IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 802.11  Client Privacy discussion | | | | |
| Date: 2019-03-13 | | | | |
| Author(s): | | | | |
| Name | Company | Address | Phone | email |
| Thomas Derham | Broadcom |  |  | [thomas.derham@broadcom.com](mailto:thomas.derham@broadcom.com) |
| Nehru Bhandaru | Broadcom |  |  |  |

**Abstract**

This document provides discussion related to LB comments with CID 2689 (and also 2690, 2590).

**Introduction**

There are various indicators sent in-the-clear in 802.11 networks that might be used by a third-party adversary (i.e. neither the client device / end-user, nor the network operator) for fingerprinting and persistent tracking of a client device, giving rise to privacy concerns.

The recent 11aq amendment defines MAC address randomization for a non-AP STA, whereby the STA might use a different randomized MAC address when sending Probe Request frames and each time it visits a network (associates with an ESS). This allows to mitigate privacy concerns arising from the permanent use of a static MAC address (sent in-the-clear in 802.11 MAC headers), whereby a third-party passive observer could uniquely identify the device.

However, there are various other indicators within fields and elements sent during the association process that could also enable tracking across multiple visits to the network, including:

1. PMKID – A STA and AP might cache a PMK for an extended time period (hours, days or more) and use PMKSA caching for fast authentication each time the STA visits the network. The cached PMKID is included in-the-clear in (Re)Association Request frame or FILS Authentication frame, and also in M1 of the 4-way handshake or FILS Authentication frame response. The PMKID uniquely identifies the device for the duration that the PMK is cached.
   1. (Note: Current REVmd (11aq) text requires STA to return to its previous MAC address when using PMKSA caching; we might also discuss if this is absolutely necessary, although it is not discussed in this document).
2. SAE Password Identifier – A non-AP STA uses an SAE Password Identifier as a static identifier of an SAE password that is used by a group of users/devices (typically authentication using different identifiers/passwords results in access to different network resources such as VLANs). The SAE Password Identifier is sent in-the-clear in Authentication frames. Since the group of users/devices using the identified password may be small (e.g. one), in some use cases it might be a quasi-identifier of a device/user. In addition, depending on how the network administrator wishes to generate/manage credentials, the identifier string itself might include explicitly personally identifiable information such as an account username.
3. SSID – A non-AP STA may be configured to associate with one of several SSIDs advertised by a network (typically different SSIDs provide access to different network resources such as VLANs). The SSID is included in-the-clear in (Re)Association Request frames. Since the number of users/devices using the specific SSID may be small (e.g. one), in some use cases it might be a quasi-identifier of a device/user.
4. Presence of unusual/characteristic elements and/or values of fields – The presence and/or values of certain fields and elements (e.g. capability support indications) may provide sufficient entropy to be a “fingerprint” quasi-identifier of a device/user.

It should be noted that certain fundamental fields in the MAC header might also provide identifiable information in some use cases. For example, the BSSID might be a quasi-identifier if the number of non-AP STAs that frequently associate with a particular BSS/AP is known to be very small. Therefore, it may not be reasonable to fully guarantee that there is no information sent in-the-clear that could implicitly assist identification of a user in all use cases. However, it would be beneficial to take reasonable steps to minimize such information, to the extent that attempts to track users become significantly harder and less reliable, and therefore less valuable.

**High-level objectives**

High level objectives are proposed as follows:

* Minimize in-the-clear transmission of explicit personally identifiable information (including that which might be, for convenience or other reasons, incorporated in field values that might be freely chosen by the end-user or network administrator, such as network credentials)
* Minimize transmission of values that (might) remain static across multiple associations to the same ESS, and therefore might be used as quasi-identifiers
  + Focus especially on values that are expected to provide the highest entropy for reliable identification of the same user/device across the multiple associations to the ESS
  + Values that (might) remain static as the STA contiguously roams between APs within an ESS might be considered of lesser concern (nice-to-have) since the privacy impact is relatively narrow (limited to short-term tracking of the location of a device within the limited coverage area of the contiguous ESS) and the infrastructure required by the adversary to perform such tracking is relatively complex

- Mitigate active attacks (e.g. rogue AP, MiTM relay, …) that attempt to obtain explicit personally identifiable information or quasi-identifiers from a STA

It is noted that authentication encryption of data and robust management frames is sufficient to provide strong protection against both passive and active attacks for information sent between STA and AP post-association. Therefore, the focus of this document is on information contained in pre-association frames.

**Comparison of solution approaches**

There are multiple potential approaches to addressing the above issues; a comparison is provided below:

|  |  |  |
| --- | --- | --- |
| **Approach** | **Pros** | **Cons** |
| (1) Dynamic Pseudonyms (for potential identifiers) | - No new crypto | - Complexity of pseudonym storage (e.g. AP needs to store pseudonyms it has dynamically generated for all STAs – similar to AAA server for EAP-AKA)  - Requires trusted distribution of new pseudonyms generated by AP to STAs (e.g. in previous association)  - Complexity of pseudonym management in network (either APs in network need to coordinate/synchronize pseudonyms, or need to manage separate pseudonyms for each AP)  - Downgrade attack mitigation (esp. with need to handle out-of-sync scenarios)  - Only applies to identifiers for which the AP has explicitly generated and provided a pseudonym |
| (2a) Identifier masking using salted hash with salt generated by STA | - Simple crypto  - Avoids complexity/state of distributing/managing pseudonyms  - No (or fewer) downgrade attack issues  - Possibility to bind to MAC randomization (use randomized MAC as the salt) | - AP complexity/scalability (AP needs to recalculate hash for all possible identifier values in real-time when responding to a frame from a STA with a new salt)  - Only applies to identifiers for which all possible values are known by AP, and does not scale to large number of such candidate values |
| (2b) Identifier masking using salted hash with salt generated by AP | - Simple crypto  - Avoids complexity/state of distributing/managing pseudonyms  - No (or fewer) downgrade attack issues  - Somewhat reduced AP complexity (AP can decide when to update salt and therefore when to recalculate hashes for the possible identifier values) | - No migitation against active attacks (Rogue AP could deliberately not update its salt and cause STA to reveal a constant value for the identifier multiple times)  - Reduced privacy (STA cannot control when the salt is updated, and therefore might send same value for an identifier multiple times)  - Only applies to identifiers for which all possible values are known by AP, and does not scale to large number of such candidate values |
| (3a) Public-key encryption of (most of) pre-association frame bodies | - Avoids complexity/state of distributing/managing pseudonyms  - No (or fewer) downgrade attack issues  - Generic solution (can encrypt entire frame bodies in general) to address fingerprinting based on all elements in auth/assoc frames | - More crypto, somewhat higher computational complexity (could include DoS mitigation similar to anti-clogging token)  - Requires trusted distribution of AP’s (static) public key to STAs (e.g. in previous association) or existing authenticated keying material |
| (3b) Shared-secret probabilistic encryption of (most of) pre-association frame bodies (static secret per AP/network) | - Avoids complexity/state of distributing/managing pseudonyms  - No (or fewer) downgrade attack issues  - Generic solution (can encrypt entire frame bodies in general) to address fingerprinting based on all elements in auth/assoc frames | - Somewhat more crypto (but less than 3a)  - Requires secret distribution of AP/network’s shared secret to STAs (e.g. in previous association)  - Reduced security against previous-insider adversary (Rogue AP that has knows the AP’s shared secret could decode the frame to reveal identifiers) |
| (3c) Shared-secret probabilistic encryption of (most of) pre-association frame bodies (unique secret per AP-STA pair) | - Avoids complexity/state of distributing/managing pseudonyms  - Generic solution (can encrypt entire frame bodies in general) to address fingerprinting based on all elements in auth/assoc frames | - Somewhat more crypto (but less than 3a)  - Complexity of shared secret storage (e.g. AP needs to store shared secrets it has dynamically generated for all STAs)  - Requires secret distribution of shared secret from AP to STA (e.g. in previous association)  - Complexity of shared secret management in network (either APs in network need to coordinate/synchronize secrets, or need to manage separate secrets for each AP)  - Downgrade attack mitigation (esp. with need to handle out-of-sync scenarios) |

**Outline example**

In order to provide a concrete example, an outline of a possible approach based on option (3a) is described as follows.

(1) All elements in Authentication and (Re)Association Request/Response frames are masked by performing probabilistic encryption (e.g. AEAD in AES-SIV mode with nonce). Ideally fixed fields (e.g. FC group) could be protected/authenticated too. The same encryption key is used for all Authentication and (Re)Association Request/Response frames in a single association procedure. Possible approaches to deriving the encryption key include:

* (a) static-ephemeral Diffie-Hellman
  + The encryption key is derived from a Diffie-Hellman shared secret generated from the AP’s static public key and the STA’s ephemeral private key
  + The STA generates a new ephemeral private key when it initiates authentication/association, and provides it to the AP in an element that prepends the encrypted contents of the frame.
  + The AP’s static public key is obtained by the STA using one of several possible mechanisms, such as post-association protected-dual ANQP
    - If the STA has not yet authenticated the AP’s public key (e.g. it only obtained it by pre-association ANQP), the frame contents are protected from passive observation but not from active attack (where active attacker convinced the STA to use its public key). If the STA has authenticated the AP’s public key (e.g. using protected-dual ANQP after associating to the AP for the first time, or by some out-of-band mechanism), then the frame contents are protected from both passive and active attacks.
* (b) PASN (11az)
  + The encryption key is derived from a Diffie-Hellman shared secret generated from ephemeral AP and STA key exchange and an existing cached PMK (from a previous RSN authentication)
    - If the cached PMK exists between the STA and AP and is used with PASN, then the encryption key is authenticated and the frame contents are protected from both passive and active attacks. If not (i.e. because no cached PMK exists), then the frame contents are protected from passive observation but not from active attack.
* Note: anti-clogging mechanism similar to SAE / PASN can also be considered

1. Key Data field of 4-way handshake (which may contain PMKID/RSNE) is protected as follows:

* When PMKSA caching is performed, the PMKID is included in the encrypted (Re)Association Response frame instead of being unencrypted in 4-way M1.
* Key Data field (containing RSNE) of 4-way M2 is encrypted (using KEK).
* When NIST AES Key Wrap is used in 4-way handshake, a random nonce is appended so that the wrapped key is not deterministic (could be applies to group key handshake too)

A similar approach could be taken in principle for individually-addressed Probe Request/Responses, other non-protected ANQP/GAS frames, etc. For broadcast probe requests that cannot reasonably be protected, a different approach is needed such as suppressing any potentially identifiable information.