Revisiting AES Key Wrap: Forgery Bounds, Structural Observations, and Recommendations for the Revision of NIST SP 800-38F

John Preuß Mattsson
Ericsson Research, Stockholm, Sweden

October 2026

Abstract

NIST is revising SP 800-38F, which specifies the AES Key Wrap modes KW and KWP. This paper is an updated and expanded version of our technical comments to NIST. We show that the forgery probabilities stated in SP 800-38F do not account for adaptive attacks, give the resulting forgery probabilities for KW, KWP, and TKW, and show that checking the expected plaintext length can raise the security level of KWP considerably. Revisiting Kelsey's long-message observation, we correct the collision conditions, show that a successful forgery leaks a plaintext prefix, derive an encryption-only distinguisher, and conclude that KW does not have beyond-birthday-bound security. We explain the role of the step counter, analyse the single-semiblock path of KWP and the trade-off between the number of passes and the length limit, and give recommendations for the revision, for implementations, and for future key-wrap designs.

Keywords: key wrapping, AES Key Wrap, KW, KWP, DAE, deterministic authenticated encryption, NIST SP 800-38F, reforgeability.

1 Introduction

Key wrapping, the authenticated encryption of cryptographic keys under a key-encryption key (KEK), is one of the most widely deployed uses of symmetric cryptography. NIST first posted an informal *AES Key Wrap Specification* in 2001 [4]; equivalent algorithms were published by the IETF as RFC 3394 [5] and, with a padding scheme, as RFC 5649 [6]. In 2012 NIST standardized the algorithms in SP 800-38F [3] as KW and KWP (with AES) and TKW (with TDEA, also known as Triple DES or 3DES). AES-KW and AES-KWP are used in, among other places, CMS and S/MIME [7], JOSE [8], and the PKCS#11 [9] and KMIP [10] interfaces to hardware security modules. AES-KWP is required in the S/MIME profile for the Commercial National Security Algorithm Suite 2.0 (CNSA 2.0) [24].

On May 5, 2026 NIST requested pre-draft public comments on a revision of SP 800-38F [2]. The stated goals are to remove TKW and to revisit the approval, in Section 3.1 of SP 800-38F, of unspecified combinations of approved encryption and authentication methods. NIST received eleven sets of comments, which have been published [1]. Our comments (pp. 4–14 of [1]) consist of general comments addressing NIST's questions and a set of detailed technical comments on SP 800-38F itself. This paper is based on the detailed comments, which we believe are of interest beyond the standardization process: they concern the concrete security of a construction that has seen remarkably little public analysis given its deployment, and several of them apply to future key-wrap designs as well. We have expanded the comments with derivations and proofs, and we have corrected or refined several of the numbers.

Prior work. Public analysis of the AES Key Wrap is scarce. Rogaway and Shrimpton [14] formalized deterministic authenticated encryption, criticized the lack of stated design goals for the algorithms in ANS X9.102 [13], including AESKW, and noted the absence of security proofs. Kelsey's long-message observation, described in Appendix A.4 of SP 800-38F [3], is to our knowledge the only published structural attack on the wrapping function and motivates its length limits. Comments received by NIST during the development and revision of SP 800-38F [11, 1] contain further observations. We are not aware of a security proof for KW or KWP.

Contributions. Our main observations are:

- The "1 in 2^{64}" forgery probability stated in SP 800-38F is for a random string. Taking Kelsey's adaptive attack into account, the total forgery probability is $P_{\mathrm{F}} = 2^{-64} + P_{\mathrm{A}}$, where P_{A} is the additional probability from Kelsey's attack; the requirement is therefore $P_{\mathrm{A}} \ll 2^{-64}$, not $P_{\mathrm{A}} < 2^{-64}$. Under the mandatory limits, $P_{\mathrm{F}} \approx 2^{-63.97}$ for KW and $P_{\mathrm{F}} \approx 2^{-31.56}$ for TKW (Section 4).
- An exact expression for the forgery probability of KWP against known attacks, $\approx 2^{-63.99}$, and the observation that a receiver that enforces the expected plaintext length obtains up to 2^{-112} for common key sizes (Section 4).
- A revisit of Kelsey's observation: corrected collision conditions; confirmation that the threshold of about 2^{67} semiblocks in SP 800-38F is correct; a successful forgery reveals a plaintext prefix, contradicting the robustness goal of the original NIST requirements; an encryption-only distinguisher; and the conclusion that W does not have beyond-birthday-bound security (Section 5).
- A structural justification for the step counter in W: without it, W has about $\sigma_0(6m)$ fixed points, several times more than a random permutation (Section 6).
- An analysis of the single-semiblock path of KWP, which reuses the KEK with the bare block cipher, and a justification of the 2^{32}-octet plaintext limit of KWP based on reforgeability resistance [12] (Section 7).
- A parametrised forgery estimate for KW-like designs with r passes, which suggests that, with respect to Kelsey's attack, fewer than six passes combined with a stricter length limit might give a comparable forgery probability (Section 8).
- Implementation, deployment, and editorial recommendations (Sections 3 and 9).

Scope. Our general comments to NIST, which concern generic composition of encryption and authentication, unauthenticated modes, ECB terminology, and key transport in signalling protocols, are outside the scope of this paper.

