IEEE P802.11
Wireless LANs

|  |
| --- |
| Resolution of a Few Security Comments |
| Date: 2013-05-15 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Dan Harkins | Aruba Networks | 1322 Crossman ave, Sunnyvale, CA | +1 408 227 4500 | dharkins at aruba networks dot com |
|  |  |  |  |  |

Abstract

The submission proposes simple resolutions to CIDs: 1373, 1374, 1375, 1376, 1383, 1384

CID 1373

(TR) Clause 11.11.1, p. 81, l. 23-24, fourth dash: the dot11RSNAConfigDLCGroupTable refers to the IANA groups in RFC 2409 (see also 802.11-2012, Clause 11.3.4.1). These are, however, not all created equal! Some of the DLP groups in that registry are not secure (e.g., #1, #3, #4), and some are DLP groups with so-called “truncated exponents” (the MODP groups #2, #5, #14 - #18), which explicitly violates NIST requirements on DLP groups, as stipulated in NIST SP 800-56a. While it remains to be seen whether these groups are even secure with a simple Diffie-Hellman protocol, these MODP groups are certainly completely flawed if used with a signature scheme and, thereby, are completely unsuitable for use with the FILS protocol without TTP (contrary to what this clause suggests). Frankly, only the DLP groups #22-#24 are based on well-studied cryptography, as are the NIST elliptic curve groups #8-#13, #19-#21, #25-#26. IEEE 802.11 should maintain its own table of suitable DLP groups (rather than relying on an external body, who could “Trojan-horse” things in or take things out). An extra argument for this is that IETF has taken the stance that this table should be grand-fathered.

**Commenter’s suggested remedy:** IEEE should refer to its own table of acceptable groups to be used with FILS authentication. For “FILS authentication without TTP”, this table should only include NIST compatible groups (i.e., #8-#13, #19-#26) and explicitly forbid use of the (insecure for this purpose) MODP groups. For “FILS authentication with TTP and PFS”, some people might want to take their chances and use unproven MODP schemes (I would prefer not, since cryptographers should take a conservative stance), but – frankly – this would hurt interoperability and drive-up implementation cost (since too many options without clear justification).

**Proposed remedy**: Reject. These groups are secure for their intended use-- Diffie-Hellman. Nothing "remains to be seen", they have received extensive evaluation as part of IKE and there are no problems using these groups in the way that FILS uses them. There is nothing in FILS that says to use keys from these groups for the purpose of digital signature and futhermore, there is no reason to require that the same group be used for D-H and digital signatures. This takes away from "cryptographic agility" and is a radical departure from the way that security protocols have been traditionally designed (viz, IKE and TLS).

CID 1374

(TR) Clause 11.11.1, p. 81, l. 23-24: Some of the DLP groups in the dot11RSNAConfigDLCGroupTable are enormous and can be expected to fall far short of reaching the time latency requirements for FILS (after all, the “F” stands for “Fast”!).

**Commenter’s suggested remedy:** Ditch the MODP groups with modulus size larger than 3072 bits, since these have no reasonable chance meeting the time latency requirements in 802.11ai ‘s charter.

**Proposed remedy**: Reject. There is no justification for this statement. If one does not wish to use a large MODP group because his hardware is inadequate to do modular exponentiation in a reasonable amount of time then he is free to use other groups. But it is not appropriate to prohibit others from using the groups if they wish (and have adequate hardware).

CID 1375

(TR) Clause 11.11.2.2.2, p. 83, l. 54-55: The specification leaves considerable freedom as to which DLP group is to be used for Diffie-Hellman key agreement and which group for signing messages during key confirmation (Clause 11.11.2.4). This “laissez-faire” policy may lead to insecure combinations (see my earlier comment on unsuitable of using MODP groups for e.g., DSS). It may also lead to extremely high implementation cost, due to the need to support widely varying groups with different underlying arithmetic and need for group-specific side channel secure implementations for these, and to far higher computational complexity than would be required if the DLP group used with key agreement would have to be the same as that used with signing. As case in point, if one uses ECDH and ECDSA with the same elliptic curve group, one can employ implementation tricks that make the expensive burden of ECDSA signature verification almost go away.

