### IEEE P802.11 Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Protected Password Identifiers for Privacy | | | | |
| Date: 2023-01-09 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Dan Harkins | HPE |  |  |  |
| Michael Montemurro | Huawei |  |  |  |
| Jouni Malinen | Qualcomm |  |  |  |
|  |  |  |  |  |

Abstract

This submission proposes a way to provide privacy protections to SAE password identifiers.

SAE password identifiers allow an ESS (identified by a single SSID) to support multiple passwords for access. A password identifier maps to one, and only one, password. Passwords and their identifiers can be given out on a per-STA basis or to a group of STAs that share common access permissions. It is possible to assign authorization policy—VLAN, ACLs, etc—to users based on the password they used when authenticating.

Vendors have done similar per-user credential schemes with PSK mode. These schemes suffer from supporting a limited number of unique credentials and requirements for the AP to do a considerable amount of work to support the scheme. In spite of that, they are still popular. But these schemes do not work with SAE due to the forward secrecy properties of SAE. Password identifiers are the way to provide support for this use case with the enhanced security that SAE provides.

Unfortunately, SAE password identifiers are passed in the clear. This has brought up privacy concerns and an unwillingness to deploy SAE password identifiers in spite of them being extremely useful and solving legitimate use cases. This is causing a reluctance in certain markets to adopt stronger SAE authentication in favour of insecure PSK authentication in order to maintain the per-user credential capability. This also creates friction when stronger SAE authentication is being mandated for certain bands because this use case cannot be supported without password identifiers.

To address privacy concerns, it is proposed to use HPKE (RFC 9810) to wrap a plaintext password identifier in a public key of the AP or mesh peer.

This scheme has the following security properties:

* An attacker cannot determine a protected identity;
* An attacker cannot connect protected identities across distinct SAE protocol runs;
* Password identifiers can be arbitrarily padded to foil passive traffic analysis;
* Protected identities are secure under a security bounds of the elliptic curve and AEAD cipher used in HPKE;
* An attacker cannot tamper with or substitute identifiers to connect distinct runs of SAE;
* An AP needs to only manage a single credential;
* Identities are protected against members of the same group;
* Protected password identifiers in a mesh is supported.

The proposal can be implemented by accepting the following changes to Draft P802.11REVme\_2.0:

*Instruct the editor to add the following to section 2:*

IETF RFC 9180, Hybrid Public Key Encryption, February 2022.

*Instruct the editor to obtain a number assignment from ANA for <ANA-1> and to modify section 9.3.3.2 as indicated:*

**9.3.3.2 Beacon frame format**

|  |  |  |
| --- | --- | --- |
| Order | Information | Notes |
| 91 | WUR Discovery | The WUR Discovery element is optionally present if dot11WUROptionImplemented is true and either dot11WURDiscoveryImplemented or Dot11WURNeighborDiscoveryImplemeneted is true; otherwise it is not present |
| <ANA-1> | Privacy Public Key | The Privacy Public Key element is present if the AP supports SAE password identity privacy; otherwise it is not present |
| Last – 1 | Vendor Specific | One or more Vendor Specific Elements are optionally present. |
| Last | MME | The MME is present if dot11BeaconProtectionEnabled is true at the AP |

*Instruct the editor to obtain a number assignment from ANA for <ANA-2> and to modify section 9.3.3.11 as indicated:*

**9.3.3.11 Authentication frame format**

**Table 9-68—Authentication frame body**

|  |  |  |
| --- | --- | --- |
| **Order** | **Information** | **Notes** |
| 22 | Password Identifier | The Password Identifier element is optionally present in certain Authentication frames as defined in Table 9-69 (Presence of fields and elements in Authentication frames) |
| 23 | Rejected Groups | The Rejected Groups element is present only in certain Authentication frames as defined in Table 9-69 (Presence of fields and elements in Authentication frames). |
| 24 | Anti-Clogging Token Container | The Anti-Clogging Token Container element is present only in certain Authentication frames as defined in Table 9-69 (Presence of fields and elements in Authentication frames). |
| 25 | AKM Suite Selector | The AKM Suite Selector element is present only in certain Authentication frames as defined in Table 9-69 (Presence of fields and elements in Authentication frames). |
| <ANA-2> | Protected Password Identifier | The Protected Password Identifier element is optionally present in Authentication frames as defined in Table 9-69 (Presence of fields and elements in Authentication frames). |
| Last | Vendor Specific | One or more Vendor Specific elements are optionally present. These elements follow all other elements. |

