Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 26, 2026

P. Gupta
MobileStack

April 24, 2026

EAP-WSIM: A SIM-Based EAP Method Using the MILENAGE-ECDH-FWD Authentication Construction

draft-gupta-emu-eap-wsim-00

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-gupta-emu-eap-wsim/.

Discussion of this document takes place on the EAP Method Update Working Group mailing list (mailto:emu@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/emu/. Subscribe at https://www.ietf.org/mailman/listinfo/emu/.

Source for this draft and an issue tracker may be found at https://github.com/pgupta-mobilestack/eap-wsim (to be created upon WG adoption).

Abstract

This document specifies EAP-WSIM, a new EAP authentication method for enterprise wireless networks that performs offline mutual authentication without contacting any Mobile Network Operator (MNO) Home Subscriber Server (HSS), Home Location Register (HLR), or Unified Data Management (UDM) function during the authentication exchange.

EAP-WSIM uses a new cryptographic construction called MILENAGE-ECDH-FWD, which combines the MILENAGE authentication and key agreement algorithm [TS35.205] with an ephemeral Elliptic Curve Diffie-Hellman (ECDH) exchange. This construction simultaneously provides mutual authentication via MILENAGE challenge-response and forward secrecy via ephemeral ECDH, such that compromise of long-term key material does not expose keys from prior sessions.

The key distinction from all existing SIM-based EAP methods -- EAP-SIM [RFC4186], EAP-AKA [RFC4187], EAP-AKA' [RFC9048], and EAP-AKA' FS [RFC9678] -- is that EAP-WSIM requires no MNO backend contact during authentication. The EAP server holds a SIM card (the WSIM) that functions as a self-contained Authentication Centre, enabling deployment in air-gapped enterprise environments without MNO infrastructure dependency.

This document specifies the protocol core: the four-round EAP-WSIM exchange, message formats, attribute encodings, and key derivation. Key slot selection [WSIM-KEY-SELECT] and IEEE 802.11r Fast Transition integration [WSIM-FT] will be submitted as additional specifications only if required based on review and comments on this document.

This document does not update any existing RFC.

Status of This Memo

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

This Internet-Draft will expire on October 26, 2026.

Copyright Notice

IPR Disclosure

The author has filed US Provisional Patent Application 64/048,069 covering aspects of the WSIM architecture described in this document. In accordance with BCP 79 [RFC8179], the author commits to license any patents reading on this specification to implementers of any RFC produced from this document on royalty-free, reasonable, and non-discriminatory terms (RAND-z). An IPR disclosure will be filed at https://datatracker.ietf.org/ipr/about/ concurrent with submission of this Internet-Draft.

Note on RFCs That Could Be Updated

This document defines a new, independent EAP method. It does not update any existing RFC. Specifically:

- RFC 4186 (EAP-SIM), RFC 4187 (EAP-AKA), RFC 9048 (EAP-AKA'), and RFC 9678 (EAP-AKA' FS) are informative references only. EAP-WSIM does not modify the operation of any of these methods.
- RFC 3748 (EAP) is the framework within which EAP-WSIM operates. No changes to the base EAP framework are required or proposed.
- RFC 5247 (EAP Key Management Framework) defines the MSK/EMSK model that EAP-WSIM follows. No changes to RFC 5247 are required.

Should the IETF determine that EAP-WSIM warrants an update to IANA registries under RFC 3748, a formal "Updates: 3748" header would be added prior to publication. This is an open question for WG discussion.

Table of Contents

1. Introduction
2. Terminology
3. Motivation and Design Goals
4. The MILENAGE-ECDH-FWD Construction
    4.1. Overview
    4.2. Inputs
    4.3. Construction Steps
    4.4. Output Key Material
    4.5. Security Properties

5. EAP-WSIM Protocol
    5.1. Protocol Overview
    5.2. Message Header
    5.3. Attribute Encoding
    5.4. WSIM-Start (Subtype 0x01)
    5.5. WSIM-Challenge (Subtype 0x02)
    5.6. WSIM-Confirm (Subtype 0x03)
    5.7. WSIM-Complete (Subtype 0x04)
    5.8. WSIM-Error (Subtype 0x05)
    5.9. Anti-Replay
6. MSK and EMSK
7. Relationship to RFC 9678 (EAP-AKA' FS)
    7.1. MNO Backend Dependency
    7.2. Protocol Architecture
    7.3. Key Derivation
    7.4. Pre-Session Integrity
    7.5. Summary Comparison Table
8. Security Claims (per RFC 3748 Section 7.2)
9. IANA Considerations
10. Security Considerations
11. Privacy Considerations
12. References
    12.1. Normative References
    12.2. Informative References
Appendix A. Test Vectors (MILENAGE-ECDH-FWD)
Appendix B. Comparison with Prior EAP SIM-Based Methods
Appendix C. Future Work
Author's Address

1. Introduction

EAP-SIM [RFC4186] and EAP-AKA [RFC4187] authenticate subscribers using SIM credentials, but require the EAP server to contact the MNO AuC, HSS, HLR, or UDM during each exchange to obtain fresh authentication vectors. EAP-AKA' [RFC9048] improved the key derivation over EAP-AKA but retains this backend dependency. RFC 9678 [RFC9678] added forward secrecy via ECDH to EAP-AKA', but likewise still requires MNO backend contact for each authentication.

EAP-WSIM removes this dependency entirely. The EAP server holds a SIM card -- the WSIM -- that contains pre-provisioned master key material and can generate MILENAGE authentication vectors entirely on-card, functioning as a self-contained Authentication Centre. The subscriber's UE SIM holds device-specific keys pre-derived from the same root material by an offline trust anchor. No MNO backend contact occurs at any point during the authentication exchange.

The cryptographic core of EAP-WSIM is a new named construction, MILENAGE-ECDH-FWD, which integrates MILENAGE [TS35.205] with ephemeral ECDH key agreement. MILENAGE provides mutual authentication; raw ECDH provides key agreement; MILENAGE-ECDH-FWD binds both such that session key material is cryptographically bound to both the MILENAGE authentication outcome and the ephemeral ECDH exchange. This provides mutual authentication and forward secrecy simultaneously.

By defining MILENAGE-ECDH-FWD as a named construction rather than an ad-hoc combination, this document enables independent security analysis and allows future protocols to reuse it without re-specifying the combination.

Section 7 provides a detailed technical comparison with RFC 9678 to precisely characterize the non-overlap between EAP-WSIM and EAP-AKA' FS.

This document does not update any existing RFC. It defines a new, independent EAP method.

2. 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 BCP 14 [RFC2119] [RFC8174].

WSIM:
: Wireless SIM. A SIM card or eSIM deployed on the network/authenticator side, holding master key material and executing MILENAGE on-card. The WSIM functions as a self-contained Authentication Centre without requiring MNO contact.