**Commenter’s suggested remedy:** Stipulate that, with “non-TTP” FILS protocols, one has to use the same DLP group for key agreement and signing.

**Proposed remedy**: Reject. This "laisseq-faire" policy is called "cryptographic agility" and it is the way that modern security protocols are designed. There is massive deployment of protocols that use these groups so any fear over misconfiguration or high implementation cost is Fear-Uncertainty-and-Doubt (FUD) not borne out of experience. There is no justification in diverging from this well-established practice and forcing the same group to be used for both D-H and DSS.

CID 1376

(TR) Clause 11.11.2.4, p. 84, l. 2: The element-to-octet-string conversion rule in Clause 11.3.7.2.4 seems overly restrictive (at least with elliptic curve points), since only allowing for so-called “affine” representation of elliptic curve points and, ruling out other representations that may have considerable performance benefits (not just now, but also in the future). Recent crypto speed papers have shown that flexibility of representation scheme may have a significant impact on computational complexity of scalar point multiplication. Moreover, the conversion rule in Clause 11.3.7.2.4 completely deviates from standard conversion rules used almost anywhere else, thus imposing a hard to justify hurdle in reuse of existing elliptic curve routines.

**Commenter’s suggested remedy:**  Use more conventional element-to-octet-string conversion rules, as used with, e.g., IETF 5480 and SEC1. (This would still allow implementation of inflexible schemes, such as the one in Clause 11.3.7.2.4, by just preprocessing a received SAE elliptic curve point (X || Y) and transforming this to ( 0x04 || X || Y) and then feeding this into the generally used conversion routine. Thus, nothing is lost, but flexibility due to option to define a new “control byte” is gained.)

**Proposed remedy**: Reject. Having implemented this already I am unaware of what "generally used conversion routine" requires an octet to infer compression or not. There is no hurdle, much less a hard-to-justify one.

CID 1383

(TR) Clause 11.11.2.2.1, p. 82, l. 16: The difference between the TTP without PFS scenario and the one with TTP and with PFS seems to be marginal: both require STA and AP to have a common TTP, who then distributes keying material to either in an inline fashion, after performing an authentication check on forwarded contributions by STA and AP via an outside scope mechanism that runs between AP and TTP. The only difference is that with the PFS-scheme, the key is computed using a public-key based mechanism (ephemeral Diffie-Hellman) whereas with the non-PFS-flavor this is done using a symmetric-key scheme. This difference is so marginal that it is hard to justify making a distinction here and one might as well always use the PFS flavor.

**Commenter’s suggested remedy:** Remove the “TTP without PFS” scheme.

**Proposed remedy**: Reject. PFS is not just some adjective, it is a very valuable property for a key exchange to have, especially one that uses very long-term secrets. PFS is invaluable for FILS and the suggestion to remove it is without merit.

CID 1384

(TR) Clause 11.11.2.4, p. 86, l. 4-5: The mechanism by which key confirmation messages are encrypted-and-authenticated is highly inflexible, since one always encrypts all information element in the Association Request and Response payloads, rather than allowing some flexibility in terms of which information elements are to be encrypted and authenticated, resp. only authenticated. In the January 2013 session, Qualcomm and Aruba Networks people made the argument that the vendor-specific element might need to be visible in some deployment scenarios, e.g., for network monitoring and management purposes. Thus, in that case enciphering should happen as an OFF-ON-OFF pattern that is currently not provided.

**Commenter’s suggested remedy:** Implement the authenticated encryption scheme in a more flexible fashion, along the lines of 13/311r1.

**Proposed remedy**: Reject. No such flexibility is needed.

**References:**