March 2022

QUANTUM RESISTANT RADSEC FOR OPENROAMING AND WIRELESS BROADBAND APPLIANCE OPERATIONS

NIRANJAN M M

QUANTUM RESISTANT RADSEC FOR OPENROAMING AND WIRELESS BROADBAND APPLIANCE OPERATIONS

AUTHOR:
NIRANJAN M M

ABSTRACT

OpenRoaming allows a Wi-Fi device to automatically and securely connect/roam to any Wi-Fi access point that is supported by OpenRoaming Federation. OpenRoaming being adopted by emerging technologies such as Wi-Fi6, Wi-Fi7, 5G, etc., which in-turn provides automatic and seamless roaming across these technologies. Now OpenRoaming standard is the part of Wireless Broadband Alliance (WBA), and it is the foundation for "One Global Wi-Fi Network". OpenRoaming defines certificate based (i.e., PKI based) policy/access to the AAA server. Also, WBA defines RADSec i.e., Radius over TLS, which is also a PKI based method to provide authentication and encryption between access server (also called Network Access Server, NAS) and radius server. As OpenRoaming and RADSec operations are based on PKI methods, they are vulnerable to cryptanalysis attacks by quantum computing using Shor's or Grover's algorithms. The techniques presented herein propose method to establish quantum resistant RADSec session between Radius Server and Client, by extending the Radius protocol to allow Radius Server to distribute post-quantum identifier to the Radius Client. Subsequently, Radius Server and Radius Client will use the negotiated PQPSK ID to get the same post-quantum pre-shared key from Quantum Key Source, which will be used to derive TLS keys for encryption/decryption of the traffic between Radius Server and Client.

DETAILED DESCRIPTION

OpenRoaming allows a Wi-Fi device to automatically and securely connect/roam to any Wi-Fi access point that is supported by OpenRoaming Federation. Federation members include wireless access providers (like service providers, managed service providers and property owners with guest Wi-Fi service), identity providers (like Mobile Virtual Network Operators [MVNOs], social networks, and consumer brand companies) and equipment makers (like handset and networking companies).

OpenRoaming is a very flexible framework (a roaming standard) and can also support cellular, LoRaWAN (Long Range WAN), IoT etc., OpenRoaming being adopted by emerging technologies such as Wi-Fi6, Wi-Fi7, 5G, etc., which in-turn provides automatic and seamless roaming across these technologies (for example: roaming between Wi-Fi6 and 5G). With the ability to access more applications through more bandwidth and reliability, users will gain an enhanced wireless experience as they move between 5G and Wi-Fi6 networks. Now OpenRoaming standard is the part of Wireless Broadband Alliance (WBA), and it is the foundation for "One Global Wi-Fi Network".

With the above background, OpenRoaming defines certificate based (i.e., PKI based) policy/access to the AAA server. Also, WBA defines RADSec (Radius Security) i.e., Radius over TLS, which is also a PKI based method to provide authentication and encryption between access server (also called Network Access Server, NAS) and radius server. WBA also defines the use of

HUBs that act as proxy for AAA signalling to another AAA server. As OpenRoaming and RADSec operations are based on PKI methods, they are vulnerable to cryptanalysis attacks by quantum computing using Shor's or Grover's algorithms. In other words, public key (asymmetric encryption) methods used in OpenRoaming and RADSec are not immune to cryptanalysis attacks, whereas pre-shared key (symmetric encryption) methods are immune to these attacks. From the OpenRoaming with RADSec operations point of view, we need a quantum resistant key exchange mechanism to establish secure connection/session between Network Access Server (NAS), which acts as Radius Client and Radius server.

As we know, RADIUS uses UDP protocol as transport mechanism and uses MD5 Digest with pre-shared key to provide authentication and integrity. To address issues (reliability, packet size etc.,) of Radius over UDP, Radius over TCP is being adopted (https://tools.ietf.org/html/rfc6613) and to address security issues, Radius over TLS (https://tools.ietf.org/html/rfc6614) Or Radius over IPSec (https://tools.ietf.org/html/draft-ietf-radius-ipsec-00) tunnels are used between Radius Server and Client.

- When Radius over TLS is used, it uses pre-shared key (PSK) (need to be configured manually) for small-scale deployments and certificate based for large-scale deployments to provide authentication, integrity, and encryption. Radius protocol over TLS is also called RADSec protocol.
- When Radius over IPSec is used, it would use key exchange mechanism as per IKEv1 (https://tools.ietf.org/html/rfc2409) or IKEv2 (https://tools.ietf.org/html/rfc5996).
- New standard Post-quantum Pre-shared Keys for IKEv2 (https://tools.ietf.org/id/draft-fluhrer-qr-ikev2-03.xml) is used to provide quantum resistant connection establishment, but these standards assume that the pre-shared symmetric key is already in place when the key establishment session begins. It does not attempt to solve the symmetric key distribution problem (https://tools.ietf.org/id/draft-fluhrer-qr-ikev2-03.xml#n-assumptions).