UE SIM:
: The SIM card in the subscriber's User Equipment, pre-provisioned at manufacturing with K_UE: a device-specific MILENAGE key derived from the master key material using the subscriber's IMSI. The Peer does NOT hold the root master key; it holds K_UE only.

K_SLOT[i]:
: One of 15 independent 256-bit master keys (i = 0..14) held in the WSIM card. K_SLOT[i] is NEVER exported from the WSIM. Each slot has an associated status: ACTIVE (0x01), REVOKED (0x00), or RESERVE (0x02, slot 14 only). Revoked slots are skipped by WSIM-SLOT-SELECT on both Server and Peer. Slots 0-11 correspond to calendar months; slots 12-13 are overlap periods; slot 14 is the emergency reserve used only when all operational slots are revoked.

K_UE[i]:
: The 128-bit MILENAGE authentication key for subscriber IMSI and slot i. Derived as:

  K_UE[i] = HKDF-SHA-256(K_SLOT[i], salt, IMSI || i || version)

  The Server (WSIM) derives K_UE[i] on-card at authentication time. The Peer (UE SIM) holds K_UE[0..14] for ALL slots, pre-provisioned at manufacturing. When a slot is revoked on the Server side, the Peer automatically uses the next active slot via WSIM-SLOT-SELECT -- no OTA update required.

WSIM-SLOT-SELECT:
: A non-reversible, IMSI-bound, Time-of-Day algorithm executed independently by both Server and Peer to determine the active slot index. Given the same IMSI and UTC hour, both parties deterministically select the same slot without communication. The algorithm computes HMAC-SHA-256(K_SELECT, IMSI||TOD||label), takes (H[0]*256 + H[1]) mod 14 as the base candidate, then walks forward until it finds the first ACTIVE slot. If all operational slots (0-13) are revoked, slot 14 is returned. K_SELECT is stored READ=NEVER on both WSIM and UE SIM.

K:
: Used in this document as shorthand for K_UE[active_slot] in protocol operation contexts, where active_slot is determined by WSIM-SLOT-SELECT.

MILENAGE-ECDH-FWD:
: The cryptographic construction defined in Section 4 combining MILENAGE mutual authentication with ephemeral ECDH key agreement to produce authenticated session key material with forward secrecy.

Server: The WSIM card acting as EAP authenticator.

Peer: The UE SIM acting as EAP supplicant.
MSK: Master Session Key, 64 bytes, per [RFC5247].
EMSK: Extended Master Session Key, 32 bytes, per [RFC5247].

Forward Secrecy:
Compromise of long-term key K after session completion does not expose session keys from that completed session.

Offline Authentication:
Authentication that completes without any network communication to MNO backend infrastructure (HSS, HLR, UDM, or AuC).

3. Motivation and Design Goals

REQ-1 (Offline Authentication): The EAP server MUST authenticate provisioned subscribers without contacting any MNO HSS, HLR, UDM, or AuC during the authentication exchange.

REQ-2 (Mutual Authentication): Both Server and Peer MUST authenticate each other. Server authenticates to Peer via AUTN; Peer authenticates to Server via RES.

REQ-3 (Forward Secrecy): Compromise of long-term key K MUST NOT expose session keys from prior completed sessions.

REQ-4 (Named Construction): The combination of MILENAGE and ECDH MUST be defined as a named, reusable cryptographic construction with independently verifiable security properties.

REQ-5 (MSK Compatibility): The MSK and EMSK produced MUST be compatible with [RFC5247].

REQ-6 (Anti-Replay): The protocol MUST prevent replay of captured exchanges via sequence number and nonce mechanisms.

REQ-7 (Minimal Scope): This document covers the protocol core. Key slot selection and 802.11r integration may be submitted as additional specifications depending on WG review feedback.

4. The MILENAGE-ECDH-FWD Construction

4.1. Overview

MILENAGE-ECDH-FWD is a two-party authenticated key agreement construction. The Server (WSIM) holds K_SLOT[0..14] and selects the active slot using WSIM-SLOT-SELECT, then derives K_UE[slot] on-card. The Peer (UE SIM) holds K_UE[0..14] for all slots and independently selects the same active slot using WSIM-SLOT-SELECT. If a slot has been revoked on the Server side, WSIM-SLOT-SELECT walks forward to the next ACTIVE slot on both sides independently, requiring no OTA update to the Peer.

Two tracks run in parallel:

Track A (MILENAGE): Server computes MILENAGE(K, RAND) to obtain XRES, CK, IK, AK; constructs AUTN. Peer verifies AUTN (authenticating Server) and computes RES, CK, IK. Server verifies RES == XRES (authenticating Peer).

Track B (ECDH): Each party generates a fresh ephemeral P-256 keypair and exchanges public keys. Each computes the same shared secret SS = P256-ECDH(sk_local, pk_remote).

Combination: Session key material derives from (SS, CK, IK,

NONCE_S, NONCE_P) via HKDF-SHA-256, binding both tracks.
Compromise of K alone cannot recompute SS (forward secrecy);
compromise of SS alone cannot derive session keys without CK
and IK (key binding).

4.2. Inputs

slot	Active slot index selected by WSIM-SLOT-SELECT (independently computed by both Server and Peer)
K_UE[slot]	128-bit MILENAGE key for this subscriber and active slot (Server derives on-card; Peer holds pre-provisioned)
RAND	128-bit random challenge (generated fresh by Server)
(sk_S,pk_S)	Server ephemeral P-256 keypair (fresh per session)
(sk_P,pk_P)	Peer ephemeral P-256 keypair (fresh per session)
NONCE_S	128-bit random nonce (Server, fresh per session)
NONCE_P	128-bit random nonce (Peer, fresh per session)
SQN	48-bit MILENAGE sequence number
AMF	16-bit Authentication Management Field