**Table 9-69—Presence of fields and elements in Authentication frames**

|  |  |  |  |
| --- | --- | --- | --- |
| Authentication algorithm | Authentication transaction sequence number | Status code | Presence of fields and elements from order 4 onwards |
| SAE | 1 | Any | The Scalar field is present if the Status Code field is zero or 126.  The FFE field is present if the Status Code field is zero or 126.  When the hunting-and-pecking method is used to drive the PWE, the Anti-Clogging Token field is present if the Status Code field is ANTI\_CLOGGING\_TOKEN\_REQUIRED or if the Authentication frame is in response to a previous rejection with the Status Code field equal to ANTI\_CLOGGING\_TOKEN\_REQUIRED.  The Finite Cyclic Group field is present if the Status Code field is zero, ANTI\_CLOGGING\_TOKEN\_REQUIRED, 77, or 126.  The Password Identifier element is optionally present if the Protected Password Identifier element is not present and the Status Code field is zero, 123, or 126.  The Protected Password Identifier element is optionally present if the Password Identifier field is not present and the Status Code field is zero, 123, or 126. If the Status Code field is 126, the Rejected Groups element is conditionally present as described in 12.4.7.3 (Encoding and decoding of SAE Commit messages); otherwise the Rejected Groups element is not present.  The Rejected Groups element is present if the Status Code field is 126.  When the hash-to-element method is used to derive the PWE, the Anti-Clogging Token Container  element is present if the Status Code field is ANTI\_CLOGGING\_TOKEN\_REQUIRED or if the Authentication frame is in response to a previous rejection with the Status Code field equal to ANTI\_CLOGGING\_TOKEN\_REQUIRED.  If the Status Code field is 126, the Rejected Groups element is conditionally present as described in 12.4.7.3 (Encoding and decoding of SAE Commit messages); otherwise the Rejected Groups element is not present. When the hash-to-element method is used to derive the PWE, the Anti-Clogging Token Container element is present if the Status Code field is ANTI\_CLOGGING\_TOKEN\_REQUIRED or if the Authentication frame is in response to a previous rejection with the Status Code field equal to ANTI\_CLOGGING\_TOKEN\_REQUIRED. (M67)The AKM Suite Selector element is present if 00-0F-AC:24 or 00-0F-AC:25 is the intended AKM (see 12.4.5.3 (Construction of an SAE Commit message) and 12.4.5.4 (Processing of a peer’s SAE Commit message)); otherwise, it is not present. |

*Instruct the editor to obtain an assignment from ANA for <ANA-3> and to modify table 9-78 as indicated:*

**Table 9-78—Status codes**

|  |  |  |
| --- | --- | --- |
| 129 | TCLAS\_PROCESSING\_TERMINATE  D\_POLICY\_CONFLICT | Requested TCLAS processing has been terminated by the AP due to conflict with higher layer QoS policies. |
| <ANA-3> | BAD\_PROTECTED\_IDENTITY | The SAE protected password identifier in the SAE Commit message was invalid. |
| 131-65 535 |  | Reserved |

*Instruct the editor to modify table 9-128 as indicated, obtain an assignment from ANA for <ANA-4> and <ANA-5> .*

**9.4.2 Elements**

**9.4.2.1 General**

**Table 9-128—Element IDs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Element ID** | **Element ID Extension** | **Extensible** | **Fragmentable** |
| Originator Preferred MCS (see 9.4.2.297 (Originator Preferred MCS element | 255 | 116 | Yes | No |
| Privacy Public Key (see 9.4.2.X (Privacy Public Key)) | 255 | <ANA-4> | Yes | No |
| Protected Password Identifier (see 9.4.2.X+1 (Protected Password Identifier)) | 255 | <ANA-5> | Yes | No |
| Reserved | 255 | <ANA-5> + 1-255 |  |  |

