M}$. Two keys, $K_1 \in \mathcal{K}_1$ and $K_2 \in \mathcal{K}_2$, can be dependent or independent. Specifically, we consider tweak consisting of two parts, $i \in \mathcal{M}$ and $j \in \mathcal{J}$ ($(i,j) \in \mathcal{T} = \mathcal{M} \times \mathcal{J}$).

The encryption is written as:

$$
\begin{aligned}
T &\leftarrow f(j, G_{K_2}(i)) \\
PP &\leftarrow P \oplus T \\
CC &\leftarrow E_{K_1}(PP) \\
C &\leftarrow CC \oplus T
\end{aligned}
$$

where $(i,j) \in \mathcal{T}$ is the tweak and $P \in \mathcal{M}$ ($C \in \mathcal{M}$) is the plaintext (ciphertext) block. Decryption procedure is clear, thus omitted. The function $f : \mathcal{J} \times \mathcal{M} \to \mathcal{M}$ is a deterministic function[2] called the offset function. Let us denote the above scheme by $\mathrm{TW}[E_{K_1}, G_{K_2}, f]$.

We say f is (ϵ, γ, ρ)-uniform if

1. $\max_{j \neq j', \delta \in \mathcal{M}} \Pr[f(j,V) \oplus f(j',V) = \delta] \leq \epsilon$
   and $\max_{j \neq j', \delta \in \mathcal{M}} \Pr[f(j,V) \oplus f(j',V') = \delta] \leq \epsilon$
2. $\max_{j \in \mathcal{J}, \delta \in \mathcal{M}} \Pr[f(j,V) = \delta] \leq \gamma$
3. $\max_{j \in \mathcal{J}, \delta \in \mathcal{M}} \Pr[f(j,V) \oplus V = \delta] \leq \rho$

where V and V' are independent and uniformly random over $\mathcal{M}$.

[2] This can be probabilistic, but for simplicity we consider only deterministic functions.

Let URP_n denote the uniform random permutation (URP), which is a permutation uniformly chosen from all permutations over $\mathcal{M}$. We derive an upper bound of maximum advantage in distinguishing $\mathrm{TW}[\mathsf{URP}_n, \mathsf{URP}_n, f]$ from the perfect tweakable permutation (PTP), which is a collection of independent URPs indexed by the tweak, for any CCA-adversaries with q queries and infinite computational power. Here, two URPs in $\mathrm{TW}[\mathsf{URP}_n, \mathsf{URP}_n, f]$ share *the same internal randomness*. This maximum advantage is denoted by $\mathtt{Adv}_{\mathrm{TW}[\mathsf{URP}_n,\mathsf{URP}_n,f]}(q)$. Then, Theorem 4 of [3] proved that if f is (ϵ, γ, ρ)-uniform, we have

$$\mathtt{Adv}_{\mathrm{TW}[\mathsf{URP}_n,\mathsf{URP}_n,f]}(q) \leq \left(2\epsilon + \gamma + \rho + \frac{1}{2^{n+1}}\right) q^2. \tag{1}$$

If we define f as

$$f(j, v) \stackrel{\text{def}}{=} v \otimes \alpha^j, \tag{2}$$

where α is a primitive element over $\mathrm{GF}(2^n)$ and $\mathcal{J} = \{1, 2, \ldots, 2^{n-2}\}$, we clearly obtain an instance of XEX. In this case, we can easily confirm that f is $(2^{-n}, 2^{-n}, 2^{-n})$-uniform and thus the security bound for XEX using URP_n, denoted by XEX-URP_n, is

$$\mathtt{Adv}_{\text{XEX-}\mathsf{URP}_n}(q) \leq (2 \cdot 2^{-n} + 2^{-n} + 2^{-n} + 2^{-(n+1)})q^2 = 4.5q^2/2^n. \tag{3}$$

Note that $j = 0$ must be excluded, as $f(0, v) = v$ for any v, which implies $\rho = 1$. Moreover, if $j = 0$ was allowed, a simple attack based on this fact existed, as pointed out by [6] and [3]. **Hence if XEX is used, one must be careful to avoid j being 0.**

The security of XEX-AES is obtained by combining Equation (3) for $n = 128$ and the standard technique for converting information-theoretical result into computational one, which is as follows.

$$\mathtt{Adv}_{\text{XEX-AES}}(q, \tau) \leq \mathtt{Adv}_{\mathrm{AES}}(2q, \tau') + \frac{4.5q^2}{2^{128}}, \tag{4}$$

where τ denotes the adversary's time complexity and $\mathtt{Adv}_{\mathrm{AES}}(2q, \tau')$ is the maximum advantage in distinguishing AES from URP_{128} using CCA with $2q$ queries and $\tau' = \tau + O(q)$ time complexity ($O(q)$ can be more specific, see ,e.g. Rogaway [6]).