4.3. Construction Steps

```
Server side:
  (a) Receive Peer identity (IMSI or pseudonym) from outer identity
  (b) Run WSIM-SLOT-SELECT(IMSI, UTC_now, K_SELECT, SLOT_STATUS):
        H = HMAC-SHA-256(K_SELECT, IMSI || TOD_block || label)
        slot_base = (H[0]*256 + H[1]) mod 14
        for offset in 0..13:
            idx = (slot_base + offset) mod 14
            if SLOT_STATUS[idx] == ACTIVE: slot = idx; break
        if no active slot found: slot = 14 (emergency reserve)
  (c) Derive K_UE[slot] on-card:
        K_UE[slot] = HKDF-SHA-256(K_SLOT[slot], salt, IMSI||slot||ver)
  (d) Run MILENAGE(K_UE[slot], RAND):
        XRES = f2(K_UE[slot], RAND)
        CK   = f3(K_UE[slot], RAND)
        IK   = f4(K_UE[slot], RAND)
        AK   = f5(K_UE[slot], RAND)
  (e) AUTN = (SQN XOR AK) || AMF || f1(K_UE[slot], SQN, RAND, AMF)
  (f) Generate ephemeral P-256 keypair (sk_S, pk_S)
  (g) Generate NONCE_S
  (h) K_mac_start = HMAC-SHA-256(K_UE[slot], "WSIM-START-MAC-v1" || RAND)
  (i) Send WSIM-Start: RAND, AUTN, pk_S, NONCE_S, AT_COUNTER, AT_MAC
      NOTE: AT_COUNTER carries the active slot index in the high
      byte, enabling the Peer to detect slot selection mismatch
      before expending MILENAGE computation.

Peer side (upon receiving WSIM-Start):
  (j) Run WSIM-SLOT-SELECT(IMSI, UTC_now, K_SELECT, local_status)
      independently -- same algorithm, same inputs, same result.
      If computed slot != slot indicated in AT_COUNTER, retry with
      UTC_now +/- 1 hour to handle clock skew.  If still no match,
      send WSIM-Error 0x0008 (SLOT_MISMATCH).
  (k) Compute K_mac_start = HMAC-SHA-256(K_UE[slot], label || RAND)
  (l) Verify AT_MAC using K_mac_start
  (m) Verify AUTN using K_UE[slot]:
        AK   = f5(K_UE[slot], RAND)
        SQN  = AUTN[0:6] XOR AK
        Recompute MAC_A = f1(K_UE[slot], SQN, RAND, AUTN[6:8])
        Verify MAC_A == AUTN[8:16]        -- Server authenticated
        Verify SQN is in acceptable range
  (n) Compute RES = f2(K_UE[slot], RAND); CK = f3; IK = f4
  (o) Generate ephemeral P-256 keypair (sk_P, pk_P)
  (p) Generate NONCE_P
  (q) Compute SS = P256-ECDH(sk_P, pk_S)
  (r) Derive K_session per Section 4.4
  (s) Send WSIM-Challenge: RES, pk_P, NONCE_P, AT_MAC_PEER
```

```
   Server side (upon receiving WSIM-Challenge):
     (t) Verify RES == XRES                  -- Peer authenticated
     (u) Compute SS = P256-ECDH(sk_S, pk_P)
     (v) Derive same K_session
```

4.4. Output Key Material

```
     IKM  = SS || CK || IK
             (32 + 16 + 16 = 64 bytes total)

     salt = NONCE_S || NONCE_P
             (16 + 16 = 32 bytes)

     info = "MILENAGE-ECDH-FWD-v1"
             (21 bytes, UTF-8, fixed version label)

     L    = 128 bytes

     OKM  = HKDF-SHA-256(IKM, salt, info, L)
```

```
   OKM partitioned as:
     OKM[0:64]    MSK        (64 bytes, per [RFC5247])
     OKM[64:96]   EMSK       (32 bytes, per [RFC5247])
     OKM[96:112]  K_auth     (16 bytes, MAC key for WSIM-Challenge)
     OKM[112:128] K_confirm  (16 bytes, MAC key for WSIM-Confirm)
```

4.5. Security Properties

Mutual Authentication: Server authenticates to Peer via AUTN verification (step h). Peer authenticates to Server via RES == XRES (step o).

Forward Secrecy: Ephemeral private keys sk_S and sk_P are discarded after use. An adversary who later learns K_UE[slot] cannot recompute SS from recorded public keys pk_S, pk_P. Since SS contributes to IKM, OKM is computationally independent of K_UE[slot] alone.

Key Binding: IKM includes SS (ECDH) and CK||IK (MILENAGE). Neither component alone determines OKM. A failure in one component does not automatically yield the session keys.

IMSI-Bound Key Isolation: Each subscriber’s K_UE[i] is independently derived from K_SLOT[i] using the subscriber’s IMSI. Compromise of one subscriber’s K_UE[i] does not affect other subscribers or other slots. K_SLOT[0..14] never leaves the WSIM card.

Slot Revocation Without Re-Provisioning: When K_SLOT[i] is revoked (SLOT_STATUS[i] = 0x00), WSIM-SLOT-SELECT on both Server and Peer independently walks forward to the next ACTIVE slot. The Peer holds K_UE[0..14] for all slots pre-provisioned at manufacturing, so slot rotation on the Server side requires no OTA update to any UE SIM.

Session Uniqueness: Fresh RAND, fresh ECDH keypairs, and fresh NONCE_S, NONCE_P per session ensure distinct OKM values.

Note on Key Size: MILENAGE uses 128-bit K; P-256 provides approximately 128-bit security. The construction provides 128-bit security overall.

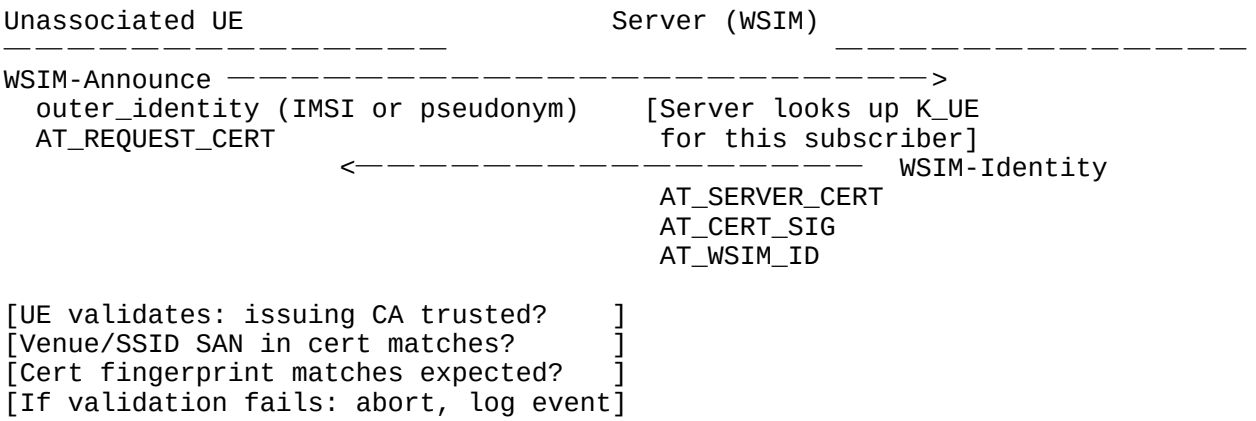
5. EAP-WSIM Protocol

5.1. Pre-Association Phase and Protocol Overview

5.1.1. Pre-Association: Unassociated UE Identity Announcement

Before the EAP exchange begins, an unassociated UE operating in a pre-authentication VLAN sends an initial announcement message to the Server (WSIM). This phase allows the Server to authenticate itself to the UE via its operator certificate before the MILENAGE exchange begins, preventing the Peer from engaging a rogue authenticator.

The pre-association exchange:



AT_REQUEST_CERT: Type 0x1C, Length 0. Signals the Peer requests the Server's operator certificate before proceeding.

AT_SERVER_CERT: Type 0x1D, variable length. The Server's X.509 operator certificate (DER-encoded), issued by the MNO CA and containing the operator-id, venue-id, and SSID in the SAN.