*Instruct the editor to create a new section as below, replacing X with the appropriate number and assigning the figure numbers appropriately:*

**9.4.2.X Privacy Public Key element**

The Privacy Public Key element is used to convey information necessary to construct an elliptic curve public key used to afford privacy to a STA’s personally identifiable information. The format of the Privacy Public Key element is shown in Figure 9-XYZ (Privacy Public Key element).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element ID | Length | Element ID Extension | Finite Cyclic Group | Privacy Public Key |

Octets: 1 1 1 2 variable

The Element ID, Length, and Element ID Extension fields are defined in 9.4.2.1 (General).

The Finite Cyclic Group field contains an unsigned integer from a repository maintained by IANA as “Group Description” attributes for IETF RFC 2409 (IKE). It is identical to the field defined in 9.4.1.42 except that it is restricted to public keys supported by IETF RFC 9180 (HPKE).

The Privacy Public Key field contains an octet string that, when converted to an unsigned integer per 12.4.7.2.3 is the x-coordinate of an elliptic curve public key from the specified finite cyclic group.

**9.4.2.X+1 Protected Password Identifier element**

The Protected Password Identifier element is used to convey a password identifier during an authentication exchange in a manner that will hide the actual identifier from attackers. The format of the Protected Password Identifier element is shown in Figure 9-XYZ+1 (Protected Identifier element format).

|  |  |  |  |
| --- | --- | --- | --- |
| Element ID | Length | Element ID  Extension | Protected Identifier |

Octets: 1 1 1 variable

**Figure 9-XYZ+1—Protected Identifier element format**

The Element ID, Length, and Element ID Extension fields are defined in 9.4.2.1 (General).

The Protected Identifier field contains an opaque octet string.

*Instruct the editor to modify section 12.4.1 as indicated:*

**12.4.1 SAE Overview**

The parties involved are called STA-A and STA-B. They are identified by their MAC addresses, STA-A-MAC

and STA-B-MAC, respectively. STAs begin the protocol when they discover a peer by receiving Beacon or Probe Response frame(s), or when they receive an Authentication frame indicating SAE authentication from

a peer.

SAE supports the use of password identifiers to enable groupings of STAs under a single password or for unique, per-STA assignment of passwords, all under a single SSID. There is a 1:1 mapping of password identifier to password. For privacy, password identifiers can be protected using the trusted public key of an AP or mesh STA. Public keys can be pre-provisioned with the password and password identifier or they can be disclosed in beacons. Unless the public key is provisioned at the same time as the password and identifier, the first connection a STA establishes will be susceptible to an active attack (if the public key has not yet been authenticated) or to snooping by third parties (if using unprotected password identifiers). Public keys and the groups from which they are created are stored in dot11RSNAConfigPasswordPeerPubKey and dot11RSNAConfigPasswordPubKeyGrp, respectively. A STA may elect to support trust on use or restrict protected password identifiers to pre-provisioned public keys only by setting the dot11RSNAConfigPasswordPubKeyLocked variable to False or True, respectively.

SAE is an RSNA authentication protocol and is selected according to 12.6.2 (RSNA selection).

*Instruct the editor to modify section 12.4.3 as indicated:*

**12.4.3 Representation of passwords and password identifiers**

In an infrastructure BSS for which an SAE AKM is indicated, the AP shall set the SAE Password Identifiers In Use subfield of the Extended Capabilities field of the Extended Capabilities element to 1 if any entry in the dot11RSNAConfigPasswordValueTable has a non-NULL dot11RSNAConfigPasswordIdentifier, and shall set it to 0 otherwise. Similarly, an AP shall set the SAE Password Identifiers Used Exclusively subfield of the Extended Capabilities field of the Extended Capabilities element to 1 if every entry in the dot11RSNAConfigPasswordValueTable has a non-NULL dot11RSNAConfigPasswordIdentifier and shall set it to 0 otherwise.

