

Praktischer Teil: Breitbandtechnik

Sommersemester 2017

Motivation: Netzwerkverbindungen im Internet werden mittlerweile immer öfter abgesichert, das beste Beispiel dafür ist die zunehmende Migration von HTTP zu HTTPS.

Hier wird das unverschlüsselte HTTP-Protokoll durch einen sicheren Tunnel zwischen Client und Server geleitet, der mit dem TLS-Protokoll (Transport Layer Security, Nachfolger von SSL) aufgebaut wird.

TLS bietet eine große Anzahl von Ciphersuites an, d.h. Client und Server einigen sich auf Algorithmen für den Schlüsselaustausch, die Authentifizierung, die Chiffre und den Integritätsschutz und die verwendeten Schlüssellängen. Beispielsweise

	Schlüsselaustausch	Authentifizierung	Chiffre	Integritätsschutz
ECDHE-RSA-AES256-GCM-SHA384	ECDHE (Elliptische Kurven DHE)	RSA (Bitlänge nicht festgelegt)	AES 256 Bit (GCM-Mode)	SHA2-384
DHE-DSS-AES256-GCM-SHA384	DHE (DH mit temporären Schlüsseln)	DSS (Bitlänge nicht festgelegt)	AES 256 Bit (GCM-Mode)	SHA2-384
DH-RSA-DES-CBC3-SHA	DH (Diffie-Hellman mit statischen Schlüsseln)	RSA (Bitlänge nicht festgelegt)	DES 56 Bit (CBC-Mode)	SHA1

(Auszug aus `openssl ciphers`)

Mit der Existenz von Quantencomputern, die ausreichend Speicher (Qubits) zur Verfügung haben, wird sich die Welt der Kryptographie ändern. Herkömmliche asymmetrische Kryptographieverfahren wie RSA und Diffie-Hellman (DH, DHE, ECDH, ECDHE) werden in kurzer Zeit gebrochen werden können. Damit auch dann noch sichere Kryptographie betrieben werden kann, wurden Post-Quantum Computing (PQC) Verfahren entwickelt.

Es wird eine virtuelle Maschine mit einer Variante (<https://github.com/open-quantum-safe/openssl>) von OpenSSL zur Verfügung gestellt, die für den Schlüsselaustausch verschiedene PQC-Verfahren anbietet. Ihre Aufgabe ist die Erhebung der Performance, d.h. Dauer des Verbindungsaufbaus und Dauer der Datenübertragung für unterschiedliche Ciphersuites.

Hinweise zur Software:

- Als Server bietet sich OpenSSL (`openssl s_server -key ?.pem -cert ?.pem -accept <Port> -HTTP`) oder der Apache2-Webserver an.
- Als Client können Sie OpenSSL verwenden (`openssl s_client -connect localhost:4433 -www /? -time ? -new -cipher ?`). Dieser Client ist automatisierbar.
- Vorab müssen Sie für den Server einen privaten Schlüssel und ein Zertifikat generieren. (z.B. RSA2048: `openssl req -x509 -newkey rsa:2048 -keyout key_rsa.pem -out cert_rsa.pem -days 365` oder ECDSA256: `openssl ecparam -name secp256k1 -genkey -out param.pem; openssl req -x509 -newkey ec:param.pem -keyout key_ecdsa.pem -out cert_ecdsa.pem -days 365`).
- Die Aufzeichnung der Übertragungsdauer können Sie mit Wireshark durchführen.

Aufgabe: Wählen Sie aus dem folgenden Katalog das Thema für Ihre Projektarbeit aus (jedes Thema wird nur 1x vergeben).