AT_CERT_SIG: Type 0x1E, 64 bytes. ECDSA-P256 signature over (outer_identity || AT_SERVER_CERT || timestamp) using the Server's private key, proving possession.

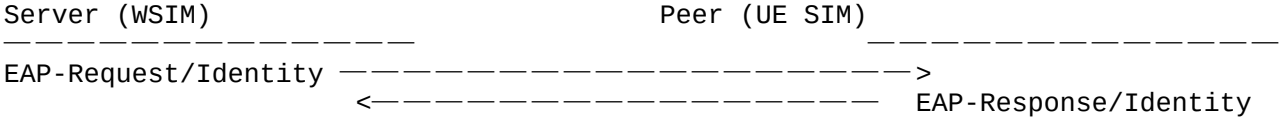
AT_WSIM_ID: Type 0x1F, variable length. A unique identifier for this WSIM instance, allowing the UE application to cross-check against an expected hardware fingerprint if available.

Upon successful certificate validation, the UE proceeds to the standard EAP exchange. The Server's verified identity is bound into the EAP-WSIM session by including the certificate fingerprint in the HKDF info parameter (see Section 4.4, Note on Channel Binding).

NOTE: Implementations that operate within a standard 802.1X/EAP framework where pre-EAP messaging is not available MAY omit the pre-association phase and begin directly with the EAP exchange. In this case, the Server's identity is authenticated implicitly via the AUTN verification in WSIM-Start (the Server proves possession of K_UE, which requires knowledge of K_SLOT).

5.1.2. EAP Exchange

Following the pre-association phase (or directly in constrained deployments), the four-round EAP-WSIM exchange proceeds. EAP-WSIM uses EAP type 0xFE (Expanded Type [RFC3748]) with a vendor-specific method type. [IANA type assignment requested; see Section 9.]



```
                                          (outer_identity: NAI)

  [Server: derive K_UE on-card from K_SLOT + IMSI; steps (a)-(g)]

  WSIM-Start ————————————————————————————————————————————>
    AT_RAND, AT_AUTN, AT_ECDH_SERVER,       [Peer: steps (h)-(o)]
    AT_NONCE_S, AT_COUNTER, AT_MAC
                       <————————————————————  WSIM-Challenge
                                          AT_RES, AT_ECDH_PEER,
                                          AT_NONCE_P, AT_MAC_PEER

  [Server: steps (p)-(r); compute AT_MAC_CONFIRM]

  WSIM-Confirm ——————————————————————————————————————————>
    AT_MAC_CONFIRM                        [Peer: verify confirm]
                       <————————————————————  WSIM-Complete

  EAP-Success (+ MSK to lower layer) ————————>
```

Figure 1: EAP-WSIM Pre-Association and Four-Round Exchange

5.2. Message Header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Type = 0xFE  |              Vendor-Id (3 octets)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Vendor-Type = 0x00000001                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Subtype     |   Reserved    |       Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
Code:        1 = Request (Server->Peer); 2 = Response (Peer->Server)
Type:        0xFE (Expanded Type per [RFC3748] Section 5.7)
Vendor-Id:   [PEN assigned to this document; see Section 9]
Vendor-Type: 0x00000001 (EAP-WSIM)
Subtype:     0x01 WSIM-Start   0x02 WSIM-Challenge
             0x03 WSIM-Confirm 0x04 WSIM-Complete  0x05 WSIM-Error
```

5.3. Attribute Encoding

All attributes use TLV encoding: Type(1B) | Length(1B) | Value

Type	Name	Length	Description
0x10	AT_RAND	16	MILENAGE RAND challenge
0x11	AT_AUTN	16	MILENAGE AUTN token
0x12	AT_ECDH_SERVER	65	Server P-256 uncompressed pubkey
0x13	AT_ECDH_PEER	65	Peer P-256 uncompressed pubkey
0x14	AT_NONCE_S	16	Server session nonce
0x15	AT_NONCE_P	16	Peer session nonce
0x16	AT_RES	16	MILENAGE RES from Peer (f2 output)
0x17	AT_MAC	32	HMAC-SHA-256(K_mac_start, message)
0x18	AT_MAC_PEER	32	HMAC-SHA-256(K_auth, message)
0x19	AT_MAC_CONFIRM	32	HMAC-SHA-256(K_confirm, confirm)
0x1A	AT_COUNTER	4	Anti-replay counter (monotonic)
0x1B	AT_ERROR_CODE	2	Error code (WSIM-Error only)
0x1C	AT_REQUEST_CERT	0	Pre-assoc: Peer requests Server cert
0x1D	AT_SERVER_CERT	variable	Pre-assoc: DER-encoded X.509 cert
0x1E	AT_CERT_SIG	64	Pre-assoc: ECDSA-P256 signature
0x1F	AT_WSIM_ID	variable	Pre-assoc: WSIM instance identifier

AT_MAC and AT_MAC_PEER are computed over the complete EAP message

with the MAC field set to all zeros during computation.

5.4. WSIM-Start (Subtype 0x01, Server to Peer)

Mandatory: AT_RAND, AT_AUTN, AT_ECDH_SERVER, AT_NONCE_S, AT_COUNTER, AT_MAC

The AT_MAC in WSIM-Start uses K_mac_start, derived before session keys are established:

```
K_mac_start = HMAC-SHA-256(K_UE, "WSIM-START-MAC-v1" || RAND)
```

where K_UE is the per-subscriber, IMSI-bound MILENAGE key derived on-card by the Server and pre-provisioned on the Peer's UE SIM.

5.5. WSIM-Challenge (Subtype 0x02, Peer to Server)

Mandatory: AT_RES, AT_ECDH_PEER, AT_NONCE_P, AT_MAC_PEER

AT_MAC_PEER uses K_auth (OKM[96:112]) derived in Section 4.4.

Peer MUST:

1. Verify AT_MAC in WSIM-Start using K_mac_start.
2. Verify AUTN per Section 4.3 step (h); on failure send WSIM-Error with AT_ERROR_CODE = 0x0002.
3. Complete key derivation per Section 4.4.
4. Transmit WSIM-Challenge.

5.6. WSIM-Confirm (Subtype 0x03, Server to Peer)

Mandatory: AT_MAC_CONFIRM