Here, Radius over TLS (RADSec) is considered, which needs manual configuration of pre-shared keys or PKI based certificate authentication and encryption. Manual configuration is error prone, and PKI based methods suffers from cryptanalysis attacks by quantum computing using Shor's or Grover's algorithms. Hence, we need a method to make Quantum resistant RADSec TLS tunnel between Radius Server and Client in large OpenRoaming deployments.

The techniques presented herein propose method to establish quantum resistant RADSec session between Radius Server and Client, by extending the Radius protocol to allow Radius Server (which acts as Key-Server) to distribute post-quantum identifier (PQPSK ID) to the Radius Client. Subsequently, Radius Server (which acts as Key-Server) and Radius Client (which is a Non-Key-Server) will use the negotiated PQPSK ID to get the same post-quantum pre-shared key (PQPSK key) from Quantum Key Source (QKS), which will be used to derive TLS keys for encryption/decryption of the traffic between Radius Server and Client.

To avoid DoS attack on PQPSK ID, Key Server (KS) should include some authentication information along with PQPSK ID generated by the QKS such as

- Signature: Signed on PQPSK ID by Radius Server using private key (generated and distributed by WBA OpenRoaming Federation).
- QKS on Radius Server generates (PQPSK Key, PQPSK ID) pair, which can be unique per Radius Client.

- Radius Server would generate signature by signing on PQPSK ID using its private key and include that signature also as part of PQPSK ID TLV.
- Radius Server shares this consolidated PQPSK ID TLV to the Radius Client.
- Radius Client validates the signature using public key of Radius Server before accepting the PQPSK ID sent by the Radius Server.

A set of PQPSK TLVs are introduced to provide post-quantum security.
- PQPSK Capability TLV - For exchanging PQPSK capabilities to identify the ability for post-quantum security.
- PQPSK ID Information TLV - For sending PQPSK ID from Radius Server (Key-Server) to Radius Client (Non-Key-Server)
- PQPSK ID Status TLV - For sending PQPSK ID Status from Radius Server (Key-Server) to Radius Client (Non-Key-Server)

I. PQPSK Capability TLV:
Each Radius Client which supports post-quantum Pre-Shared Key (PQPSK) announces its capability as part of PQPSK capability TLV in Radius "Status-Server" packet (https://tools.ietf.org/html/rfc5997) by using TLV from radius protocol extension as per https://tools.ietf.org/html/rfc6929. Whenever Radius Server or Client is configured to support PQPSK as mandatory, it can continue to drop the received Radius packets from the unsupported peers.

II. Distribution of PQPSK ID information TLV (Type, Signature, PQPSK ID, Checksum) (Radius Server->Radius Client):
When both Radius Server and Client are PQPSK capable, then Radius Server (Key-Server) generates the PQPSK ID with the help of QKS and distribute to the Radius Client in PQPSK ID TLV in Radius "Access-Accept" packet which is sent in-response to Status-Server packet (https://tools.ietf.org/html/rfc5997) by using TLV from Radius protocol extension as per https://tools.ietf.org/html/rfc6929.

III. PQPSK ID Status information TLV (Radius Server->Radius Client):
This TLV is sent from Radius Server to Radius Client. When PQPSK Capability TLV is received by Radius Server from the Radius Client and if Radius Server does not support PQPSK functionality, then Radius Server sends back PQPSK status as 'Reject" in Radius "Access-Reject" packet. Upon receiving PQPSK status as "Reject", Radius Client would fallback to legacy TLS connection.

The techniques presented herein is explained in two phases as below:
Here, Network Access Server (NAS) acts as Radius Client. NAS could be WLC, SDWAN controller, 5G/LTE base station etc.,

Phase-1 (Share secret seed value between Radius Server and Radius Client):
- Generate McEliece public/private key-pair on both Radius Server and Radius Client.
- With WBA being the OpenRoaming Federation, it acts as root certificate authority to generate and distribute the keys to the Radius Server and Radius Client.
- Radius Client establishes classical secure TLS connection with the Radius Server (as part of RADSec protocol).

- If root certificate authority (OpenRoaming Federation) has not distributed public keys in earlier steps, then exchange public keys over secure connection.
- Radius Server acts as Key-Server (KS) and Radius Client acts as Non-Key-Server (NKS).
- Radius Server generates random secret seed value and encrypts using Radius Client's public key. This could be generated "one for each Radius client" or "common for all Radius Client".
- Radius Server transmits the encrypted secret seed value to Radius Client.
- This is sent in "Access-Accept" packet which is in-response to "Status-Server" packet
- Radius Client decrypts the seed using its private key.
- Now Radius Server and Radius Client have a common secret seed.