2 Preliminaries

We follow the notation of SP 800-38F [3]. For KW and KWP the block cipher CIPH_K has a 128-bit block, and a *semiblock* is 64 bits; for TKW the block is 64 bits and a semiblock 32 bits. $[x]_s$ denotes the s-bit encoding of the integer x.

The wrapping function W. On input $S = S_1 \parallel \cdots \parallel S_n$ of $n \geq 3$ semiblocks, let $m = n - 1$ and $s = 6m$. Set $A^0 = S_1$ and $R_i^0 = S_i$ for $2 \leq i \leq n$. For $t = 1, \ldots, s$:

$$A^t = \mathrm{MSB}_{64}\big(\mathrm{CIPH}_K(A^{t-1} \parallel R_2^{t-1})\big) \oplus [t]_{64} \,, \tag{1}$$

$$R_i^t = R_{i+1}^{t-1} \quad (2 \leq i \leq n-1), \qquad R_n^t = \mathrm{LSB}_{64}\big(\mathrm{CIPH}_K(A^{t-1} \parallel R_2^{t-1})\big) \,. \tag{2}$$

The output is $C = A^s \,\|\, R_2^s \,\|\, \cdots \,\|\, R_n^s$. Thus W is an unbalanced Feistel-like construction with a 64-bit accumulator A and an m-semiblock shift register R, iterated for six passes over the register. The unwrapping function W^{-1} runs the steps backwards, $t = s, \ldots, 1$, using CIPH_K^{-1}. TKW uses the analogous TW with 32-bit semiblocks.

KW, KWP, and TKW. KW-AE$(P) = W(\mathrm{ICV}_1 \,\|\, P)$ with $\mathrm{ICV}_1 =$ `0xA6A6A6A6A6A6A6A6` for plaintexts of 2 to $2^{54} - 1$ semiblocks; KW-AD applies W^{-1} and checks the first semiblock. KWP-AE accepts octet strings P of ℓ octets, $1 \le \ell \le 2^{32} - 1$, and forms

$$S = \mathrm{ICV}_2 \,\|\, [\ell]_{32} \,\|\, P \,\|\, 0^{8 \cdot padlen}, \qquad \mathrm{ICV}_2 = \texttt{0xA65959A6},\ padlen = 8\lceil \ell/8 \rceil - \ell \in \{0, \ldots, 7\}\ , \quad (3)$$

and outputs $\mathrm{CIPH}_K(S)$ if $\ell \le 8$ and $W(S)$ otherwise. KWP-AD checks ICV_2, checks that the 32-bit length field $[\ell]_{32}$ is consistent with the ciphertext length, and checks that the padding octets are zero. TKW is KW with TDEA, 32-bit semiblocks, and $\mathrm{ICV}_3 =$ `0xA6A6A6A6`, limited to $2^{28} - 1$ plaintext semiblocks. All three are deterministic authenticated-encryption (DAE) schemes in the sense of Rogaway and Shrimpton [14], following the encode-then-encipher paradigm [15]: a fixed redundancy is added and the result is enciphered with a variable-input-length permutation.

3 Security Goals and Design Rationale

3.1 Key wrapping versus protection of general data

Section 3.1 of SP 800-38F states that KW, KWP, and TKW each provide "an option for protecting keys in a manner that is distinct from the methods that protect general data", and that "segregating keys from general data can provide an extra layer of protection." The same section, however, also approves KW, KWP, and TKW for the protection of general data, so they are not, in fact, distinct from the methods that protect general data. More importantly, the extra layer of protection does not come from using a distinct algorithm as such. It comes from (i) using independent, and potentially longer, keys for key protection than for data protection, (ii) separating key protection from general data protection so that a compromise or misuse of one does not affect the other, and (iii) choosing a more robust algorithm for the protection of long-lived, high-value keys. We suggest that the revision explains this rationale.

KW was a good design for its era: it pioneered the standardization of robust authenticated encryption, using a variable-input-length permutation and encode-then-encipher, long before these notions were common. In KW, the expected number of successful forgeries after q forgery attempts is $\approx q/2^{64}$, and KW therefore provides reforgeability resistance (Section 4). We would have liked to see wider use of AES-KW(P) in applications where performance is not critical. Modes derived from accordions [17, 18] should retain the advantages of KWP while addressing its limitations: performance, nonce-misuse resistance, IND-CCA2 security with nonces, support for associated data, variable-length padding, lower forgery probability, formal security proofs, and AERO-style replay protection [16].

3.2 Determinism and nonces

The 2001 NIST Key Wrap requirements [4] are, in our view, well designed and sound. They include the robustness requirement that every ciphertext bit should depend in a highly nonlinear way on every plaintext bit, and vice versa when unwrapping. In contrast, later documents motivating deterministic key wrap lack a clear rationale. The ANSI request for review [13] identifies IND-CCA2 as the intended goal but does not explain why nonces are excluded or why algorithms that do not even achieve IND-CPA security are considered. Similarly, Rogaway and Shrimpton [14] incorrectly equate the absence of nonces with the absence of randomness,

and suggest that a deterministic scheme without nonces is "just as good" as IND-CCA2 for transporting cryptographic keys.