Notes. The above analysis crucially depends on the methodology developed by Maurer [2]. It is not much popular, however, for some cases (including this case) it can offer a slight tighter bound than previous ones.

For example, the security bound of 3-round Feistel permutation using independent pseudorandom functions, (aka 3-round Luby-Rackoff cipher) was previously known as $2\binom{q}{2}/2^n + \binom{q}{2}/2^{2n}$ [4]. However Maurer [2] proved that the latter term could be removed.

3.2 Security Proof of XTS

As currently described in the standard, XTS differs from XEX with ciphertext stealing in that it uses two keys. We have argued in section 2 that only one key should be used, and this remains our primary suggestion. However, if this suggestion is not accepted, we note that the current security analysis cited in the document does not apply to XTS, since it does not adjust the analysis of XEX for the use of two keys. Below, we give a security analysis for XTS as written that is better than the current draft. Specifically, it (1) achieves a better security bound, and (2) avoids trying to apply the analysis of XEX inappropriately.

3.3 Analysis

Using the definitions of previous section, XTS-AES without ciphertext stealing is defined as $\mathrm{TW}[\mathrm{AES}_{K_1}, \mathrm{AES}_{K_2}, f]$ (K_1 and K_2 are independent), while XEX-AES is $\mathrm{TW}[\mathrm{AES}_K, \mathrm{AES}_K, f]$, where f is as defined as Equation (2).

Although the security proof of XTS can not be directly derived from the original proof of Rogaway [5][6], it is easily derived from the analysis of Section 3.1 with a minor modification. The result is almost the same as the results of Liskov et al. [1], but slightly improves it. The analysis is as follows. We first observe that the procedure defined as

$$T \leftarrow f(j, \mathsf{URP}_n(i)), \text{ where } f \text{ is defined as Eq. (2)} \tag{5}$$

is 2^{-n}-Almost XOR uniform (2^{-n}-AXU) for any $(i, j) \in \mathcal{T}$, where $\mathcal{T} = \mathcal{M} \times \{0, 1, \ldots, 2^{n-2}\}$. That is, for all $(i, j), (i', j') \in \mathcal{T}$ and $(i, j) \neq (i', j')$, we have

$$\Pr[f(j, \mathsf{URP}_n(i)) \oplus f(j', \mathsf{URP}_n(i')) = \delta] \leq 2^{-n}. \tag{6}$$

Using this fact and a result of Liskov et al. (Theorem 2 of [1]), we can obtain

$$\mathtt{Adv}_{\mathrm{TW}[\mathsf{URP}_n^{(1)}, \mathsf{URP}_n^{(2)}, f]}(q) \leq \frac{3q^2}{2^n}, \tag{7}$$

where $(\mathsf{URP}_n^{(1)}, \mathsf{URP}_n^{(2)})$ denotes a pair of independent URPs over $\mathcal{M}$. Moreover, Minematsu (Theorem 1 of [3]) improved[3] this to

$$\mathtt{Adv}_{\mathrm{TW}[\mathsf{URP}_n^{(1)},\mathsf{URP}_n^{(2)},f]}(q) \leq \frac{q^2}{2^n}. \tag{8}$$

The maximum (q,τ)-CCA advantage in distinguishing $(\mathsf{URP}_{128}^{(1)}, \mathsf{URP}_{128}^{(2)})$ from $(\mathrm{AES}_{K_1}, \mathrm{AES}_{K_2})$, is at most $2\mathtt{Adv}_{\mathrm{AES}_K}(q,\tau')$. The formal security proof of XTS (w/o ciphertext stealing) is obtained by this observation and Equation (8), which is as follows.

$$\mathtt{Adv}_{\text{XTS-AES}}(q,\tau) \leq 2\mathtt{Adv}_{\mathrm{AES}}(q,\tau') + \frac{q^2}{2^{128}}. \tag{9}$$

This bound is almost the same as the bound of XEX shown by Equation (4).