The Figure-1 shows common secret seed establishment using McEliece algorithm:

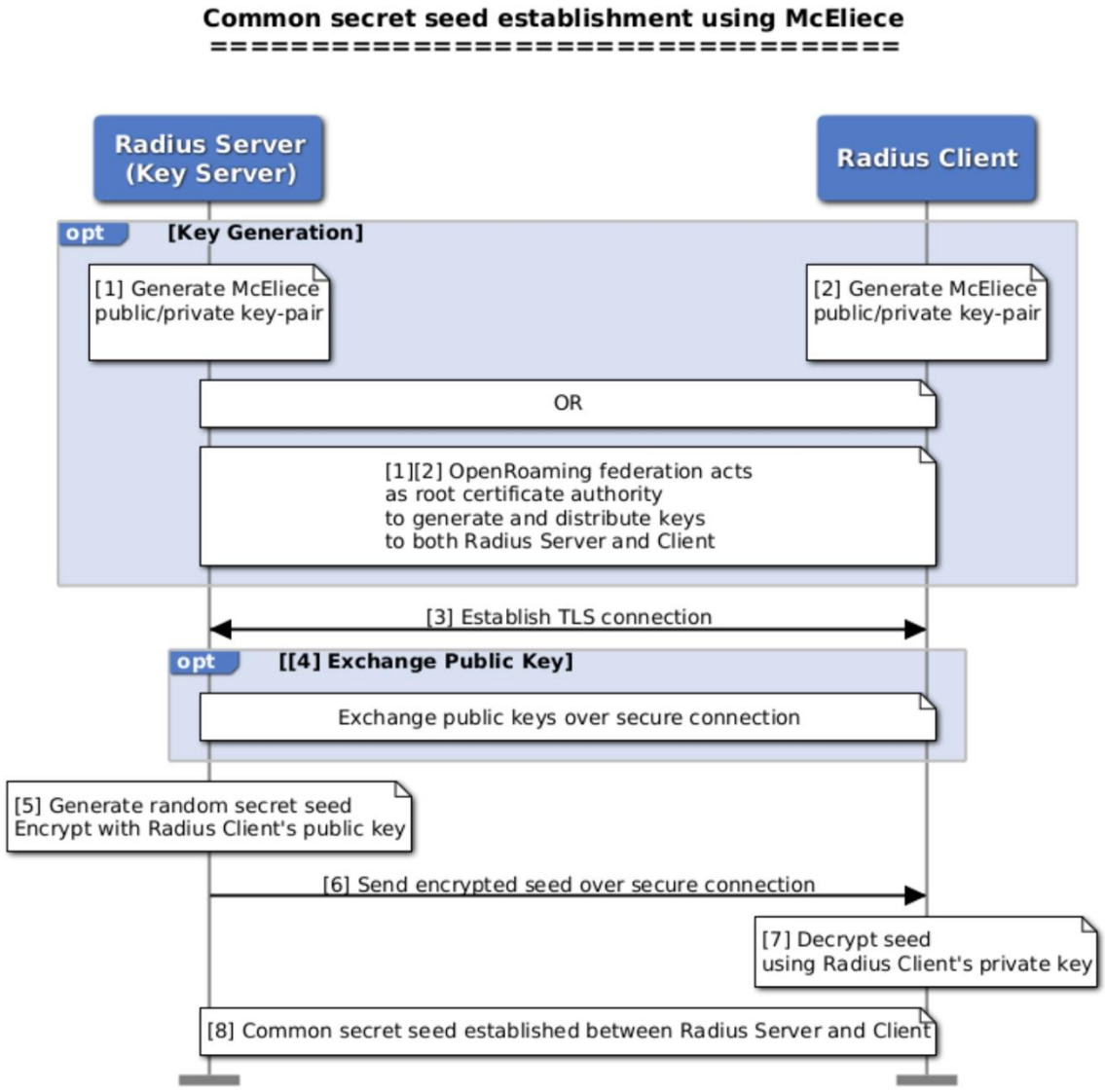

Figure-1

Phase-2 (Generate same pre-shared key using QKS):
Radius Server and Client will generate a (PQPSK Key, PQPSK ID) pair based on the common seed exchanged in Phase-1, using QKS, which is software entity implemented inside Radius Server and Client.