We do not believe that determinism and the absence of nonces should be design criteria for key wrapping. Determinism directly causes the "equality of plaintexts" leakage of SP 800-38F, Appendix A.1: wrapping the same key twice under the same KEK yields identical ciphertexts. Future designs should instead aim for IND-CCA2 security when nonces are unique, while retaining DAE (misuse-resistant) security [14] when the random number generator is poor or nonces repeat. We also suggest that NIST republishes the original 2001 specification [4]; archives are essential for traceability of design decisions and for trust in the standardization process.

3.3 The ICV constants

SP 800-38F calls ICV_1 and ICV_2 "default" ICVs, suggesting that other values may be used, while Section 8 of SP 800-38F states that W and W^{-1} are not approved for use independently of KW and KWP. We suggest removing "default". The revision could also explain how the constants were chosen. We note that `0xA6` = `10100110` has Hamming weight 4, and `0x59` is its bitwise complement. Special constants invite the question of whether security depends on them. We welcome that NIST plans to use all-zero ICVs in accordion-derived functions [17], which makes clear that the security of these constructions does not depend on carefully chosen non-trivial constants.

3.4 Choice of cipher direction and key separation

SP 800-38F allows the "designated cipher function" to be either the AES forward or inverse transformation. This choice does not improve security and can hinder interoperability; we suggest restricting KW-AE and KWP-AE to the forward direction. We also recommend requiring that KW and KWP use cryptographically independent KEKs: the two modes share the wrapping function W, and, as discussed in Section 7, KWP also uses the bare block cipher.

3.5 Length limits

Section 5.3.1 of SP 800-38F states that "KW can accept longer inputs" than $2^{54}-1$ semiblocks. Since the analysis in Section 5 shows that KW cannot securely support longer inputs, we suggest removing this statement. Further, the plaintext and ciphertext limits of KWP are not aligned. A plaintext of $2^{32}-1$ octets is padded to 2^{32} octets, i.e., 2^{29} semiblocks, and the ciphertext is $2^{32}+8$ octets, i.e., $2^{29}+1$ semiblocks. The ciphertext limit in Table 1 of SP 800-38F should therefore be "2 to $2^{29}+1$ semiblocks".

4 Forgery Probabilities

4.1 Random strings versus adaptive adversaries

Appendix A.3 of SP 800-38F states that a random string of valid length is a genuine ciphertext with probability exactly 2^{-64} for KW, approximately 2^{-64} for KWP, and 2^{-32} for TKW. This is not a realistic attack model: an adversary would mount adaptive chosen-ciphertext attacks designed to maximize the success probability, such as the attack in Appendix A.4 of SP 800-38F (Section 5). In the model used in that appendix, the probability that a single modified ciphertext is accepted is

$$P_{\mathrm{F}} \;\approx\; 2^{-64} + P_{\mathrm{A}}, \qquad P_{\mathrm{A}} \approx \frac{1}{6!}\left(\frac{m}{2^{64}}\right)^{6}, \tag{4}$$

where m is the number of plaintext semiblocks. SP 800-38F, which calls this term P, concludes that $P_{\mathrm{A}} < 2^{-64}$ if $m < 2^{54}$. However, P_{A} is the *additional* forgery probability; to claim that KW achieves a forgery probability of about 2^{-64} one needs $P_{\mathrm{A}} \ll 2^{-64}$. This does hold for $m < 2^{54}$:

$$\mathsf{KW},\ m < 2^{54}: \qquad P_{\mathrm{F}} \lesssim 2^{-64} + \frac{2^{-60}}{6!} \approx 2^{-64} + 2^{-69.49} \approx 2^{-63.97}\ . \tag{5}$$

For TKW the analogous estimate with 32-bit semiblocks is

$$\mathsf{TKW},\ m < 2^{28}: \qquad P_{\mathrm{F}} \lesssim 2^{-32} + \frac{2^{-24}}{6!} \approx 2^{-32} + 2^{-33.49} \approx 2^{-31.56}\ , \tag{6}$$

which is not "approximately 2^{-32}". A limit of $m < 2^{27}$ gives $P_{\mathrm{A}} \approx 2^{-39.5}$ and $P_{\mathrm{F}} \approx 2^{-31.99}$, and is what the reasoning in Appendix A.4 of SP 800-38F actually supports. We recommend that the revision states that the forgery probability for KW is approximately 2^{-64} *including* adaptive attacks, and that it distinguishes clearly between P_{A} and P_{F}.

4.2 Forgery probability of KWP against known attacks

Proposition 1. *Model* CIPH_K^{-1} *(for $n = 2$) and, for each $n \geq 3$, W^{-1} on n-semiblock strings as independent uniformly random permutations. Let C be any purported ciphertext of $n \geq 2$ semiblocks that was not output by KWP-AE, possibly chosen adaptively after observing $q \ll 2^{64}$ wrapping and unwrapping queries. Then, up to a factor $1 + O(q \cdot 2^{-64n})$, C is accepted by KWP-AD with probability*

$$P_{\mathrm{F}} \;=\; 2^{-64} \sum_{padlen=0}^{7} 2^{-8 \cdot padlen} \;=\; 2^{-64} + 2^{-72} + 2^{-80} + \cdots \;\approx\; 2^{-63.994}\ . \tag{7}$$

If the receiver accepts only a single plaintext length ℓ, with the corresponding padlen, the probability is $2^{-64-8 \cdot padlen}$.

