HAW HAMBURG

BACHELORTHESIS
Abdal Neda

Performance-Analyse hybrider Post-Quantenkryptografie-Algorithmen in TLS

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Abdal Neda

Performance-Analyse hybrider Post-Quantenkryptografie-Algorithmen in TLS

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Angewandte Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Klaus-Peter Kossakowski
Zweitgutachter: Prof. Dr. Martin Becke

Eingereicht am: 30. November 2023

Abdal Neda

Thema der Arbeit

Performance-Analyse hybrider Post-Quantenkryptografie-Algorithmen in TLS

Stichworte

Kryptografie, Quantencomputer, TLS, Post-Quantenkryptografie, Sicherheit

Kurzzusammenfassung

Diese Arbeit untersucht die Performance und praktische Anwendbarkeit von hybriden Kryptografie-Algorithmen, eine Kombination aus Post-Quantenkryptografie (PQC) und klassischer Kryptografie, im Transport Layer Security (TLS) Protokoll. Ziel der Arbeit ist es, zu evaluieren, ob hybride Ansätze gegenüber reinen PQC-Lösungen einen Mehrwert bieten. Dies umfasst die Bewertung, ob hybride Methoden nur die Komplexität der Systeme erhöhen oder tatsächlich effizienter im Kontext der derzeitigen Unsicherheiten rund um die Sicherheit von PQC sind. Die ermittelten Messergebnisse deuten darauf hin, dass die Verwendung hybrider Algorithmen insbesondere in der Übergangsphase zur vollständigen Implementierung von PQC eine robuste und effiziente Lösung darstellt. Die Ergebnisse stärken damit das Argument für die Verwendung von hybriden Lösungen als pragmatischen und sicheren Ansatz in der gegenwärtigen Landschaft der Kryptografie.

Abdal Neda

Title of Thesis

Performance analysis of hybrid post-quantum cryptography algorithms in TLS

Keywords

Cryptography, Quantum Computers, TLS, Post-Quantum Cryptography, Security

Abstract

This thesis investigates the performance and practical applicability of hybrid cryptography algorithms, a combination of post-quantum cryptography (PQC) and classical cryptography, in the Transport Layer Security (TLS) protocol. The aim of the work was to evaluate whether hybrid approaches offer added value compared to pure PQC solutions. This includes evaluating whether hybrid methods only increase the complexity of the systems or are actually more efficient in the context of the current uncertainties surrounding the maturity of PQC. The results suggest that the use of hybrid algorithms is a robust and efficient solution, especially in the transition phase towards full implementation of PQC. They strengthen the argument for the use of hybrid solutions as a pragmatic and secure approach in the current landscape of cryptography.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

PQC	Post-Quantenkryptografie (Post-Quantum Cryptography)
TLS	Transport Layer Security
NIST	National Institute of Standards and Technology
KEM	Key Encapsulation Mechanism
OQS	Open Quantum Safe

1 Einleitung

In der heutigen digitalen Ära, in der Datenschutz und -sicherheit eine zentrale Rolle spielen, wird die Grundlage unserer Datensicherheit durch kryptografische Verfahren gebildet. Dies reicht von einfacheren, wenig aufwändigeren Anwendungen wie dem Online-Banking bis hin zu komplexen Sicherheitssystemen auf nationaler Ebene. Für die kommenden Jahre wird vorausgesagt, dass ein Großteil der aktuellen Sicherheitsprotokolle durch die Entwicklung von Quantencomputern in Gefahr geraten kann.

Die noch in ihrer Entwicklungsphase befindlichen Quantencomputer versprechen Rechenfähigkeiten, die weit über die der aktuellen klassischen Computer hinausgehen. Ihre potenzielle Fähigkeit, in kurzer Zeit als sicher geltende kryptografische Schlüssel zu brechen, wird als signifikante Herausforderung für bestehende Sicherheitssysteme angesehen. In diesem sich verändernden technologischen Umfeld wird die Frage aufgeworfen, wie Datenschutz und -sicherheit auf eine Post-Quantenwelt vorbereitet werden können.

Hierzu wird angenommen, dass die Post-Quantenkryptografie (PQC) eine Lösung in dieser neuen Ära darstellt, da hiermit ausreichend robuste Algorithmen entwickelt werden, um den Angriffen von Quantencomputern zu widerstehen. Obwohl intensiv in diesem Bereich geforscht wird, bleibt die Integration von PQC in bestehende Systeme, insbesondere in das weit verbreitete Transport Layer Security (TLS) Protokoll, eine Herausforderung.