```
AT_MAC_CONFIRM = HMAC-SHA-256(
    key  = K_confirm,
    data = "WSIM-CONFIRM-v1" || AT_RAND || AT_NONCE_S || AT_NONCE_P
)
```

Server MUST verify AT_RES == XRES before sending WSIM-Confirm. If RES != XRES, Server sends WSIM-Error with AT_ERROR_CODE = 0x0003.

5.7. WSIM-Complete (Subtype 0x04, Peer to Server)

No mandatory payload. Signals Peer has verified AT_MAC_CONFIRM. Server responds with EAP-Success.

5.8. WSIM-Error (Subtype 0x05)

Mandatory: AT_ERROR_CODE

Code	Meaning	
0x0001	UNSUPPORTED_METHOD	WSIM-Start is malformed or unsupported
0x0002	AUTN_FAILURE	Peer could not verify AUTN
0x0003	RES_FAILURE	Server: RES != XRES
0x0004	CONFIRM_FAILURE	Peer: AT_MAC_CONFIRM invalid
0x0005	MAC_FAILURE	AT_MAC or AT_MAC_PEER failed
0x0006	REPLAY_DETECTED	AT_COUNTER out of acceptable range
0x0007	GENERAL_FAILURE	Unspecified failure
0x0008	SLOT_MISMATCH	Peer slot selection does not match Server slot; clock skew > 1 hour

On receiving WSIM-Error, the recipient MUST terminate the exchange. The Server MUST send EAP-Failure.

5.9. Anti-Replay

AT_COUNTER is a monotonically increasing 32-bit counter maintained per (Server, Peer) pair, included in WSIM-Start. The Peer MUST verify it is strictly greater than the last accepted value.

The SQN embedded in AUTN provides independent MILENAGE-layer anti-replay per [TS35.205]. Session nonces (AT_NONCE_S, AT_NONCE_P) ensure session key uniqueness as a third independent mechanism.

6. MSK and EMSK

MSK (OKM[0:64]) and EMSK (OKM[64:96]) are defined per [RFC5247]. MSK is exported to the lower layer upon EAP-Success. When used with RADIUS [RFC2865], it is transported in MS-MPPE-Recv-Key and MS-MPPE-Send-Key attributes. EMSK MUST NOT be exported.

When used with IEEE 802.11r, PMK = MSK[0:32] per [IEEE80211] Section 12.7.1.3. The integration of this PMK with the R0KH for Fast Transition key pre-distribution is specified in [WSIM-FT].

7. Relationship to RFC 9678 (EAP-AKA' FS)

RFC 9678 [RFC9678] (EAP-AKA' FS) is the most closely related existing specification. Both EAP-WSIM and EAP-AKA' FS use ephemeral ECDH with MILENAGE to achieve forward secrecy. However, they address fundamentally different deployment scenarios and are not in conflict. This section documents the technical differences.

7.1. MNO Backend Dependency

EAP-AKA' FS [RFC9678], like all prior SIM-based EAP methods, requires the EAP server to obtain authentication vectors from the MNO Authentication Database during each exchange. The ECDH extension in RFC 9678 adds forward secrecy to the session keys but does not change this architecture: the Server cannot authenticate a subscriber without contacting the MNO AD.

EAP-WSIM eliminates this dependency entirely. The WSIM card is a self-contained Authentication Centre: it holds master key material on-card, derives device-specific MILENAGE keys on-card, generates authentication vectors on-card, and verifies subscriber responses on-card. No network communication to any MNO infrastructure occurs during or before the authentication exchange.

This architectural difference enables EAP-WSIM to operate in:

- Enterprise deployments without MNO connectivity agreements
- Air-gapped environments (manufacturing, defence, healthcare)
- Deployments where MNO backend latency or availability is not guaranteed for all authentication events

7.2. Protocol Architecture

EAP-AKA' FS [RFC9678] is an extension to EAP-AKA'. It adds the AT_KDF_FS and AT_PUB_ECDHE attributes to the existing EAP-AKA' message structure. A negotiation round may be required if the Peer's preferred KDF differs from the Server's first proposal.

EAP-WSIM is a new, independent EAP method with its own four-round exchange (WSIM-Start, WSIM-Challenge, WSIM-Confirm, WSIM-Complete). It does not extend any existing EAP method and shares no message types, attribute namespaces, or key derivation steps with EAP-AKA'.

7.3. Key Derivation

In EAP-AKA' FS [RFC9678], forward secrecy is added by computing a

shared ECDH key MK_ECDHE and incorporating it into the existing EAP-AKA' PRF' key derivation alongside the AKA-derived CK' and IK'. The AT_KDF_FS attribute selects the ECDH group and KDF variant.

In EAP-WSIM, MILENAGE and ECDH are combined into the named MILENAGE-ECDH-FWD construction (Section 4). The construction uses HKDF-SHA-256 with IKM = SS || CK || IK, directly binding the ECDH shared secret and MILENAGE key material in a single extraction step with the versioned info string "MILENAGE-ECDH-FWD-v1". This construction is defined as a standalone named primitive with independently verifiable security properties (Section 4.5), facilitating security analysis and future reuse.

7.4. Pre-Session Integrity

In EAP-AKA' FS, the first challenge message is integrity-protected by AT_MAC derived from EAP-AKA' key material (which requires the MNO backend to have provided authentication vectors first).

In EAP-WSIM, the first message (WSIM-Start) is protected by:
K_mac_start = HMAC-SHA-256(K, "WSIM-START-MAC-v1" || RAND)
This is a pre-session MAC derived directly from K and the challenge RAND. It provides integrity and binds the first message to K, though it does not provide forward secrecy for the first message (see Section 10.4 for analysis).

7.5. Summary Comparison Table

Property	RFC 9678 (EAP-AKA' FS)	EAP-WSIM (this document)
MNO backend required	YES	NO
Air-gapped deployment	No	Yes
Protocol structure	Extension to EAP-AKA'	Independent new EAP method
Forward secrecy	Yes (ECDH)	Yes (ECDH)
Mutual authentication	Yes	Yes
Named crypto construction	No	Yes: MILENAGE-ECDH-FWD
First-msg integrity key	From AKA+MNO AD	K_mac_start (offline, from K)
Key slot management	Single root key per subscriber	Multi-slot bundle [WSIM-KEY-SELECT]
802.11r integration	Not specified	[WSIM-FT]
Updates existing RFC	RFC 5448, RFC 9048	None

Table 1: EAP-WSIM vs RFC 9678 (EAP-AKA' FS)

8. Security Claims (per RFC 3748 Section 7.2)

Mechanism:
Shared symmetric key (K, 128-bit MILENAGE root key) plus ephemeral P-256 ECDH.

Ciphersuite negotiation:
None in this version. The construction is fixed as MILENAGE-ECDH-FWD with P-256 and HMAC-SHA-256/HKDF-SHA-256.

Mutual authentication:
YES. Server authenticates to Peer via AUTN; Peer authenticates to Server via RES. Both verifications are required.

Integrity protection:

YES. AT_MAC (K_mac_start) protects WSIM-Start. AT_MAC_PEER (K_auth) protects WSIM-Challenge. AT_MAC_CONFIRM (K_confirm) protects WSIM-Confirm. All MACs use HMAC-SHA-256.

Replay protection:
YES. AT_COUNTER (EAP layer) and SQN in AUTN (MILENAGE layer) provide two independent anti-replay mechanisms. Session nonces provide a third.

Confidentiality:
No. EAP-WSIM does not encrypt EAP messages. Session keys protect subsequent link-layer traffic.

Key derivation:
YES. MSK (64 bytes) and EMSK (32 bytes) per [RFC5247] via MILENAGE-ECDH-FWD (Section 4.4).

Key strength:
128-bit (limited by K size and P-256 security level).

Dictionary attack resistance:
YES. 128-bit K provides resistance. Active attacks require network interaction.

Fast reconnect:
Not defined in this version.

Cryptographic binding:
YES. AT_MAC_PEER binds pk_P and NONCE_P to the MILENAGE authentication outcome via K_auth.

Session independence:
YES. Each session uses fresh RAND, fresh ephemeral ECDH keypairs, and fresh NONCE_S, NONCE_P.

Fragmentation:
YES. EAP-WSIM inherits EAP fragmentation [RFC3748].

Channel binding:
YES. The HKDF info parameter MAY be extended to include server identity; this is a candidate for a future version.

9. IANA Considerations

This document does not update any existing RFC or registry.

9.1. EAP Type

IANA is requested to allocate a permanent EAP method type for EAP-WSIM in the "Method Types" sub-registry of the "Extensible Authentication Protocol (EAP) Registry" at https://www.iana.org/assignments/eap-numbers/.

Until a permanent assignment is available, implementations MUST use EAP Expanded Type (0xFE) [RFC3748] with a Vendor-Id equal to the author’s Private Enterprise Number and Vendor-Type 0x00000001.

9.2. EAP-WSIM Attribute Type Registry

IANA is requested to create a new registry titled "EAP-WSIM Attribute Types" with the initial entries from Section 5.3.
Values 0x00-0x0F and 0x1C-0x7F are Reserved.
Values 0x80-0xFF are for Private Use.
New assignments in 0x00-0x7F require Standards Action [RFC8126].

9.3. EAP-WSIM Error Code Registry

IANA is requested to create a new registry titled "EAP-WSIM Error Codes" with initial values from Section 5.8.
New assignments require Standards Action [RFC8126].

9.4. MILENAGE-ECDH-FWD Construction Version Registry

IANA is requested to create a new registry titled "MILENAGE-ECDH-FWD Construction Versions" to track versions of the construction defined in Section 4. The info string "MILENAGE-ECDH-FWD-v1" (21 bytes, UTF-8) is the initial entry.
New entries require Standards Action [RFC8126].

10. Security Considerations

10.1. Security of MILENAGE-ECDH-FWD

The security of MILENAGE-ECDH-FWD rests on: (a) the security of MILENAGE [TS35.205]; (b) the hardness of the ECDLP on P-256 [FIPS186]; and (c) the security of HKDF-SHA-256 [RFC5869]. As analyzed in Section 4.5, compromise of either component individually does not yield session keys.

10.2. Long-Term Key Storage

K MUST be stored on a hardware security element (SIM card or equivalent) on both Server and Peer. Software-only K storage is strongly discouraged.

10.3. Ephemeral Key Lifecycle

Ephemeral P-256 private keys MUST be generated using a cryptographically secure random number generator and MUST be securely zeroized immediately after the ECDH shared secret SS is computed. SS itself MUST be zeroized after K_session is derived.

10.4. K_mac_start Security Properties

K_mac_start = HMAC-SHA-256(K, "WSIM-START-MAC-v1" || RAND) provides integrity for WSIM-Start but not forward secrecy. An adversary who later learns K can verify that a captured WSIM-Start was genuine, but cannot derive session keys (which require the ephemeral ECDH private key, which has been discarded). This is an acceptable trade-off: the first message must be sent before full session keys are available, and K_mac_start prevents injection of malicious WSIM-Start messages.

10.5. MILENAGE AMF Field

Implementations SHOULD use a distinct AMF value for EAP-WSIM to prevent cross-protocol attacks per [TS33.102].

10.6. Sequence Number Management

The MILENAGE SQN counter MUST be maintained persistently across power cycles. Loss of SQN state requires re-synchronization using the AUTS mechanism per [TS33.102].

11. Privacy Considerations

The EAP outer identity (EAP-Response/Identity) SHOULD use a pseudonym or temporary NAI rather than the permanent IMSI.

Unlike EAP-SIM [RFC4186], EAP-AKA [RFC4187], EAP-AKA' [RFC9048], and EAP-AKA' FS [RFC9678], EAP-WSIM does not cause any RADIUS or Diameter requests to be sent to MNO infrastructure. There is therefore no MNO-observable authentication event that could be used to track subscriber activity across venues.

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[RFC5247] Aboba, B., Simon, D., and P. Eronen, "Extensible Authentication Protocol (EAP) Key Management Framework", RFC 5247, August 2008.

[RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, May 2010.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, June 2017.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.

[RFC8179] Bradner, S. and J. Contreras, "Intellectual Property Rights in IETF Technology", BCP 79, RFC 8179, May 2017.

[TS35.205] 3GPP, "3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*", 3GPP TS 35.205.

[FIPS186] NIST, "Digital Signature Standard (DSS)", FIPS PUB 186-5, February 2023.

12.2. Informative References

[RFC2865] Rigney, C., et al., "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RFC4186] Haverinen, H. and J. Salowey, "Extensible Authentication Protocol Method for GSM Subscriber Identity Modules (EAP-SIM)", RFC 4186, January 2006.

[RFC4187] Arkko, J. and H. Haverinen, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)", RFC 4187, January 2006.

[RFC5448] Arkko, J., Pauly, D., Haverinen, H., and J. Salowey, "Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA')", RFC 5448, May 2009.

[RFC6733] Fajardo, V., et al., "Diameter Base Protocol", RFC 6733, October 2012.

[RFC9048] Arkko, J., et al., "Improved Extensible Authentication

Protocol Method for 3GPP Mobile Network Authentication and Key Agreement (EAP-AKA')", RFC 9048, October 2021.

[RFC9190] Preu Mattsson, J. and M. Tiru, "EAP-TLS 1.3", RFC 9190, February 2022.

[RFC9678] Arkko, J., Norrman, K., and J. Preu Mattsson, "Forward Secrecy Extension to the Improved Extensible Authentication Protocol Method for Authentication and Key Agreement (EAP-AKA' FS)", RFC 9678, March 2025.

[TS33.102] 3GPP, "3G Security; Security Architecture", 3GPP TS 33.102.

[IEEE80211] IEEE, "IEEE Standard for Information Technology -- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE Std 802.11-2020.

[WSIM-KEY-SELECT] Gupta, P., "WSIM Key Slot Selection: Non-Reversible Time-of-Day IMSI-Bound Key Bundle Management for EAP-WSIM", Work in Progress, draft-gupta-emu-wsim-key-select-00, 2026.

[WSIM-FT] Gupta, P., "EAP-WSIM Integration with IEEE 802.11r Fast Transition Key Pre-Distribution", Work in Progress, draft-gupta-emu-wsim-ft-00, 2026.

Appendix A. MILENAGE-ECDH-FWD Test Vectors

The following test vectors verify Section 4 independently. MILENAGE parameters use 3GPP TS 35.208 Test Set 1 values.

Reference implementation: gen_testvecs.py (contact author). The P-256 keypairs in A.5 were generated for this document. Implementations MUST generate fresh keypairs per session.

A.1. MILENAGE Input Parameters

K (MILENAGE root key, 16 bytes):
465B5CE8B199B49FAA5F0A2EE238A6BC

OP (operator variant algorithm key, 16 bytes):
CDC202D5123E20F62B6D676AC72CB318

OPc (derived: AES_K(OP) XOR OP, 16 bytes):
CD63CB71954A9F4E48A5994E37A02BAF

RAND (random challenge, 16 bytes):
23553CBE9637A89D218AE64DAE47BF35

SQN (48-bit sequence number, 6 bytes):
FF9BB4D0B607

AMF (Authentication Management Field, 2 bytes):
B9B9

A.2. MILENAGE Sub-Function Outputs

NOTE: f2 (RES), f3 (CK), f4 (IK), and f5 (AK) have been verified against 3GPP TS 35.208 Test Set 1 and match exactly. f1 (MAC_A, MAC_S) uses the formula: TEMP XOR rot(IN1 XOR OPc, 8 bytes), where TEMP = AES_K(RAND XOR OPc) and IN1 = SQN || AMF || SQN || AMF. Cross-verification of f1 against TS 35.208 is deferred to -01.