Proof. In this model, $S = W^{-1}(C)$ (or $\mathrm{CIPH}_K^{-1}(C)$) is uniformly distributed over the n-semiblock strings that are not the unwrapping of a previously observed ciphertext, i.e., it is uniform up to the stated factor, regardless of how C was chosen. KWP-AD accepts iff (i) the first 32 bits equal ICV_2 (probability 2^{-32}), (ii) the 32-bit length field $[\ell]_{32}$ encodes one of the eight values $8(n-1) - padlen$, $padlen \in \{0, \ldots, 7\}$ (each with probability 2^{-32}), and (iii) for the chosen *padlen* the last *padlen* octets are zero (probability $2^{-8 \cdot padlen}$). These events for different values of *padlen* are disjoint. If only one ℓ is accepted, only one term remains. Attacks that exploit the non-ideal structure of W for long inputs add the term P_{A} of Section 4. □

The second statement has practical value. Wrapping a 66-octet P-521 private key with KWP adds six zero octets, i.e., 48 bits of redundancy. A receiver that validates the exact expected length obtains $P_{\mathrm{F}} \approx 2^{-64-48} = 2^{-112}$ per forgery attempt. For a 32-octet plaintext no padding is added, and a receiver that checks the length obtains exactly 2^{-64}. We suggest that the revision encourages receivers to check the expected plaintext length whenever it is known.

4.3 Expected number of forgeries

For a strong integrity mechanism with sufficiently long keys, the expected number of successful forgeries after q attempts with τ bits of redundancy (typically the tag length) should be $\approx q/2^{\tau}$ for all practical q. Because KW is a (close to) ideal variable-input-length permutation for all practical lengths, a successful forgery gives the adversary no advantage for subsequent forgeries, and the expected number of forgeries after q attempts is $\approx q/2^{64}$, *even after a successful*

forgery. This compares favourably with polynomial-hash-based modes. In GCM, a single nonce repetition reveals the authentication key H, and successful forgeries leak information about H, enabling universal forgeries [19, 20], and the forgery bound of GCM grows with the message length [21]. In CCM the bound contains a term quadratic in the total amount of processed data [22]. As noted in our comments to NIST [1], for long messages AES-CCM with 128-bit tags is expected to experience more forgeries than AES-KW after only about 2^{33} forgery attempts. Despite its 64-bit ICV, AES-KW can therefore be the more robust choice in deployments where an adversary can make many forgery attempts. We suggest that NIST notes this property.

5 Kelsey's Long-Message Observation Revisited

Appendix A.4 of SP 800-38F describes an observation due to John Kelsey that motivates the length limits. Let S be a string of m semiblocks and $T \neq U$ single semiblocks, so that $S \,\|\, T$ and $S \,\|\, U$ are $(m+1)$-semiblock inputs to W^{-1} with $s = 6m$ steps. Let A^t and B^t denote the accumulator values in $W^{-1}(S \,\|\, T)$ and $W^{-1}(S \,\|\, U)$, respectively. The first unwrapping step consumes T (respectively U) and introduces a difference in both the accumulator and the register. A collision $A^i = B^i$ removes the difference from the accumulator, but the register still holds differing semiblocks, which are consumed again one pass later and re-introduce a difference. Six collisions, one per pass, are needed for the difference to vanish completely, in which case $A^0 = B^0$ and $S \,\|\, U$ passes the integrity check if $S \,\|\, T$ does.

5.1 Collision conditions and threshold

Notation. Appendix A.4 of SP 800-38F uses S for the input of W^{-1}, whereas Section 6.1 of SP 800-38F uses S for the input of W and C for the input of W^{-1}. Using C in that appendix would be clearer.

Collision conditions. For $i_6 = 6m$ there is always a collision $A^{6m} = B^{6m} = C_1$, since the first semiblock is common to both inputs; this collision occurs before the differing semiblock is consumed and does not imply collisions for the indices $5m < j < i_6$. Taking into account how differences propagate through the register, we believe the conditions should read

$$0 < i_1 < m, \qquad i_k + m \leq i_{k+1} \ (1 \leq k \leq 4), \qquad i_5 + m \leq i_6 < 6m \ , \tag{8}$$

with $A^{i_k} = B^{i_k}$ for all k. With these conditions, $W^{-1}(S \,\|\, T)$ and $W^{-1}(S \,\|\, U)$ agree on their first $d = i_1 + 1$ semiblocks, so the first i_1 semiblocks of the resulting plaintext coincide (SP 800-38F states $i_1 - 1$).

The modified semiblock. In addition to modifying the last ciphertext semiblock(s), the adversary can modify the first semiblock C_1, which also enters the first unwrapping step. We suggest that the revision mentions this possibility.

Threshold. Appendix A.4 of SP 800-38F states that forgeries become likely when S has "about 2^{67} semiblocks". This is correct: in the model of that appendix the success probability is a binomial tail, which is about 0.81 at $m = 2^{67}$. In our comments to NIST [1] we suggested $2^{65.41}$, which is incorrect, since it applied the approximation $P_{\mathrm{A}} \approx (m/2^{64})^6/6!$ for $m > 2^{64}$, where it is not valid.

