Using TEE and RV in PQ-TLS Communication [*]

Peter Spacek, Christian Colombo, and Mark Vella
peter.spacek@stuba.sk,{christian.colombo, mark.vella} @um.edu.mt

University of Malta and Slovak University of Technology in Bratislava

Security of the system is as strong as the security of the weakest point. Even in the recent years of post quantum cryptography development, we still need to think in a way that attacker would; If we use post quantum public key algorithms, and quantum-safe size of AES, the attacker may simply put a malware on the client PC and get his information in simpler way. With this in mind, we can use more modular design of security, where more features are performed under trusted execution environment (TEE). Most of the traffic on the web uses TLS to secure the communication. In ”Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH” by Eric Crockett, Christian Paquin, and Douglas Stebila, the post quantum (PQ) algorithms for handshake were introduced. But even if the keys are protected with stronger public key cryptography, the adversary can simply attack the key inside the client machine. In TLS protocol we can find ”sub-protocol” called TLS Record. It is securing the application data by using the keys and parameters established in the handshake. TLS Record is also responsible for verifying its integrity and origin. The recent version, TLS 1.3 removed all insecure options and uses only ChaCha20-Poly1305 or AES-GCM to produce a ciphertext of equal length. The ciphers are considered as Authenticated Encryption with Associated Data (AEAD). Based on the key and IV, it will ”hash” the ciphertext, additional data and lengths. Final MAC is this hash, encrypted, which is added to ciphertext. Lets set the experiment model; starting with the model presented by M. Vella and Ch. Colombo in ”Towards a Comprehensive Solution for Secure Cyptographic Protocol Execution based on Runtime Verification”, we build TLS client on the host PC (untrusted domain). Client received encrypted data in form of TLS (record) packet. The critical operation, AES decryption, can be performed in TEE. In our case, we choose SEcube (open source security-oriented hardware and software platform) to store session keys, and perform AES operations. Now, lets introduce the attacker; a malware present on the client computer. The most critical operations are moved to TEE, and we need to make sure that the communication between host and TEE is safe. For this reason, we use Runtime Verification (RV) to monitor and analyse communication, changes to plaintext and ciphertext, and key leakage. For a future research we can apply similar approach to some PQ algorithm, e.g. BIKE, what can be used as more secure PQ TLS-like protocol and also to protocol developed by NATO G5448 project.

[*] Supported by the NATO Science for Peace and Security Programme through project G5448 Secure Communication in the Quantum Era.