Note that XTS does not require to avoid $j = 0$, as the offset function of XTS needs not be (ϵ,γ,ρ)-uniform, but only be ϵ-AXU if it is combined with URP. This difference is significant in security, but has no impact on effectiveness for practical applications.

3.4 Ciphertext Stealing

In addition, the XTS construction varies from the XEX construction in that XTS is to be used in an implied mode of operation: namely, a large chunk of data is to be encrypted block by block, where a large number of blocks use the same i value but sequential j values. XEX is intended for use in this mode also. However, XTS handles the use of a non-full block via the well-known "ciphertext stealing" trick, whereas XEX does not. The draft does not justify the implied assertion that the use of ciphertext stealing in this algorithm is secure.

The following is a sketch of a proof that ciphertext stealing is sound if the basic scheme is sound. Let A be an adversary. If, each time A makes a query involving a partial block, A were to learn CP in addition to $C_1,\ldots,C_m$, the probability for A to succeed would not diminish. Thus, it cannot hurt to reveal $C_1,\ldots,C_m|CP,C_{m-1}$. This, in turn, can be determined by the adversary without making any queries involving a partial block, if the adversary may specify i,j at will: the adversary simply queries $P_1,\ldots,P_{m-1}$ for $i,1$

[3]In fact, the results are more general: if Equation (5) is replaced with any keyed function being ϵ-AXU for all possible tweaks and the key is *independent*, Liskov et al. proved the bound $3\epsilon q^2$ and Minematsu improved it to ϵq^2.

through $i, m-1$, which reveals $C_1, \ldots, C_{m-2}, C_m|CP$. Now that the adversary has learned CP, the simply make another query with respect to the same tweaks: $P_1, \ldots, P_{m-1}, P_m|CP$. Thus, if we assume that no adversary can break the scheme if they were to make no partial block queries, then no adversary can break the scheme with ciphertext stealing either.

4 Editorial and Other Comments

1. The draft seems to confuse the conference version and the full version Rogaway's paper. The bibliography lists only the conference version [5], but many of the references directly reference the full version [6]. (For instance, there is no Theorem 8 in the conference version, but it is cited on page 16 line 11.)

2. It should be mentioned explicitly in the description that when enciphering many blocks, successive T values can and *should* be computed from prior ones via multiplication by α (providing that i remains fixed). This optimization, which is one of the best features of XEX, should be explicitly recommended in the standard.

3. In the description of α on page 3, lines 6-7, we take it that α is the $GF(2^{128})$ element corresponding to 2. But this should be made explicit. Furthermore, XEX mode allows for various choices for α and even for the use multiple distinct α values. The choice of this particular value needs to be explained and justified. We believe that the implied value of α is a good choice, because multiplication by 2 is especially inexpensive.

4. On page 5, line 12: i should be "the value of the 128-bit *nonce*", since the term "tweak" refers, properly, to the tweak input of the tweakable blockcipher, in this case, one of the values (i, q) through $(i, m-1)$.

5. In the security analysis (section D.4.2), we feel there is too much emphasis on the security advantage bound (e.g. $4.5q^2/2^n$) and not enough emphasis on the fact that XTS-AES is not absolutely secure, and its security rests on the security of AES.

References

[1] M. Liskov, R. Rivest, and D. Wagner. "Tweakable Block Ciphers." *Advances in Cryptology*- CRYPTO'02, LNCS 2442, pp. 31-46, 2002.

[2] U. Maurer. "Indistinguishability of Random Systems." *Advances in Cryptology*- EUROCRYPT'02, LNCS 2332, pp. 110-132, 2002.

[3] K. Minematsu. "Improved Security Analysis of XEX and LRW Modes." *Selected Areas in Cryptography*- SAC'06, LNCS 4356, pp.96-113, 2007.

[4] M. Naor and O. Reingold. "On the Construction of Pseudorandom Permutations: Luby-Rackoff Revisited." *Journal of Cryptology*, vol. 12, no. 1, pp. 29-66, 1999.

[5] P. Rogaway. "Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC." *Advances in Cryptology*- ASIACRYPT'04. LNCS 3329, pp. 16-31, 2004.

[6] P. Rogaway. "Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC." (the full version of [5]), available from `http://www.cs.ucdavis.edu/~rogaway/papers`

[7] The XTS-AES Tweakable Block Cipher (An Extract from IEEE Std 1619-2007),
`http://grouper.ieee.org/groups/1619tmp/1619-2007-NIST-Submission.pdf`

[8] NIST Key Management Guidelines SP800-57.