The Deimos AEAD is insecure

JP Aumasson 28 August 2026

The Deimos AEAD authenticates the encrypted data but not its nonce. Replacing the nonce in any valid ciphertext by any other nonce yields a distinct valid ciphertext, defeating authenticity.

Construction

For password p, salt s, nonce N, and plaintext M, Deimos [1] computes

$$\begin{aligned}(K_1, K_2, K_3) &= \text{HKDF-BLAKE2b}(p, s),\\ Y &= M \oplus \text{XChaCha20}_{K_1}(N),\\ T &= \text{HMAC-SHA256}_{K_3}(Y),\\ X &= s \parallel N \parallel Y \parallel T.\end{aligned}$$

The decryptor first verifies T from Y alone. Only after accepting the tag does it use the supplied nonce N to decrypt. Algorithms 1–2 and the reference C++ implementation follow this data flow [2].

Forgery

Let

$$X = s \parallel N \parallel Y \parallel T$$

be any valid ciphertext. Choose any other 192-bit nonce $N' \neq N$ and submit

$$X' = s \parallel N' \parallel Y \parallel T.$$

The salt is unchanged, so the same K_3 is derived. Since Y and T are also unchanged, verification still succeeds:

$$\text{HMAC-SHA256}_{K_3}(Y) = T.$$

The decryptor then uses N' and returns

$$\text{Dec}_p(X') = Y \oplus \text{XChaCha20}_{K_1}(N') \neq \perp.$$

Therefore $X' \neq X$ is a valid forgery and

$$\Pr[\text{Dec}_p(X') \neq \perp] = 1.$$

This is a one-query ciphertext-integrity forgery. It falsifies the paper's claim that "HMAC-SHA256 prevents undetected modifications." Deimos, as specified and implemented, is therefore insecure.

AI disclosure. GPT-5.6 Sol (Daybreak Blue) selected the target, found the attack, and drafted the initial note. Claude Sonnet 5 independently checked the result against Algorithms 1–2 and the reference C++ implementation. The human reviewed the analysis and edited the paper; he takes responsibility for any remaining error.

References. [1] M. Belam, "Deimos Cipher: A High-Entropy, Secure Encryption Algorithm with Strong Diffusion and Key Sensitivity," IACR ePrint 2025/414, 2025. [2] M. Belam, *Deimos-Cipher*, reference C++ implementation, accessed 28 August 2026.