IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Privacy Protection for SAE Credentials | | | | |
| Date: 2025-02-27 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Dan Harkins | HPE |  |  |  |
| Michael Montemurro | Huawei |  |  |  |
| Jouni Malinen | Qualcomm Technologies, Inc. |  |  | jouni@qca.qualcomm.com |
|  |  |  |  |  |

Abstract

This submission proposes a mechanism to neet the approved P802.11bi requirement 1 to prevent an eavesdropper distinguishing whether authentication exchanges between CPE Client and CPE AP use identical SAE credentials or distinct SAE credentials by encrypting the SAE password identifier over air using HPKE.

r1: Editorial fixes based on discussion during the January 2024 meeting Mon AM2 slot.

r2: Adds the requested NOTEs describing possibility of automatic learning of the public key to avoid preprovisioning on a non-AP STA and an example protocol for managing shared public keys for APs in an ESS; and updates changes to be against REVme/D5.0.

r3: Update changes to be against REVme/D6.0 and P802.11bi/D0.4.

r4: Update changes to be against P802.11bi/D1.0.

**Discussion**

TGbi has approved a document describing the requirements for the project:

<https://mentor.ieee.org/802.11/dcn/21/11-21-1848-16-00bi-requirements-document.docx>

This contribution proposes draft text changes to address Requirement 1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Requirement** | **Issue / Use Case Reference** | **Status** | **Information** |
| 1 | 11bi shall define a mechanism to prevent an eavesdropper distinguishing whether authentication exchanges between CPE Clients and CPE AP use identical **SAE credentials** or distinct SAE credentials (where a CPE AP supports multiple SAE credentials). | I1, I5 | Approved | Proposed - 22/107r2  (9 March 2022)  To be motioned –agreed by unanimous consent 4/21/2022  **Approved** (Motion #13, 13 May 2022) |

|  |  |
| --- | --- |
|  | **Issues/Use Cases** |
| I1 | **Protecting password identifiers** |
| I5 | **Protecting authentication identifiers and key 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 despite them being useful and solving legitimate use cases.

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/ESS.

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 (or two during a rekeying period);
* When the APs in an ESS share the same public key, non-AP STAs need to manage a single public key for the ESS;
* Identities are protected against members of the same group.

These properties meet the requirements approved for the TGbi Requirement 1.

**Proposed changes**

*Add the following to Clause 2:*

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

*Insert the following new rows to Table 9-70 (P802.11bi/D1.0 P38 L24:*

**9.3.3.11 Authentication frame format**

**Table 9-70—Authentication frame body**

|  |  |  |
| --- | --- | --- |
| **Order** | **Information** | **Notes** |
| <ANA-2> | Protected Password Identifier | The Protected Password Identifier element is optionally present in Authentication frames as defined in Table 9-71 (Presence of fields and elements in Authentication frames). |
| <ANA-2b> | Privacy Public Key | The Privacy Public Key element is optionally present in Authentication frames as defined in Table 9-71 (Presence of fields and elements in Authentication frames). |

*Modify Table 9-71 as shown:*

**Table 9-71—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 element 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 the intented AKM is 00-0F-AC:24 or 00-0F-AC:25 (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. If the Status Code field is BAD\_PROTECTED\_IDENTITY, the Privacy Public Key element is optionally present; otherwise, it is not present. |

*Instruct the editor to obtain an assignment from ANA for <ANA-3>, modify Table 9-80 as indicated, and update the reserved value range:*

**Table 9-80—Status codes**

|  |  |  |
| --- | --- | --- |
| 143 | GAS\_QUERY\_REQUEST\_ TOO\_ LARGE | GAS query request is larger than the dot11GASQueryRequestLengthLimit value. |
| <ANA-3> | BAD\_PROTECTED\_IDENTITY | The SAE protected password identifier in the SAE Commit message was invalid. |
| 144-65 535 |  | Reserved. |

*Insert the following rows into Table 9-130 (P802.11bi/D1.0 P50 L60.*

**9.4.2 Elements**

**9.4.2.1 General**

**Table 9-130—Element IDs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Element ID** | **Element ID Extension** | **Extensible** | **Fragmentable** |
| Protected Password Identifier (see 9.4.2.X (Protected Password Identifier)) | 255 | <ANA-5> | No | No |
| Privacy Public Key (see 9.4.2.X+1 (Privacy Public Key)) | 255 | <ANA-6> | Yes | No |

