Network Working Group
Internet-Draft
Intended status: Experimental
Expires: 22 June 2026

A. Maurette
IUT R&T Bethune

19 December 2025

HMTFTP - HMAC-based Trivial File Transfer Protocol (v0.1)
draft-maurette-hmtftp-00

Abstract

HMTFTP ([RFC1350]) is a lightweight UDP file transfer protocol that preserves TFTP-style simplicity (block-and-ACK) while adding an optional authenticated encryption mode using AEAD AES-256-GCM with pre-shared keys derived via HKDF-SHA-256. The protocol targets managed environments (e.g., CPEs, OpenWrt, embedded devices) where provisioning and configuration have constrained requirements. This document specifies v0.1, including message formats, negotiation, cryptographic processing, timers, and security considerations.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 22 June 2026.

Copyright Notice

Table of Contents

1. Introduction

TFTP ([RFC1350]) is extremely simple but offers no security properties. HMTFTP v0.1 keeps UDP, a fixed header with TLVs, and a block-and-ACK transfer model, while introducing an optional AEAD protection based on a pre-shared key (PSK). The intended scope covers LAN or controlled networks, device provisioning, and small images/configurations.

HMTFTP aims to remain comprehensible and implementable on constrained systems, reusing common cryptographic primitives and clear operational guidance. It is not meant to compete with secure, general-purpose transports such as TLS or QUIC ([RFC9000]), but to provide a small and deterministic surface fit for specific operational niches.

2. Conventions and Terminology

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, NOT RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in [RFC2119] and [RFC8174] when, and only when, they appear in all capitals.

Terminology: PSK - pre-shared key; AEAD - Authenticated Encryption with Associated Data; AAD - Additional Authenticated Data; CSPRNG - cryptographically secure pseudorandom number generator; PMTU - Path Maximum Transmission Unit.

3. Protocol Overview

HMTFTP exchanges begin with a stateless discovery and capability exchange, followed by a transfer request and a TFTP-like data/acknowledgment loop. When encryption is negotiated, all DATA payloads are protected with AES-256-GCM and the header and TLVs are authenticated as AAD.

Flow (happy path): HELLO -> HELLO_ACK; NEGOTIATE -> NEGOTIATE_ACK; XFER_REQ (GET or PUT) -> DATA/ACK ... -> EOF.

Opcodes: HELLO(0), HELLO_ACK(1), NEGOTIATE(2), NEGOTIATE_ACK(3), DATA(4), ACK(5), ERROR(6 - reserved codes), XFER_REQ(7).

TLVs (Type:Length): CNONCE(0x08,16), SNONCE(0x09,16), BLKSIZE(0x10,2), FNAME(0x20,var), MODE(0x21,1 - 1=PUT, 2=GET), ENC(0x22,1 - 1=request), CIPHER(0x23,1 - 1=AES-256-GCM). Additional TLVs MAY be defined; unknown TLVs MUST be ignored if the critical bit is not set.

Default UDP port: user-configured (examples use 49696). BLKSIZE range: 64..4096 octets (default 512). EOF is signaled by a DATA block strictly shorter than BLKSIZE.

4. Wire Format

All multi-octet fields use network byte order (big-endian). The fixed header is 24 octets:

Magic (16 bits) 0x484D ("HM").

Ver (4) | Op (4) Protocol version and opcode.

Flags (8) Bitmask; see below.

Reserved (8) MUST be zero.

Pad (8) MUST be zero.

SessionID (32) Chosen by the initiator.

Seq (32) Monotonically increasing data sequence starting at 0.

Ack (32) Last contiguous block received.

HdrLen (16) Length in octets of the TLV area following the header.

PayLen (16) Length in octets of the payload (ciphertext when ENC=1).

Reserved2 (16) MUST be zero.

Flags: 0x01 = TLVs present; 0x02 = ENCRYPTED. When ENCRYPTED=1, the GCM tag (16 octets) follows the payload.

AAD: the AEAD AAD MUST cover the entire 24-octet fixed header, and in v0.1 also the TLV area (HdrLen octets).

4.1. TLV Encoding

TLVs are encoded as Type(1), Length(1), Value(Length). Types are unsigned 8-bit; Length is unsigned 8-bit. Multi-octet values within TLVs are big-endian.

5. Transport and PMTU

HMTFTP runs over UDP/IPv4 (and MAY run over UDP/IPv6). The internal maximum datagram size is 4096 octets. To avoid IP fragmentation, BLKSIZE SHOULD be selected such that IP header + UDP header + HMTFTP header + TLVs + payload (+ tag if encrypted) remain within the PMTU.

Implementations SHOULD follow [RFC8085] for UDP usage, and use [RFC1191] (IPv4) and [RFC8201] (IPv6) PMTU discovery or [RFC4821] PLPMTUD. Senders SHOULD react to ICMP Packet Too Big or retransmission timeouts by reducing BLKSIZE.

Retransmissions: a sender maintains an RTO with exponential backoff. ACKs carry the last in-order block number; selective ACKs are out of scope for v0.1.