- 1.) Performance-Analyse der Verbindungsaufbaus der folgenden PQC-Ciphersuites:
OQSKEK-*{RLWE-BCNS15, RLWE-NEWHOPE, RLWE-MSRLN16, LWE-FRODO-RECOMMENDED, SIDH-CLN16}*-ECDSA-AES128-GCM-SHA256 und
OQSKEK-*{RLWE-BCNS15, RLWE-NEWHOPE, RLWE-MSRLN16, LWE-FRODO-RECOMMENDED, SIDH-CLN16}*-ECDHE-ECDSA-AES128-GCM-SHA256
- 2.) Performance-Analyse des (1) Verbindungsaufbaus und der (2) Datenübertragung von 10 Mbyte mit folgenden Ciphersuites:
{ECDHE, ECDH, OQSKEK-RLWE-BCNS15, OQSKEK-RLWE-NEWHOPE, OQSKEK-RLWE-MSRLN16, OQSKEK-LWE-FRODO-RECOMMENDED, OQSKEK-SIDH-CLN16}-ECDSA-AES256-GCM-SHA384
- 3.) Performance-Analyse des (1) Verbindungsaufbaus und der (2) Datenübertragung von 10 Mbyte mit folgenden Ciphersuites:
{ECDHE, ECDH, OQSKEK-RLWE-BCNS15, OQSKEK-RLWE-NEWHOPE, OQSKEK-RLWE-MSRLN16, OQSKEK-LWE-FRODO-RECOMMENDED, OQSKEK-SIDH-CLN16}-RSA-AES256-GCM-SHA384
- 4.) Performance-Analyse des (1) Verbindungsaufbaus und der (2) Datenübertragung von 10 Mbyte mit folgenden Ciphersuites:
ECDHE-RSA-AES256-GCM-SHA384, ECDHE-RSA-AES256-SHA384, DHE-RSA-AES256-GCM-SHA384, ECDHE-RSA-AES128-GCM-SHA256, DHE-RSA-AES128-GCM-SHA256
- 5.) Performance-Analyse des (1) Verbindungsaufbaus und der (2) Datenübertragung von 10 Mbyte mit folgenden Ciphersuites:
DH-DSS-AES256-GCM-SHA384, DHE-DSS-AES256-GCM-SHA384, DH-DSS-AES128-GCM-SHA256, SRP-DSS-3DES-EDE-CBC-SHA, DH-DSS-DES-CBC-SHA
- 6.) Performance-Analyse des (1) Verbindungsaufbaus und der (2) Datenübertragung von 10 Mbyte mit folgenden Ciphersuites:
ECDHE-ECDSA-AES256-GCM-SHA384, ECDHE-ECDSA-AES256-SHA384, ECDH-ECDSA-AES128-GCM-SHA256, ECDHE-ECDSA-RC4-SHA, ECDH-ECDSA-DES-CBC3-SHA
- 7.) Performance-Analyse des Verbindungsaufbaus für die asymmetrischen Schlüssellängen: 512 Bit, 1024 Bit, 2048 Bit, 4096 Bit, 8192 Bit und der Ciphersuites:
DH-DSS-AES256-GCM-SHA384, DHE-DSS-AES256-GCM-SHA384, DH-RSA-AES256-GCM-SHA384, DHE-RSA-AES256-GCM-SHA384
- 8.) Performance-Analyse des Verbindungsaufbaus für die elliptische Kurven: *secp112r1, secp128r1, secp224r1, secp384r1, secp521r1* und der Ciphersuites:
ECDHE-ECDSA-AES256-GCM-SHA384, ECDH-ECDSA-AES256-GCM-SHA384

Erstellen Sie eine Arbeit nach den Maßstäben der Wissenschaft, d.h.

- Erarbeiten und Beschreiben der theoretischen Hintergründe.
- Erarbeiten Sie mit Hilfe von praktischen Versuchen eine Lösung. Wählen Sie dabei nur sinnvolle Versuchsaufbauten und Versuchsparameter. Sie können für jeden durchgeführten Versuch die folgenden Fragen beantworten:
 - Welche Erkenntnisse sollen durch den bevorstehenden praktischen Versuch ermittelt werden?
 - Wurden die Erkenntnisse durch den Versuch erbracht? Wie sind die Ergebnisse zu bewerten?
 - Ist die Anzahl der Versuchsdurchführungen statistisch relevant?
- Erarbeiten Sie ein abschließendes Fazit auf Basis der durchgeführten Versuche und der notwendigen Theorie.

Gruppengröße: max. 3 Personen

Abgabe: Die Auswertung ist bis zum 16.6.2017 (23:59) per Email einzureichen. Zusätzlich zu der Auswertung wird eine Abschlusspräsentation (25 Min) erwartet, in der die eigenen Ergebnisse und die Hintergründe erläutert werden. Der Präsentationstermin wird nach dem 16.6.2017 festgelegt. Auswertung und Präsentation gehen jeweils zu 50% in die Endnote ein.

Bei Fragen können Sie mich gerne kontaktieren:

Prof. Dr.-Ing. Andreas Noack

Haus 4, Raum 222

Tel.: 03831 45-6626

Email: andreas.noack@fh-stralsund.de