*Instruct the editor to create following two subclauses as below at the end of 9.4.2 (i.e., just before 9.4.3; P802.11bi/D1.0 P63 L27), replacing X, XYZ, X+1, and XYZ+1 with the appropriate number and assigning the figure numbers appropriately:*

**9.4.2.X 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 (Protected Password Identifier element format).

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

Octets: 1 1 1 variable

**Figure 9-XYZ—Protected Password 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.

**9.4.2.X+1 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+1 (Privacy Public Key element format).

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

Octets: 1 1 1 2 variable

**Figure 9-XYZ—Privacy Public Key element format**

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 9.4.1.40 (Finite Cyclic Group field) 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 (Octet string to integer conversion) is the x-coordinate of an elliptic curve public key from the specified finite cyclic group.

*Insert two paragraphs into 12.4.1 (REVme/D7.0 P3001 L17) 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 public key of an AP or mesh STA. Public keys are preprovisioned with the password and password identifier. Public keys and the groups from which they are created are stored in dot11RSNAConfigPasswordPeerPubKey and dot11RSNAConfigPasswordPubKeyGrp, respectively.

NOTE—It is also possible for a non-AP STA to learn the public key of an AP during the first successful authentication with that AP. This allows the preprovisioning step of the public key for a non-AP STA to be avoided in cases where it is acceptable to perform the first SAE authentication using an unprotected password identifier. The mechanism for an AP to provide its current public key for successive SAE authentication exchanges is described in 12.4.3.

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

*Modify 12.4.3 (title and REVme/D7.0 P3002 L31) 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. To provide privacy, protected password identifiers can 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 are in the possession of the private key that corresponds to the privacy public key. 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 this standard.

NOTE 1—Control And Provisioning of Wireless Access Points (CAPWAP) protocol defined in IETF is one example of a possible approach for managing the shared public keys for APs in an ESS.

An AP may indicate the currently used privacy public key when rejecting authentication with status code BAD\_PROTECTED\_IDENTITY. This indication is unprotected.

An AP may indicate that a non-AP STA should move to using a new privacy public key by including the Privacy Public Key KDE in 4-way handshake message 3. This indication is protected. A non-AP STA that receives such indication as a part of a successfully completed 4-way handshake should store the received public key as the privacy public key for this ESS for consecutive SAE authentication instances.

NOTE 2—An AP might maintain two last used privacy public keys and allow either one to be used during SAE authentication. If a non-AP STA uses the older one of the public keys in SAE authentication, the AP might request the non-AP STA to move to using the newer public key during 4-way handshake. An AP might also request a non-AP STA to start using a privacy protection key if the non-AP STA did not use protected password identifier during SAE authentication. This enables protected key management for rekeying the privacy public key.

A STA protects SAE password identifiers by first obtaining the public key of an AP or peer mesh STA. If there is a public key in dot11RSNAConfigPasswordPeerPubKey, the STA uses that value as the x-coordinate of an elliptic curve defined by dot11RSNAConfigPasswordPubKeyGrp. 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 public key to wrap its password identifier using HPKE (IETF RFC 9180) with a compressed output KEM 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 Password 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 compressed output
* 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).An AP or peer mesh STA that receives a Protected Password Identifier element in an SAE Commit message shall unwrap it using HPKE in the “single shot” mode of decryption to its public key. The AAD used in the HPKE operation shall be the scalar field of the SAE Commit message. 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 this octet and the corresponding padding octets shall be removed, leaving the plaintext password identifier used to complete the SAE protocol.

*Modify 12.4.4.2.3 (REVme/D7.0 P3007 L4) 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 plaintext password identifier that was wrapped to or unwrapped 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.

...

*Modify 12.4.4.3.3 as indicated at REVme/D7.0 P3010 L61:*

**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 plaintext password identifier that was wrapped to or unwrapped 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.

*Modify 12.4.5.3 (REVme/D7.0 P3012 L13) 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 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.

*Modify 12.4.5.4 as indicated (REVme/D7.0 P3012 L26):*

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

If the peer’s SAE Commit message contains a Password Identifier element, the identifier in that element shall be used in construction of the PWE for this exchange. If the peer’s SAE Commit message contains a protected password identifier, the corresponding plaintext password 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, may include Privacy Public Key element indicating the currently used privacy public key in that SAE Commit message, and shall fail authentication.