5.2 Confidentiality and robustness

SP 800-38F notes that, after a successful forgery, the first semiblocks of the resulting plaintext are identical to those of the plaintext from which $S \,\|\, T$ was generated. This implies that an adversary with access to a decryption oracle that returns the plaintext of accepted ciphertexts learns a *prefix of the unknown plaintext*, i.e., the forgery is also a confidentiality attack. That a forgery enables attacks on confidentiality is common, but here it shows that W does not satisfy the robustness requirement of the original NIST Key Wrap requirements [4], that every plaintext bit should be a highly non-linear function of every ciphertext bit. We suggest that the revision says this explicitly.

5.3 An encryption-only distinguisher

The observation also gives a theoretical distinguisher in the chosen-plaintext setting. Consider q plaintexts of $m \approx 2^{54}$ semiblocks that are identical except in their first plaintext semiblock. By an argument analogous to the one above, applied to W instead of W^{-1}, each pair of resulting ciphertexts agrees on its last semiblocks with probability $\approx (m/2^{64})^6/6! = 2^{-60}/6!$. Over $\binom{q}{2}$ pairs, the probability that some pair of ciphertexts shares a common suffix is

$$p \;\approx\; \frac{q^2}{2} \cdot \frac{2^{-60}}{6!}\,, \tag{9}$$

which is $1/2$ for $q \approx 2^{34.75}$, i.e., a total data complexity of $\approx 2^{88.75}$ semiblocks. For a random permutation on strings of this length such suffix collisions do not occur with non-negligible probability. The attack is far from practical, but it is an encryption-only attack within the length limits of the standard.

5.4 Beyond-birthday-bound security

Appendix A.5 of SP 800-38F quotes Rogaway and Shrimpton's expectation that AES Key Wrap may achieve "beyond-birthday-phenomenon security". The analysis above shows that W does not achieve this in a strong sense: the success probability of Kelsey's attack is governed by the 64-bit accumulator. Section 7 gives a further, simpler, reason in the case of KWP. This is not an attack on KW within its intended use, but it is an important qualification of its security properties. We recommend removing the quoted phrase and adding to Appendix A.4 of SP 800-38F that the observation shows that KW falls short of beyond-birthday-bound security.

5.5 Mitigation by reversing a pass

The attack relies on all six passes processing the register in the same direction, so that a difference consumed at the end of one pass reappears at the same relative position in the next. Reversing the direction of one or more passes, for example processing the final pass in the opposite order, appears to break this alignment and mitigate the attack.

6 The Role of the Step Counter

Rogaway and Shrimpton [14] criticise "mysterious aspects" of the construction, among them the XOR of the counter into the block-cipher output. We disagree with that particular criticism. Injecting a step-dependent constant is standard practice: it breaks the symmetry between iterations and prevents short cycles, fixed points, and slide-like behaviour [23]. Without the counter, all s steps of W apply the same function g, so that $\widetilde{W} = g^{6m}$, where g maps $(A; R_2, \ldots, R_n)$ to $(\mathrm{MSB}_{64}(x); R_3, \ldots, R_n, \mathrm{LSB}_{64}(x))$ with $x = \mathrm{CIPH}_K(A \,\|\, R_2)$. A state is a fixed point of $\widetilde{W} = g^{6m}$ if and only if it lies on a cycle of g whose length k divides $6m$. A cycle of length k is

described by k accumulator values $A^0, \ldots, A^{k-1}$ and k register values $B_0, \ldots, B_{k-1}$ (extended periodically, $B_{j+k} = B_j$), subject to the k block equations

$$\mathrm{CIPH}_K\big(A^j \,\|\, B_j\big) = A^{j+1} \,\|\, B_{j+m}, \qquad j = 0, \ldots, k-1 \text{ (indices mod } k) \,. \tag{10}$$

Proposition 2. *Model* CIPH_K *as a uniformly random permutation on 128-bit blocks, and let* $N = 2^{128}$*. For* $k \ll 2^{64}$*, the expected number of states on cycles of* g *of length exactly* k *is* $1 + O(k^2 \cdot 2^{-64})$*. Consequently,*

$$\mathbb{E}\big[\#\{S : \widetilde{W}(S) = S\}\big] \;=\; \sigma_0(6m) + O\big(m^2\, \sigma_0(6m) \cdot 2^{-64}\big) \,, \tag{11}$$

where $\sigma_0(x)$ *is the number of divisors of the integer* x*.*

Proof. A state S determines its trajectory under g, and hence a sequence $(A^j, B_j)_{0 \le j < k}$. Conversely, S lies on a cycle of length exactly k if and only if this sequence satisfies (10) and S does not return earlier. We therefore count sequences that satisfy (10). Write $u_j = A^j \,\|\, B_j$ for the block-cipher inputs and $o_j = A^{j+1} \,\|\, B_{j+m}$ for the required outputs.

Distinct inputs. Consider the sequences in which $u_0, \ldots, u_{k-1}$ are distinct and $o_0, \ldots, o_{k-1}$ are distinct. There are $N^k\big(1 - O(k^2/N)\big)$ such sequences, since the $2k$ semiblocks can be chosen freely. For each, a uniformly random permutation satisfies the k equations with probability $1/\big(N(N-1)\cdots(N-k+1)\big) = N^{-k}\big(1 + O(k^2/N)\big)$. Since the inputs are distinct, the state does not return earlier. The expected number of such states is therefore $1 + O(k^2/N)$.