6. Cryptographic Processing

Material and inputs:

1. PSK: 32 octets generated with at least 128 bits of entropy (preferably 256), stored with OS permissions 0400/0600.

2. CNONCE and SNONCE: 16 octets each from a CSPRNG.

3. HKDF-SHA-256 ([RFC5869]): IKM = PSK; salt = CNONCE || SNONCE (32 octets); info = "hmtftp v1 keys". Output: key (32) and iv_base (12). v0.1 uses iv_base[0..7] (8 octets) to build the IV.

4. GCM IV construction: IV = iv_base[0..7] || seq_be32, totaling 12 octets. For a given key, key/IV pairs MUST be unique. Before seq wraps (2^32), a new session SHOULD be negotiated well in advance (for example at 2^24 blocks).

5. AAD and tag: AAD is the 24-octet header and the TLV area (HdrLen). The GCM tag is 16 octets appended after the payload.

6. Retransmissions: a given Seq MUST retransmit the exact same ciphertext and tag; an endpoint MUST NOT re-encrypt a modified payload under the same (key, IV).

7. On AEAD failure: silently drop (no ERROR, no ACK) to avoid validity oracles.

6.1. Cipher Negotiation

The CIPHER TLV selects AEAD_AES_256_GCM when ENC is requested, as specified in [RFC5116]. Future ciphersuites MAY be registered; endpoints MUST ignore unknown ciphers when ENC is not requested.

7. Transfer Procedure

XFER_REQ includes MODE (PUT or GET), FNAME, and optionally BLKSIZE, ENC, and CIPHER. The server replies with ACK(0) to confirm parameters or ERROR.

Data blocks are numbered from 0. Each DATA carries the current Seq (block number) and PayLen up to BLKSIZE. The receiver sends ACK with Ack set to the highest contiguous block received. A block shorter than BLKSIZE indicates EOF.

ERROR codes (non-exhaustive): 0x01 = Unsupported TLV; 0x02 = Invalid parameter; 0x03 = Access denied; 0x04 = Not found; 0x05 = Integrity failure; 0x06 = Internal error.

8. Operational Considerations

Logging: Implementations SHOULD avoid logging keys, PSKs, and nonces. Logging of session identifiers, aggregate sizes, timing, negotiated parameters, and ciphersuite is RECOMMENDED.

PSK lifecycle: PSKs MUST NOT be embedded in images or world-readable. Rotation policies SHOULD be in place; devices SHOULD provide out-of-band key provisioning.

Amplification: Servers SHOULD cap the size of unauthenticated responses and MAY require a cookie mechanism for untrusted networks in future versions.

9. Security Considerations

Threat model includes passive observers (confidentiality), on-path adversaries (integrity, authenticity), off-path spoofers, DoS, and reflection/amplification. See [RFC3552] and [RFC4732].

No PFS: v0.1 uses only PSK and does not provide forward secrecy. A future extension may add PSK + ECDH.

Replay and duplicates: receivers SHOULD de-duplicate on (SessionID, Seq). Anti-replay windows MAY be used; session lifetimes SHOULD be bounded.

Downgrade: endpoints SHOULD reject negotiation that drops from ENC=1 to ENC=0 after initial agreement. A future KEY_CONFIRM message can strengthen this.

Limits: Implementations MUST enforce AEAD usage limits and IV uniqueness. Resource caps SHOULD mitigate DoS. See also [RFC8452] for AEAD usage limits.

10. IANA Considerations

This document has no IANA actions.

See [RFC8126] for general IANA policy guidance if future code points are requested.

11. Appendix A: Informative Test Vectors

Example (hex):

* PSK = 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
* CNONCE = 00112233445566778899aabbccddeeff
* SNONCE = 0102030405060708090a0b0c0d0e0f10
* HKDF-salt = CNONCE || SNONCE
* info = "hmtftp v1 keys"
* Derived key (32) and iv_base (12) - implementation-defined output for this example.

12. Acknowledgments

The author thanks reviewers and operators for early feedback and implementation guidance.

13. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, DOI 10.17487/RFC5869, May 2010, <https://www.rfc-editor.org/info/rfc5869>.

[RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", RFC 5116, DOI 10.17487/RFC5116, January 2008, <https://www.rfc-editor.org/info/rfc5116>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <https://www.rfc-editor.org/info/rfc1191>.

[RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <https://www.rfc-editor.org/info/rfc8201>.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <https://www.rfc-editor.org/info/rfc4821>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

14. Informative References

[RFC1350] Sollins, K., "The TFTP Protocol (Revision 2)", STD 33, RFC 1350, DOI 10.17487/RFC1350, July 1992, <https://www.rfc-editor.org/info/rfc1350>.

[RFC4732] Handley, M., Ed., Rescorla, E., Ed., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, DOI 10.17487/RFC4732, December 2006, <https://www.rfc-editor.org/info/rfc4732>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[RFC8452] Gueron, S., Langley, A., and Y. Lindell, "AES-GCM-SIV: Nonce Misuse-Resistant Authenticated Encryption", RFC 8452, DOI 10.17487/RFC8452, April 2019, <https://www.rfc-editor.org/info/rfc8452>.

Author’s Address

A. Maurette
IUT R&T Bethune
Email: contact@c4tz.fr