SAE password identifiers can expose information that a passive attacker could use to identify and track STAs that authenticate to a network. In order to provide privacy, protected password identifiers can optionally be used by STAs. Protected password identifiers appear as opaque strings when passed in SAE Commit messages and are parsed and understood by APs and mesh STAs that advertise a privacy public key in beacons. APs in an ESS can share the same public key. The method by which the public key is shared by APs in an ESS is out of scope of the standard.

A STA protects SAE password identifiers by first obtaining the privacy public key of an AP or peer mesh STA. If there is a trusted public key in dot11RSNAConfigPasswordPeerPubKey, either through pre-provisoning or trust through prior use, that matches the public key advertised by the AP, the STA uses that value as the x-coordinate of an elliptic curve defined by dot11RSNAConfigPasswordPubKeyGrp. Otherwise, the STA checks the value of dot11RSNAConfigPasswordPubKeyLocked. If it is True the STA shall abandon the connection attempt. If it is False, the STA extracts the curve identifier and octet string from the beacon of the AP or peer mesh STA and converts the octet string into an x-coordinate of an elliptic curve using the octet string to integer conversion of subclause 12.4.7.2.3. It then uses the equation of the defined curve to produce a y-coordinate (the sign does not matter) and reconstruct a point on the elliptic curve. Finally, it uses the reconstructed public key to wrap its password identifier using HPKE (IETF RFC 9180) in the “single shot” mode of encryption to a public key. The AAD used in the HPKE operation shall be the scalar field from the SAE Commit message in which the protected password identifier is to be inserted.

Prior to wrapping, the password identifier shall be padded. The padding consists of a single octet indicating the number of random octets that follow, followed by that number of octets. The pad length indicator and the pad together shall be prepended to the password identifier. This padded password identifier is used as the plaintext to the HPKE wrapping. STAs should vary the amount of padding used to thwart traffic analysis. Padding, exclusive of the pad length identifier, should not be more than 16 octets and may be zero (i.e. the padding consists of a single octet whose value is zero).The output of HPKE shall become the Protected Identifier field of the Protected Identifier element and added to the SAE Commit message to which it is bound.

STAs that support protected password identities shall support the following options from IETF RFC 9180:

* KEMs using NIST p-256, with both compressed and uncompressed outputs
* KDF using SHA-256
* AEAD function of AES-GCM-128

STAs should use cryptographic primitives with HPKE that are commensurate with the primitives being used with SAE (see Table 12-1).

When dot11RSNAConfigPasswordPubKeyLocked is set to False, a STA gains trust in an AP or mesh STA’s untrusted public key when the SAE exchange in which it was used successfully terminates, at which time the key is stored in dot11RSNAConfigPasswordPeerPubKey and the group of the key is stored in dot11RSNAConfigPasswordPubKeyGrp for the Dot11RSNAConfigPasswordValueEntry containing the password and password identifier.

An AP or peer mesh STA that receives a Protected Identifier element in an SAE Commit message shall unwrap it using HPKE in the “single shot” mode of decryption to its public key using the scalar field of the SAE Commit message as AAD. Failure of HPKE unwrapping shall result in an authentication failure. The first octet of the output of HPKE indicates the amount of padding that follows and the pad indicator and pad shall be removed, leaving the plaintext password identifier used to complete the SAE protocol.

*Instruct the editor to modify section 12.4.4.2.3 as indicated:*

**12.4.4.2.3 Hash-to-element generation of the password element with ECC groups**

The SSWU method produces two values, x1, and x2, at least one of which will represent an abscissa of a point on the curve. If x1 is the abscissa, then x1 becomes the x-coordinate otherwise x2 becomes the x-coordinate. The equation of the curve with the x-coordinate produces the square of the y-coordinate which is recovered by taking the square root. The two possible results of the square root are discriminated by checking its least significant bit with the least significant bit of u. The result is a point on the curve.