Repeated inputs. If $u_i = u_j$ for some $i \neq j$, the sequence satisfies two semiblock equalities, which reduces the number of sequences by a factor N, while the number of distinct constraints on the permutation decreases by one, which increases the probability by a factor N. Consistency further requires $o_i = o_j$. The equality $A^{i+1} = A^{j+1}$ is not implied by $u_i = u_j$ unless the state returns earlier, so it costs a further factor 2^{-64}. Each pair (i, j) therefore contributes $O(2^{-64})$, and the union bound over the $O(k^2)$ pairs gives $O(k^2 \cdot 2^{-64})$.

Finally, S is a fixed point of $\widetilde{W}$ if and only if its cycle length divides $6m$. Summing over the $\sigma_0(6m)$ divisors $k \le 6m$ gives (11). □

For the common cases of wrapping AES-128, AES-192, and AES-256 keys with KW ($m = 2, 3, 4$), Proposition 2 gives $\sigma_0(12) = \sigma_0(18) = 6$ and $\sigma_0(24) = 8$ fixed points, compared with 1 for a random permutation. The step counter breaks this structure: with it, the s steps of W apply different functions and W has, as expected, about one fixed point. We suggest that the revision motivates the counter along these lines.

Fixed points from cycles of the block cipher. A special case of (10) is a cycle of CIPH_K itself of length $k \in \{1, 2, 3, 6\}$, when $m \equiv 1 \pmod{k}$. For example, if $X \,\|\, Y$ is a fixed point of AES, then $X \,\|\, Y \,\|\, Y$, $X \,\|\, Y \,\|\, Y \,\|\, Y$, … are all fixed points of $\widetilde{W}$, and if $X \,\|\, Y$ and $Z \,\|\, V$ form a cycle of length 2 of AES, then $X \,\|\, Y \,\|\, V \,\|\, Y$ and $Z \,\|\, V \,\|\, Y \,\|\, V$ (and their extensions with odd register length) are fixed points of $\widetilde{W}$. These are the fixed points described in our comments to NIST [1]; they are included in (11).

7 The Single-Semiblock Case of KWP

KWP handles plaintexts of at most 8 octets differently: the ciphertext is

$$\mathrm{CIPH}_K(\mathrm{ICV}_2 \,\|\, [\ell]_{32} \,\|\, P \,\|\, 0^{8 \cdot padlen}), \qquad padlen = 8 - \ell \,, \tag{12}$$

where $\ell \le 8$. This is a single block-cipher call under the *same* key that is used in W for longer plaintexts. Conceptually this resembles using one key in two different modes of operation. Different cryptographic functions should use independent keys or derived subkeys; similar key reuse has led to vulnerabilities in the past, for example when the same key is used for CBC encryption and CBC-MAC, or for CMAC and a key check value [25].

A trivial distinguisher. Since single-semiblock plaintexts are encrypted with a permutation, an adversary that wraps q distinct 8-octet plaintexts never observes a ciphertext collision, whereas an ideal encryption oracle that returns random strings [14] would produce one with probability $\approx 1-e^{-q^2/2^{129}}$. With all $q = 2^{64}$ eight-octet plaintexts the advantage is ≈ 0.39. This is the familiar PRP/PRF switching gap, which affects any scheme with 128-bit ciphertexts, but it shows directly that, by supporting such short plaintexts, KWP cannot have beyond-birthday-bound security. We suggest that the revision mentions this.

Interaction with the last step of W. Given a valid ciphertext $C = C_1 \,\|\, \cdots \,\|\, C_n$, $n \geq 3$, the last step of W gives

$$\mathrm{CIPH}_K\big(A^{s-1} \,\|\, R_2^{s-1}\big) = (C_1 \oplus [s]_{64}) \,\|\, C_n \;. \tag{13}$$

The adversary can query the decryption of the single-block ciphertext $(C_1 \oplus [s]_{64}) \,\|\, C_n$. If it is accepted, the internal value $A^{s-1} \,\|\, R_2^{s-1}$ is, with high probability, $\mathrm{ICV}_2 \,\|\, [8]_{32} \,\|\, Y$ for some Y. The adversary can then query the encryption of an 8-octet plaintext $Y' \neq Y$, obtaining $D_1 \,\|\, D_2$. The ciphertexts C and $C' = (D_1 \oplus [s]_{64}) \,\|\, C_2 \,\|\, \cdots \,\|\, C_{n-1} \,\|\, D_2$ now have a collision $A^{s-1} = A'^{s-1} = \mathrm{ICV}_2 \,\|\, [8]_{32}$, so only five further collisions are needed, and the probability that C' is a successful forgery is

$$P_{\mathrm{F}} \;\approx\; 2^{-64} + \frac{1}{5!}\left(\frac{m}{2^{64}}\right)^5 \;. \tag{14}$$

This is a theoretical observation rather than an improved attack: the first decryption query is itself accepted only with probability $\approx 2^{-64}$, so the overall success probability is far below that of generic forgery. It illustrates the interaction between the single-block path and W. With the KWP limit of $m \leq 2^{29}$ semiblocks, the additional term is $\approx 2^{-181.9}$, and KWP provides reforgeability resistance. Had KWP used the KW limit of $m < 2^{54}$, the additional term would have been $\approx 2^{-56.9}$, i.e., larger than 2^{-64}, and KWP would have been significantly vulnerable to reforgeability. The 2^{32}-octet plaintext limit of KWP is therefore well justified.

