# Applied Cryptography

### Guest Lectures, Assignment 7, Monday, May 27, 2024

**Remarks:**

- Hand in your answers through Brightspace.
- Hand in format: PDF. Either hand-written and scanned in PDF, or typeset and converted to PDF. Please, **do not** submit photos, Word files, LaTeX source files, or similar.
- Assure that the name of **each** group member is **in** the document (not just in the file name).

**Deadline:** Sunday, June 9, 23.59

**Goals:** After completing these exercises you should have understanding in some techniques used in public-key infrastructures and TLS.

1. **(15 point)** In the first guest lecture given by Benoît Viguier, we took a look at the management of certificates for TLS and certificate authorities (CAs).
   (a) "Let's Encrypt" allows for easy issuing of certificates. Before issuing such certificate, some verifications need to be done to guarantee security. What are those verifications?
   (b) Name the protocol used by "Let's Encrypt" to perform the verifications of (1a).
   (c) Give a brief description in your own words of how the protocol in (1b) works.
   (d) Name one of the advantages of a multi-tier certificate authority.
   (e) Certificate Revocation Lists (CRLs) are part of the TLS1.3 protocol, name the biggest inconvenience with CRLs.
   (f) Name at least two problems solved by the Online Certificate Status Protocol (OCSP).

2. **(35 points)** In the lecture and in particular the guest lecture by Thom Wiggers, we took a close look at the TLS 1.3 protocol. In the future, such a protocol needs to be adapted to be quantum-secure. In this exercise, we go over several difficulties that might arise when adapting TLS to be quantum-proof.

Stripped down to its cryptographical components, the TLS 1.3 protocol can be given as follows, where $\mathsf{data}$ implies application data encrypted with a key derived from some $\mathsf{KDF}$.

| Alice's client | | Bob's server |
|---|---|---|
| $P, G$ | | $P, G, \mathsf{pk}_B, \mathsf{sk}_B$ |
| $x \overset{\$}{\leftarrow} \mathbb{Z}_q$, $X \leftarrow G^x$ | | |
| | $\xrightarrow{X}$ | |
| | | $y \overset{\$}{\leftarrow} \mathbb{Z}_q$, $Y \leftarrow G^y$<br>$\mathsf{ss} \leftarrow X^y$<br>$K \leftarrow \mathsf{KDF}(\mathsf{ss})$<br>$\sigma \leftarrow \mathsf{AE}_K($<br>$\mathsf{cert}_{\mathsf{pk}_B}$<br>$\mathsf{Sign}(\mathsf{sk}_B, \mathsf{transcript})$<br>$\mathsf{key\_conf}$<br>$)$ |
| | $\xleftarrow{Y, \sigma}$ | |
| $\mathsf{ss} \leftarrow Y^x$<br>$K, K', K'' \leftarrow \mathsf{KDF}(\mathsf{ss})$<br>$\mathsf{Verif}_K(\sigma)$<br>$\sigma' \leftarrow \mathsf{AE}_{K'}(\mathsf{key\_conf})$ | | |
| | $\xrightarrow{\sigma',\ \mathsf{AE}_{K''}(\mathsf{data})}$ | |
| | $\overset{\mathsf{data}}{\longleftrightarrow}$ | |

(a) Explain what Alice needs to do in $\mathsf{Verif}(\sigma)$

In post-quantum cryptography, Diffie-Hellman key exchange cannot be used (there is currently no post-quantum alternative), and so we resort to using key encapsulation mechanisms (KEMs), as we have seen in lecture 4.

(b) Explain which steps in this protocol need to be updated to achieve post-quantum security. In particular, describe the specific functions that need to be replaced by KEMs and post-quantum signatures.

(c) Use Kyber-768[1] as your key encapsulation mechanism and pick a post-quantum signature algorithm selected by NIST that you think fits, based on size and performance[2]. Assume 'pre-quantum' TLS 1.3 using X25519 for key exchange and RSA-2048 for signatures. Compare both size of transmitted data and clock cycles for the asymmetric cryptography data of this 'pre-quantum' TLS 1.3 and your 'post-quantum' TLS 1.3.

(d) Based on the increases you get, what kind of problems do you expect when naïvely switching to post-quantum TLS?

Alternatively, researchers have attempted to come up with modified versions of TLS that are less problematic in a post-quantum context. One of these is KEMTLS[3]. We can give the following schematic description.

[1] https://pq-crystals.org/kyber/index.shtml

[2] See an overview by Bas Westerbaan at https://blog.cloudflare.com/nist-post-quantum-surprise/

[3] Introduced in https://eprint.iacr.org/2020/534.pdf

| Alice's client | | Bob's server |
|---|---|---|
| $P, G$ | | $P, G, \mathsf{pk}_B, \mathsf{sk}_B$ |
| $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathsf{KEM.Keygen}()$ | | |
| | $\xrightarrow{\mathsf{pk}_e}$ | |
| | | $(\mathsf{ss}_e, \mathsf{ct}_e) \leftarrow \mathsf{KEM.Encaps}(\mathsf{pk}_e)$<br>$K_1, K_1' \leftarrow \mathsf{KDF}(\mathsf{ss}_e)$<br>$\sigma \leftarrow \mathsf{AE}_{K_1}(\mathsf{cert}_{\mathsf{pk}_B})$ |
| | $\xleftarrow{\mathsf{ct}_e, \sigma}$ | |
| $\mathsf{ss}_e \leftarrow \mathsf{KEM.Decaps}(\mathsf{ct}_e, \mathsf{sk}_e)$<br>$K_1, K_1' \leftarrow \mathsf{KDF}(\mathsf{ss}_e)$<br>$(\mathsf{ss}_B, \mathsf{ct}_B) \leftarrow \mathsf{KEM.Encaps}(\mathsf{pk}_B)$<br>$\sigma' \leftarrow \mathsf{AE}_{K_1'}(\mathsf{ct}_B)$ | | |
| | $\xrightarrow{\sigma'}$ | |
| | | $\mathsf{ss}_B \leftarrow \mathsf{KEM.Decaps}(\mathsf{ct}_\mathsf{B}, \mathsf{sk}_\mathsf{B})$ |
| $K_2, K_2', K_2'' \leftarrow \mathsf{KDF}(\mathsf{ss}_e \mid\mid \mathsf{ss}_B)$ | | $K_2, K_2', K_2'' \leftarrow \mathsf{KDF}(\mathsf{ss}_e \mid\mid \mathsf{ss}_B)$ |
| | $\xrightarrow{\mathsf{AE}_{K_2}(\mathsf{key\_conf}),\mathsf{data}}$ | |
| | $\xleftarrow{\mathsf{AE}_{K_2'}(\mathsf{key\_conf}),\mathsf{data}}$ | |
| | $\xleftrightarrow{\mathsf{data}}$ | |

(e) On what client-to-server message can Alice send encrypted application data in KEMTLS?

(f) Using the same schemes you picked before, calculate the size of transmitted data using KEMTLS.

(g) What are advantages of using KEMTLS compared to TLS, and what are the disadvantages?

(h) TLS has a long history of vulnerability to padding oracle attacks. Describe on a high level the idea behind these attacks.

(i) Is KEMTLS vulnerable to a padding oracle attack? Justify your answer.

(j) KEMTLS at first sight appears to completely avoid the usage of digital signatures. This is however not the case. Where and for what purpose are digital signatures used in KEMTLS? How many signatures in use can you count and which are they?