```
   f1   MAC_A (8 bytes):   4A9FFAC354DFAFB3
   f1*  MAC_S (8 bytes):   01CFAF9EC4E871E9

   f2   RES   (8 bytes):   A54211D5E3BA50BF
        TS35.208 ref:       A54211D5E3BA50BF  [MATCH]

   f3   CK    (16 bytes):  B40BA9A3C58B2A05BBF0D987B21BF8CB
        TS35.208 ref:       B40BA9A3C58B2A05BBF0D987B21BF8CB  [MATCH]

   f4   IK    (16 bytes):  F769BCD751044604127672711C6D3441
        TS35.208 ref:       F769BCD751044604127672711C6D3441  [MATCH]

   f5   AK    (6 bytes):   AA689C648370
        TS35.208 ref:       AA689C648370  [MATCH]
```

A.3. AUTN Construction and K_mac_start

```
   SQN XOR AK (6 bytes):
      55F328B43577

   AUTN = (SQN XOR AK) || AMF || MAC_A (16 bytes):
      55F328B43577B9B94A9FFAC354DFAFB3

   K_mac_start derivation:
      label  = "WSIM-START-MAC-v1" (17 bytes)
               5753494D2D53544152542D4D41432D7631
      input  = label || RAND (33 bytes)
      K_mac_start = HMAC-SHA-256(K, input) (32 bytes):
         C68233159C2A1B7A84CB3DD0172CD17A
         76EFF79600485FA66F1277158B6AC799
```

A.4. Session Nonces

```
   NONCE_S (server nonce, 16 bytes):
      5A8D3F2B1C9E7041A6D5E4F3B2C1A090

   NONCE_P (peer nonce, 16 bytes):
      A1B2C3D4E5F60718293A4B5C6D7E8F90
```

A.5. P-256 ECDH Key Exchange

```
   NOTE: These keypairs are example values for vector verification.
   Production implementations MUST generate fresh keypairs per session
   using a CSPRNG.

   Server private scalar d_S (32 bytes):
      6432A7B71C016E45760781F9E921C923
      66B2CEC9F77B78CE659CB88FA27E9BEC

   Server public key pk_S (uncompressed format: 0x04 || x || y):
      04
      x: 47775180089DDE81621F8B86EEB57F3F
         DCDE3E81512734F0E505DDC9724B1FCD
      y: EF7A5534D815387A06E63CF3507E0F40
         41B5DAFE4B02B3FD259C3515ED6E5DEE

   Peer private scalar d_P (32 bytes):
      874120DD6BA2F6E547A1E9B4C04AE761
      320C87ECEFD0C022F9124F300A66CAB1

   Peer public key pk_P (uncompressed format: 0x04 || x || y):
      04
      x: 4097F2E695DCA36726D00324E4AB1EE8
         49A0FD08F97D523E056781B37B13EA3C
      y: 4795796AACBAC948202F5B3871CB9AF0
```

```
      EEEA5ECD468171B4DF2E9E306133465E

   SS = P256-ECDH(sk_S, pk_P) = P256-ECDH(sk_P, pk_S) (32 bytes):
      5B073428B4D4CE1AF5194DAF5015FF4F
      6FD2AE2D5A442B3F2546B9C39E029571
```

A.6. HKDF-SHA-256 Key Derivation

```
   IKM = SS || CK || IK  (64 bytes total):
      5B073428B4D4CE1AF5194DAF5015FF4F6FD2AE2D5A442B3F2546B9C39E029571
      B40BA9A3C58B2A05BBF0D987B21BF8CB
      F769BCD7510446046041276727 11C6D3441

   salt = NONCE_S || NONCE_P  (32 bytes):
      5A8D3F2B1C9E7041A6D5E4F3B2C1A090A1B2C3D4E5F60718293A4B5C6D7E8F90

   info = "MILENAGE-ECDH-FWD-v1"  (21 bytes):
      4D494C454E4147452D454344482D4657442D7631

   L = 128 bytes

   OKM = HKDF-SHA-256(IKM, salt, info, 128):
      [  0: 32] 928815EBF4B5498A77A19DB6A04B9EFB
                439A604B7DC6DD558C920E4D067E21EF
      [ 32: 64] 26C8BA4802C95CF3AB3D49C7B9688019
                2F8F931CACC2405F52186A99DCD4A95D
      [ 64: 96] 996AF0830F1FD6AD4C0450563568EAE4
                7DD5E09ED9847379CA3D13A22D0C80F2
      [ 96:128] 2D682081C8A223628E2C54F449D71F35
                7741CE07737EEA89B2F422D77C132B48
```

A.7. Output Key Material

```
   MSK  = OKM[0:64]:
      [0:32]  928815EBF4B5498A77A19DB6A04B9EFB
              439A604B7DC6DD558C920E4D067E21EF
      [32:64] 26C8BA4802C95CF3AB3D49C7B9688019
              2F8F931CACC2405F52186A99DCD4A95D

   EMSK = OKM[64:96]:
      996AF0830F1FD6AD4C0450563568EAE4
      7DD5E09ED9847379CA3D13A22D0C80F2

   K_auth = OKM[96:112]:
      2D682081C8A223628E2C54F449D71F35

   K_confirm = OKM[112:128]:
      7741CE07737EEA89B2F422D77C132B48

   PMK (for IEEE 802.11r R0KH) = MSK[0:32]:
      928815EBF4B5498A77A19DB6A04B9EFB
      439A604B7DC6DD558C920E4D067E21EF
```

A.8. Message Authentication Codes