The *identifier* used in the calculations above shall be the value extracted from the SAE Commit message. If protected password identifiers are used, the identifier in the calculations above shall be the encrypted value from the Protected Identifier field of the Protected Password Identifier element, otherwise it shall be the value from the Identifier field of the Password Identifier element.

*Instruct the editor to modify section 12.4.4.3.3 as indicated where the deleted duplicative text occurs immediately before a formula for calculation of PT (formula in between the text is not shown):*

**12.4.4.3.3 Direct generation of the password element with FFC groups**

This secret PT is stored until needed to generate a session specific PWE (see 12.4.5.2 (PWE and secret generation)).

The *identifier* used in the calculations above shall be the value extracted from the SAE Commit message. If protected password identifiers are used, the identifier in the calculations above shall be the encrypted value from the Protected Identifier field of the Protected Password Identifier element, otherwise it shall be the value from the Identifier field of the Password Identifier element.

*Instruct the editor to modify section 12.4.5.3 as indicated:*

**12.4.5.3 Construction of an SAE Commit message**

This message shall be transmitted to the peer as described in 12.4.7 (Framing of SAE). The temporary secret *mask* may be deleted at this point.

If a STA possesses the public key of the SAE peer (either an AP or peer mesh STA) and has a password identifier, it shall wrap the password identifier according to Subclause 12.4.3 and shall add the Protected Password Identifier element to its SAE Commit message. If a STA does not possess the public key of the SAE peer but does have a password identifier, the STA may refuse to connect to the peer using its password identifier (and password assigned to it) or may forgo the benefits of privacy and pass the password identifier in the clear by adding a Password Identifier element to its SAE Commit message.

To derive keys for use with AKM 00-0F-AC:24 or AKM 00-0F-AC:25, an AKM Suite Selector element indicating 00-0F-AC:24 or 00-0F-AC:25 shall be included in an SAE Commit message transmitted to the peer.

*Instruct the editor to modify section 12.4.5.4 as indicated:*

**12.4.5.4 Processing of a peer’s SAE Commit message**

If the peer’s SAE Commit message contains a password identifier, the value of that identifier shall be used in construction of the password element (PWE) for this exchange. If the peer’s SAE Commit message contains an encrypted identifier, the encrypted identifier shall be used in construction of the secret element PT for this exchange (see 12.4.4.2.3 (Hash-to-element generation of the password element with ECC groups) and 12.4.4.3.3 (Direct generation of the password element with FFC groups). If a password identifier, or protected password identifier, is present in the peer’s SAE Commit message and there is no password with the given (decrypted, if protected) identifier a STA shall fail authentication. If a protected password identifier cannot be processed (see 12.4.3) the STA shall respond with an SAE Commit message with a status code of BAD\_PROTECTED\_IDENTITY and fail authentication.

NOTE—SAE Commit messages are unprotected and forgeable. A STA that receives an SAE Commit message with a status code of BAD\_PROTECTED\_IDENTITY might attempt additional authentication attempts before abandoning the exchange, and might elect to connect using an unprotected password identifier.

*Instruct the editor to modify section 12.4.5.6 as indicated:*

**12.4.5.6 Processing of a peer’s SAE Confirm message**

The peer-send-confirm shall be encoded according to 9.2.2 (Conventions). The elements and scalars shall be in

the format they were encoded in when transmitted in an SAE Commit message as described in 12.4.7.3

(Encoding and decoding of SAE Commit messages). If the verifier differs from the peer-confirm, verification

of the peer’s SAE Confirm message shall fail. If verification fails and a protected password identifier was used for this exchange, a non-AP STA or mesh STA that produced the protected password identifier shall assign a zero length string to dot11RSNAConfigPasswordPeerPubKey and a value of zero to dot11RSNAConfigPasswordPubKeyGrp in the entry of the dot11RSNAConfigPasswordValueTable that was used to perform the SAE exchange.