Recommendation. We believe KW made the right choice in disallowing single-semiblock plaintexts. We suggest raising the minimum KWP plaintext length to 9 octets. In practice, the use of 1–8 octet plaintexts is likely negligible or non-existent since single DES was disallowed. Alternatively, the single-block path should use a derived subkey.

8 Number of Passes and Length Limits

Rogaway and Shrimpton [14] observe that W uses roughly twelve block-cipher calls per 128-bit block of plaintext, six times more than simpler two-pass schemes. Besides Kelsey's observation, there is no published justification for six passes. Since each pass requires one collision in Kelsey's attack, fewer passes combined with a stricter length limit might give a comparable forgery level at lower cost. Consider a generalisation of KW with an ICV of c bits, an a-bit accumulator, a register of $(128 - a)$-bit words (semiblocks when $a = 64$), a plaintext limit of 2^λ words, and r passes. Assuming $r \cdot 2^\lambda < 2^a$, so that the step counter fits in the accumulator, and $r \gg 1$, so that the construction provides adequate diffusion, the estimate of Section 4 generalises to

$$P_{\mathrm{F}} \;\lesssim\; 2^{-c} + \frac{2^{-r(a-\lambda)}}{r!} \;. \tag{15}$$

For KW ($c = a = 64$, $\lambda = 54$, $r = 6$) this gives $2^{-63.97}$. With $r = 3$ passes and $\lambda = 40$ one obtains $P_{\mathrm{F}} \lesssim 2^{-64} + 2^{-72}/3! \approx 2^{-64.00}$, i.e., possibly a comparable forgery probability with respect to Kelsey's attack at half the cost, while still allowing plaintexts of 2^{40} semiblocks (8 TiB), far beyond any key-wrapping use case. The formula also illustrates that the accumulator, not

the block size, determines the security against long-message attacks. This parametrisation is a heuristic based on Kelsey's attack and is not a security proof.

9 Implementation and Deployment Guidance

In-place implementation. We suggest that the revision references RFC 3394 [5] as a complementary specification. Its index-based alternative description is easier to implement in software and can be implemented in place, without additional memory allocations, which is valuable on constrained IoT devices. NIST should explicitly highlight this property.

Zeroization and release of unverified plaintext. Implementations should zeroize the intermediate variables A and R before W and W^{-1} return; note that compilers may optimise away naive `memset` calls. Moreover, no part of the output of W^{-1} should be released before all ICV and padding checks have succeeded, and on failure the output should be zeroized before FAIL is returned. The pseudocode of KWP-AD (Algorithm 6 in [3]) returns at different points depending on whether the ICV check or the padding check fails, which may leak this distinction via timing. Since W^{-1} behaves as a wide-block cipher, this information is unlikely to help an adversary, but the revision should state whether it is acceptable or whether constant-time error handling is required.

Length leakage and context binding. Without additional padding, KW reveals the length of the wrapped key, which may let an adversary focus on the shortest keys or wrapped passwords. Applications can mitigate this by adding variable-length padding at the application layer before wrapping. Similarly, context binding (key usage, algorithm identifier, owner) can be achieved by appending application-specific context information to the plaintext, since KW and KWP lack associated data. Both measures also add redundancy that the receiver can verify, reducing the forgery probability in the same way as the length check in Proposition 1.

Replay protection. Replay protection is an essential security property. GCM and CCM with sequential nonces support protocol-level replay protection, whereas KW uses no nonce and must be augmented with a separate mechanism. The wrapping function W and the encode-then-encipher paradigm can provide not only integrity but also AERO-style replay protection [16]: by including a sequence number in the enciphered string, a receiver can check freshness with lower overhead than a separate nonce and tag. We suggest that the revision adds a paragraph on replay protection.

Wider blocks. Depending on the timing of NIST's Acc256 accordion and its derived functions [17, 18], NIST could consider allowing 256-bit block ciphers such as Rijndael-256 with 128-bit semiblocks. This appears conceptually straightforward; indeed, the current document would be cleaner if W and TW were specified as one function with the block size as a parameter. As Sections 5 and 8 show, the 128-bit block size of AES, through the 64-bit accumulator, is the limiting factor for KW, as for most block-cipher modes. Given the expected advantages of accordion-based constructions, we believe that derived functions based on accordions should eventually supersede both AES-KW and AES-SIV.

10 Conclusions

AES Key Wrap is a conservative and, within its limits, robust design whose concrete security is better than its 64-bit ICV suggests, especially when receivers enforce expected lengths. At the same time, the analysis in SP 800-38F understates the effect of adaptive attacks, contains

a few inaccuracies, and suggests properties, such as beyond-birthday-bound security, that the construction does not have. The revision of SP 800-38F is an opportunity to correct these points, to improve the guidance on implementation and deployment, and to set clearer goals for the next generation of key-wrapping algorithms: performance, misuse-resistant authenticated encryption (MRAE), IND-CCA2, support for associated data, variable-length padding, lower forgery probability, commitment, formal security proofs, and AERO-style replay protection.

Acknowledgements. This paper is an updated, expanded, and refined version of comments submitted to NIST by Ericsson [1]. We thank NIST for its continued efforts to produce well-written, user-friendly, and open-access security documents.