- Radius Server and Radius Client initialise QKS with the common seed secret exchanged in Phase-1.
- Radius Server (Key-Server) request QKS to generate first secret.
- QKS generates the secret key (PQPSK Key).
- A unique identity PQPSK ID is paired with each PQPSK Key generated.
- Radius Server fetches PQPSK Key and corresponding unique identity PQPSK ID from the QKS.
- Radius Server sends the identity PQPSK ID of the PQPSK Key to the Radius Client.
- This is sent in "Access-Accept" packet which is in-response to "Status-Server" packet
- Radius Client requests QKS to fetch PQPSK Key corresponding to the PQPSK ID received from the Radius Server.
- Since QKS of Radius Server and Client have started with the same seed secret, they will have the same PQPSK Key corresponding to the PQPSK ID.
- Now, Radius Server and Client will have same PQPSK Key, they can use this to derive the TLS key (e.g., AES-256 key etc.,) to establish TLS connection (part of RADSec protocol).
- Radius Client initiates a new TLS connection to Radius Server using above derived key.

Anytime Radius Server wants to refresh the TLS key, Radius Server and Client run the above procedure. Hence Radius Server and Client can refresh their keys at will, without any administrative intervention, and without ever exhausting their shared keys (since they are of generated ones).
The Figure-2 shows post-quantum secure connection establishment:

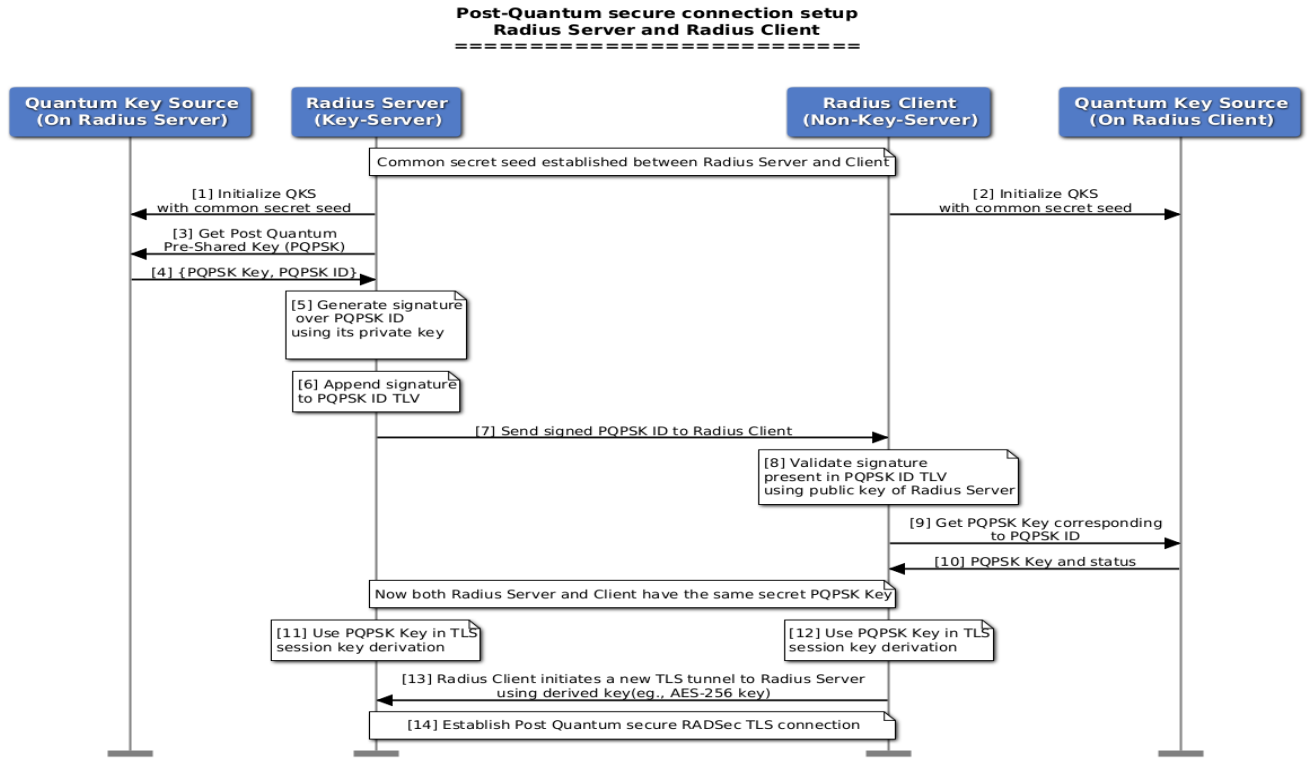

Figure-2

The techniques presented here provides quantum resistant secure RADSec session between Radius Server and Client. It allows frequent key refresh with no administrative cost. Moreover, this method provide quantum resistant secure RADSec session in OpenRoaming deployments. Additionally, it is generic and can be used between any Radius Server and Client to provide quantum resistant secure communication.