NOTE 1—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.

NOTE 2—A STA that receives an SAE Commit message with a status code of BAD\_PROTECTED\_IDENTITY and Privacy Public Key element might determine which public key is used in the ESS.

*Modify 12.4.8.6.3 as indicated (REVme/D7.0 P3023 L7):*

**12.4.8.6.3 Protocol instance behavior—Nothing state**

Upon receipt of a *Com* event, the protocol instance shall check the Status Code field of the SAE Authentication frame.

* If the Status Code field is not SUCCESS or SAE\_HASH\_TO\_ELEMENT, the protocol instance shall silently discard the frame and send a *Del* event to the parent process.
* Otherwise, the password identifier or protected password identifier, if any, shall be checked:
  + If a password identifier is present and no password is associated with that identifier, the protocol instance shall set *BadID* and construct and transmit an SAE Authentication frame with status code UNKNOWN\_PASSWORD\_IDENTIFIER, and send a *Del* event to the parent process. If a protected password identifier is present, the protocol instance shall unwrap and unpad it. If unwrapping or unpadding fails, the protocol instance shall set *BadID*, construct and transmit an Authentication frame with status code BAD\_PROTECTED\_IDENTIFIER, and send a *Del* event to the parent process. If unwrapping succeeds, the unwrapped and unpadded data becomes the password identifier for this transaction. If no password is associated with that identifier, the protocol instance shall set *BadID*, construct and transmit an SAE Authentication frame with status code UNKNOWN\_PASSWORD\_IDENTIFIER, and send a *Del* event to the parent process.

*Modify 12.4.8.6.4 as indicated (REVme/D7.0 P3024 L24):*

**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, the protocol instance shall set *BadID*, 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, the protocol instance shall set *BadID*, send a *Del* event to the parent process, and transition back to *Nothing* state.

**12.7.2 EAPOL-Key frames**

*Modify Table 12-10 (KDE selectors) by adding a row as indicated (P802.11bi/D1.0 P115 L18):*

|  |  |  |
| --- | --- | --- |
| **OUI** | **Data type** | **Meaning** |
| 00-0F-AC | <ANA> | Privacy Public Key KDE |

*Add following after the definition of the PGTK KDE in 12.7.2 (P802.11bi/D1.0 P115 L46:*

The format of the Privacy Public Key KDE is shown in Figure 12-50j (Privacy Public Key KDE format).

|  |  |
| --- | --- |
| Finite Cyclic Group | Privacy Public Key |

Octets: 2 variable

**Figure 12-50j—Privacy Public Key KDE format**

The Finite Cyclic Group field and Privacy Public Key field are defined in 9.4.2.X (Privacy Public Key element).

**12.7.6.4 4-way handshake message 3**

*Modify 12.7.6.4 (REVme/D7.0 P3113 L65) as indicated:*

Key Data =

…

— The RSNXE that the Authenticator sent in its Beacon or Probe Response frame, if this element is present in the Beacon or Probe Response frame that the Authenticator sent.

— The SSID element containing the SSID of the BSS when both the Authenticator and the Supplicant have indicated support for SSID protection in the RSNXE.

— When SAE is used, optionally contains the Privacy Public Key KDE.

*Modify C.3 (REVme/D7.0 P5563 L49) as indicated:*

**C.3 MIB detail**

Dot11RSNAConfigPasswordValueEntry ::=

SEQUENCE {

dot11RSNAConfigPasswordValueIndex Unsigned32,

dot11RSNAConfigPasswordCredential OCTET STRING,

dot11RSNAConfigPasswordIdentifier OCTET STRING,

dot11RSNAConfigPasswordPeerMac MacAddress,

dot11RSNAConfigPasswordPeerPubKey OCTET STRING,

dot11RSNAConfigPasswordPubKeyGrp INTEGER }

*Modify C.3 (REVme/D7.0 P5564 L42) as indicated:*

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."

REFERENCE "IEEE Std 802.11-2024, 12.4.7.2.2"

:: = { dot11RSNAConfigPasswordValueEntry 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)."

:: = { dot11RSNAConfigPasswordValueEntry 6 }