References

[1] NIST. Pre-Draft Public Comments on Proposed Revision of SP 800-38F, comment period May 5 – July 10, 2026. Including Ericsson (J. Preuß Mattsson), Comments on SP 800-38F (Methods for Key Wrapping), June 2, 2026, pp. 4–14. https://csrc.nist.gov/files/pubs/sp/800/38/f/r1/ipd/docs/sp-800-38Fr1-comments-rec-updated-090126.pdf

[2] NIST. Pre-Draft Call for Comments: Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping (SP 800-38F Rev. 1), May 2026. https://csrc.nist.gov/pubs/sp/800/38/f/r1/ipd

[3] M. Dworkin. Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping. NIST SP 800-38F, December 2012. https://doi.org/10.6028/NIST.SP.800-38F

[4] NIST. AES Key Wrap Specification, 2001. Archived at https://web.archive.org/web/20041031122517/http://www.csrc.nist.gov/CryptoToolkit/kms/key-wrap.pdf

[5] J. Schaad and R. Housley. Advanced Encryption Standard (AES) Key Wrap Algorithm. RFC 3394, September 2002. https://www.rfc-editor.org/rfc/rfc3394

[6] R. Housley and M. Dworkin. Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm. RFC 5649, August 2009. https://www.rfc-editor.org/rfc/rfc5649

[7] J. Schaad. Use of the Advanced Encryption Standard (AES) Encryption Algorithm in Cryptographic Message Syntax (CMS). RFC 3565, July 2003. https://www.rfc-editor.org/rfc/rfc3565

[8] M. Jones. JSON Web Algorithms (JWA). RFC 7518, May 2015. https://www.rfc-editor.org/rfc/rfc7518

[9] OASIS. PKCS #11 Specification Version 3.1. https://docs.oasis-open.org/pkcs11/pkcs11-spec/v3.1/pkcs11-spec-v3.1.html

[10] OASIS. Key Management Interoperability Protocol Specification Version 2.1. https://docs.oasis-open.org/kmip/kmip-spec/v2.1/kmip-spec-v2.1.html

[11] NIST. Public Comments on Draft SP 800-38F, 2011. https://csrc.nist.gov/CSRC/media/Projects/Block-Cipher-Techniques/documents/BCM/Comments/800-38-series-drafts/KeyWrap/CommentsSP800-38F.pdf

[12] J. Black and M. Cochran. MAC Reforgeability. FSE 2009, LNCS 5665, Springer, 2009. https://doi.org/10.1007/978-3-642-03317-9_21

[13] M. Dworkin. Request for Review of Key Wrap Algorithms. Cryptology ePrint Archive, Paper 2004/340. https://eprint.iacr.org/2004/340

[14] P. Rogaway and T. Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. EUROCRYPT 2006, LNCS 4004, pp. 373–390. Full version: Deterministic Authenticated-Encryption, Cryptology ePrint Archive, Paper 2006/221. https://eprint.iacr.org/2006/221

[15] M. Bellare and P. Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. ASIACRYPT 2000, LNCS 1976, pp. 317–330. https://doi.org/10.1007/3-540-44448-3_24

[16] D. McGrew. Authenticated Encryption with Replay prOtection (AERO). IETF Internet-Draft draft-mcgrew-aero. https://datatracker.ietf.org/doc/html/draft-mcgrew-aero

[17] NIST. Requirements for Cryptographic Accordions. NIST IR 8552, 2025. https://nvlpubs.nist.gov/nistpubs/ir/2025/NIST.IR.8552.pdf

[18] NIST. NIST Launches Development of Cryptographic Accordions (SP 800-197A pre-draft). https://csrc.nist.gov/pubs/sp/800/197/a/iprd

[19] A. Joux. Authentication Failures in NIST version of GCM. Comments to NIST, 2006. https://csrc.nist.gov/csrc/media/projects/block-cipher-techniques/documents/bcm/comments/800-38-series-drafts/gcm/joux_comments.pdf

[20] N. Ferguson. Authentication weaknesses in GCM. Comments to NIST, 2005. https://csrc.nist.gov/CSRC/media/projects/Block-Cipher-Techniques/documents/BCM/comments/CWC-GCM/Ferguson2.pdf

[21] T. Iwata, K. Ohashi, and K. Minematsu. Breaking and Repairing GCM Security Proofs. CRYPTO 2012, LNCS 7417, pp. 31–49. https://eprint.iacr.org/2012/438

[22] J. Jonsson. On the Security of CTR + CBC-MAC. SAC 2002, LNCS 2595, pp. 76–93. https://doi.org/10.1007/3-540-36492-7_7

[23] A. Biryukov and D. Wagner. Slide Attacks. FSE 1999, LNCS 1636, pp. 245–259. https://doi.org/10.1007/3-540-48519-8_18

[24] A. Becker and M. Jenkins. Commercial National Security Algorithm (CNSA) Suite 2.0 Profile for Secure/Multipurpose Internet Mail Extensions (S/MIME). IETF Internet-Draft draft-becker-cnsa2-smime-profile, 2026. https://datatracker.ietf.org/doc/draft-becker-cnsa2-smime-profile/

[25] Ericsson. Comments on SP 800-38B, The CMAC Mode for Authentication, 2024. https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/initial-comments/sp800-38b-initial-public-comments-2024.pdf