*Instruct the editor to modify sections 12.4.8.6.3 and 12.4.8.6.4 as indicated:*

**12.4.8.6.3 Protocol instance behavior—Nothing state**

Upon receipt of a Com event, the protocol instance shall check the Status of the Authentication frame. If the Status code is not SUCCESS(#1277) or SAE\_HASH\_TO\_ELEMENT, the frame shall be silently discarded and a Del event shall be sent to the parent process. Otherwise, the frame shall be processed by first checking whether a password identifier or protected password identifier is present. If a password identifier is present and there is no password associated with that identifier, BadID shall be set and the protocol instance shall construct and transmit an Authentication frame with Status Code set to UNKNOWN\_PASSWORD\_IDENTIFIER. If a protected password identifier is present it shall be unwrapped. If unwrapping fails, BadID shall be set, the protocol instance shall construct and transmit an Authentication frame with status code set to BAD\_PROTECTED\_IDENTIFIER, and a Del event shall be sent to the parent process. If unwrapping succeeds, the unwrapped data becomes the password identifier for this transaction. If there is no password identifier or if a password is associated with that identifier, the frame shall be processed by next checking the finite cyclic group field to see if the requested group is supported.

**12.4.8.6.4 Protocol instance behavior—Committed state**

* If there is a password identifier associated with the password when the protocol instance constructed its SAE Commit message and either there is no password identifier in the received frame or the password identifier in the received frame does not match the password identifier used to construct the protocol instance’s SAE Commit message, BadID shall be set, the protocol instance shall send a Del event to the parent process, and transition back to Nothing state. If a protected password identifier was included in its SAE Commit message and either there is no protected password identifier in the received SAE Commit message or the protected password identifier differs from that used to construct the protocol instance’s SAE Commit message, BadID shall be set, the protocol instance shall send a Del event to the parent process, and transition back to *Nothing* state.

*Instruct the editor to modify C.3 as indicated:*

**C.3 MIB detail**

Dot11RSNAConfigPasswordValueEntry ::=

SEQUENCE {

dot11RSNAConfigPasswordValueIndex Unsigned32,

dot11RSNAConfigPasswordCredential OCTET STRING,

dot11RSNAConfigPasswordIdentifier OCTET STRING,

dot11RSNAConfigPasswordPeerMac MacAddress

dot11RSNAConfigPasswordPeerPubKey OCTET STRING

dot11RSNAConfigPasswordPubKeyGrp INTEGER

dot11RSNAConfigPasswordPubKeyLocked TruthValue}

dot11RSNAConfigPasswordPeerMac OBJECT-TYPE

SYNTAX MacAddress

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This is a control variable.

It is written by an external management entity.

Changes take effect as soon as practical in the implementation.

This variable represents the MAC address of the peer

that is to be authenticated. A wildcard BSSID is

permitted when passwords are shared among peers or

when password identifiers are used to identify the password."

::= { dot11RSNAConfigPasswordValueEntry 4}

dot11RSNAConfigPasswordPeerPubKey OBJECT-TYPE

SYNTAX OCTET STRING

MAX-ACCESS read-write

STATUS current

DESCRIPTION

“This is a control variable.

It is written by an external management entity.

Changes take effect as soon as practical in the implementation.

This variable represents the x-coordinate of a public key as

output by the integer to octet string procedure in 12.4.7.2.2.”

:: = { dot11RSNAConfigPasswordPeerPubKey 5}

dot11RSNAConfigPasswordPubKeyGrp OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

“This is a control variable.

It is written by an external management entity.

Changes take effect as soon as practical in the implementation.

This variable refers to a finite cyclic group from an

IANA-maintained registry for IKE (IETF RFC 2409).”

:: = {dot11RSNAConfigPasswordPubKeyGrp 6}

dot11RSNAConfigPasswordPubKeyLocked OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

“This is a control variable.

It is written by an external management entity.

Changes take effect as soon as practical in the implementation.

This variable indicates whether trust in public keys is

gained through their use or only trusted through pre-provisioning.”

:: = {dot11RSNAConfigPasswordPubKeyGrp 7}

**References:**