```
   AT_MAC (WSIM-Start):
      Input = RAND || AUTN || NONCE_S  (48 bytes)
      Key   = K_mac_start
      AT_MAC = HMAC-SHA-256(K_mac_start, input) (32 bytes):
         6A915CDDEFF2221755385D5D296ED5C9
         1A384AC8F80B18BFB40FF4B11BA779BD

   AT_MAC_PEER (WSIM-Challenge):
      Input = RES (8B) || 0x04 || pk_P.x (32B) || pk_P.y (32B) || NONCE_P (16B)
      Key   = K_auth
      AT_MAC_PEER = HMAC-SHA-256(K_auth, input) (32 bytes):
```

```
      EEFD2907053B10545559295B1B69172E
      5AD03B21711361A90F467BB22F9D15A5

   AT_MAC_CONFIRM (WSIM-Confirm):
      Input = "WSIM-CONFIRM-v1" || RAND || NONCE_S || NONCE_P
      Key   = K_confirm
      AT_MAC_CONFIRM = HMAC-SHA-256(K_confirm, input) (32 bytes):
         D469BB2CCCD9C2D0F90B4C998E94B6C4
         1136D0744E7C680DCD3B36E3A2785316
```

A.9. Test Vector 2: Session Independence

Purpose: Verify that a different RAND with the same K produces a completely different MSK, confirming session independence.

```
   RAND2  = 9F7C8D556B7BE5A1234CBF89D3E4A150
   NONCE_S2 = 11223344556677889900AABBCCDDEEFF
   NONCE_P2 = FFEEDDCCBBAA00998877665544332211

   SS2 (fresh ECDH, different ephemeral keys):
      3AA3A1D8B2A311C3AA7EDBB9BD0AFE3B
      05A2B276164A67A202BBB1A8A7727222

   MSK2[0:32]:
      7C6497F0DF26A49FCC3FB98F1ADBC2B2
      4C00A83A0CAD60578A39F3ACE87167BD

   MSK2[32:64]:
      8EB2ECD1A936A744F71A3B214040CD58
      3C6265D8311D43AA09EB9FA1AD7FC8F4

   MSK1 != MSK2: TRUE  (session independence verified)
```

Appendix B. Comparison with Prior EAP SIM-Based Methods

B.1. EAP-SIM (RFC 4186)

EAP-SIM uses GSM triplets and requires HLR contact for each exchange. It provides no mutual authentication (the network is not authenticated to the UE) and no forward secrecy. EAP-WSIM provides offline operation, full mutual authentication, and forward secrecy.

B.2. EAP-AKA (RFC 4187)

EAP-AKA uses MILENAGE/UMTS AKA and provides mutual authentication but requires AuC/HLR contact and provides no forward secrecy. EAP-WSIM uses the same MILENAGE mutual authentication but eliminates the backend dependency and adds forward secrecy via MILENAGE-ECDH-FWD.

B.3. EAP-AKA’ (RFC 9048)

EAP-AKA’ improves key derivation over EAP-AKA using a SHA-256-based PRF but still requires UDM/AuC contact and provides no forward secrecy. EAP-WSIM: offline, adds forward secrecy.

B.4. EAP-AKA’ FS (RFC 9678)

RFC 9678 adds ECDH forward secrecy to EAP-AKA’ as an optional extension. It is the most similar existing specification. The critical distinction is that RFC 9678 still requires MNO Authentication Database contact for every authentication event; EAP-WSIM requires no MNO backend contact at any time. See Section 7 for a complete technical comparison.

RFC 9678 and EAP-WSIM are complementary, not competing. RFC 9678 is appropriate when MNO infrastructure is available and the operator controls both sides of the authentication. EAP-WSIM is appropriate when offline operation is required, when the enterprise does not have an MNO backend relationship, or when authentication latency or availability guarantees cannot be met by the MNO.

B.5. Named Construction

None of the prior SIM-based EAP methods define their use of MILENAGE and (where applicable) ECDH as a named, reusable construction. EAP-WSIM introduces MILENAGE-ECDH-FWD (Section 4) as an explicitly named primitive with independently stated security properties (Section 4.5). This enables independent cryptographic analysis and potential reuse in non-EAP protocols.

B.6. Key Slot Management

EAP-SIM, EAP-AKA, EAP-AKA', and EAP-AKA' FS all use a single root key per subscriber. Compromise of that key requires physical re-provisioning of all UE SIMs. EAP-WSIM supports a multi-slot key bundle with independent per-slot revocation and server-side rotation, without requiring any update to pre-provisioned UE SIMs. This is specified in [WSIM-KEY-SELECT].

Appendix C. Future Work

C.1. Key Slot Selection (draft-gupta-emu-wsim-key-select)

A non-reversible, Time-of-Day and IMSI-bound algorithm to select the active key slot from the bundle, enabling hourly rotation without UE SIM re-provisioning and without leaking subscriber identity through the slot index sequence.

C.2. IEEE 802.11r Fast Transition (draft-gupta-emu-wsim-ft)

Specification of how the EAP-WSIM MSK is used by a co-located IEEE 802.11r R0KH to pre-distribute per-AP PMK-R1 at authentication time, enabling Fast Transition roaming at 30-80ms without any RADIUS backend contact during the transition.

C.3. Ciphersuite Negotiation

Algorithm agility to support future 256-bit MILENAGE variants and alternative elliptic curves (P-384, P-521).

C.4. Fast Reconnect

Session resumption without a full four-round exchange, preserving forward secrecy properties.

C.5. Post-Quantum Hybrid Variant

A hybrid variant combining MILENAGE-ECDH-FWD with a post-quantum KEM (Key Encapsulation Mechanism) per [FIPS203] to maintain security against a cryptographically relevant quantum computer.

C.6. TPM-Based Server Deployment Option

An alternative deployment where the Server (MEA) uses a hardware TPM 2.0 module rather than a SIM card for master key storage. In this variant:

- K_SLOT is stored under a TPM 2.0 non-migratable key object

with persistent handle 0x81000001.

- K_UE derivation occurs via TPM2_HMAC using the bound key, ensuring K_SLOT never appears in cleartext outside the TPM.

- Server authentication to the Peer uses an EAP-TLS exchange with the TPM-backed operator certificate (instead of the WSIM card's MILENAGE-based AUTN).

- MNO remote provisioning (ECDH-attested enrollment) is required for the TPM option. The SIM-based deployment option requires only offline trust anchor provisioning.

Both deployment options produce an identical MSK and are transparent to the Peer (UE SIM). The TPM option is described in the companion patent disclosure and may be specified as a separate profile if WG interest warrants it.

Author's Address

Praveen Gupta
MobileStack

Email: pgupta@mobilestack.com
URI: https://datatracker.ietf.org/doc/draft-gupta-emu-eap-wsim/