1.1 Ziel der Arbeit

Das Hauptziel dieser Bachelorarbeit besteht darin, die Kombination von klassischen kryptografischen Algorithmen und PQC für den Schlüsselaustausch zu untersuchen und einer ersten Bewertung zu unterziehen. Innerhalb der wissenschaftlichen Diskussion über diesen Ansatz existieren diverse Standpunkte, die sowohl die Sicherheit als auch die Leistungsfähigkeit

betreffen. Solch ein hybrider Ansatz könnte die Robustheit gegenüber quantenbasierten Bedrohungen beibehalten, während gleichzeitig die Geschwindigkeit und Effizienz, die klassische Algorithmen auszeichnen, genutzt wird.

Um diese Hypothese zu testen, wird diese Arbeit versuchen mehrere wichtige Fragen zu beantworten:

- Wie beeinflussen kombinierte klassische und PQC-Lösungen die Effizienz und Sicherheit des Schlüsselaustausches im Vergleich zu reinen PQC-Lösungen?
- Welche Vorteile bietet eine solche hybride Lösung in realen Anwendungsszenarien, und sind diese Vorteile signifikant genug, um ihre Implementierung zu rechtfertigen?

Letztendlich ist es Ziel dieser Arbeit, nicht nur eine theoretische Bewertung vorzunehmen, sondern auch praktische Erkenntnisse zu beschreiben, die die Anwendungsfähigkeit und Relevanz dieser kombinierten Ansätze in realen Systemen bestimmen. Es wird erwartet, dass die Ergebnisse dieser Arbeit einen wertvollen Beitrag zur aktuellen Debatte über die Rolle und Implementierung von PQC im Zeitalter der Quantencomputer leisten und Wege aufzeigen, wie die kryptografische Gemeinschaft sich am besten auf die Zukunft vorbereiten kann.

1.2 Abgrenzung

Während diese Arbeit eine umfassende Analyse und Untersuchung der PQC in der Open Quantum Safe-Version (OQS) von OpenSSL anstrebt, gibt es hierfür bestimmte Grenzen. Erstens wird die Arbeit nicht alle PQC-Algorithmen im Detail behandeln, sondern sich auf die von NIST für PQC ausgewählten Algorithmen konzentrieren. Zweitens werden die Tests und Analysen unter spezifischen Bedingungen und in einer kontrollierten Umgebung durchgeführt, was bedeutet, dass die Ergebnisse in anderen Kontexten variieren können.

Die Implementation von OQS verwendet ausschließlich TLS 1.3 und daher wird in diese Arbeit auch nur TLS 1.3 analysieren. In der aktuellen kryptografischen Forschung spielt die Authentisierung zwar eine große Rolle, doch der Schwerpunkt dieser Arbeit liegt insbesondere auf der Evaluierung der Performance von KEM-Algorithmen (Key Encapsulation Mechanisms). Für eine detaillierte Untersuchung der Signaturalgorithmen wird auf (Dimitrios Sikeridis, 2020) verwiesen. In der vorliegenden Arbeit werden zwar Signaturalgorithmen erfasst, jedoch nicht

weiter analysiert. Dennoch könnten die Ergebnisse der Analyse für nachfolgende wissenschaftliche Untersuchungen von Relevanz sein.

Ferner existieren innovative Ansätze für das Transport Layer Security (TLS) Protokoll, das ohne Signaturen auskommt, beispielsweise (Peter Schwabe, 2020). Obwohl diese Ansätze nicht im NIST-Standardisierungsprozess berücksichtigt wurden, könnten sie in zukünftigen Implementierungen von Relevanz sein.

Darüber hinaus wird diese Arbeit keine direkten Empfehlungen für die Anwendung der Post-Quantenkryptografie in bestehenden Systemen geben, sondern eher darauf abzielen, ein Verständnis für ihre generelle Effizienz und potenzielle Anwendbarkeit zu vermitteln

1.3 Zielgruppe

Diese Arbeit richtet sich an Sicherheitsforscher, Kryptografen, IT-Profis und Entscheidungsträger in Unternehmen und Organisationen, die sich für die Implementierung und das Verständnis von Post-Quantenkryptografie interessieren. Sie kann auch für Studierende in den Bereichen Informatik, Kryptografie und IT-Sicherheit von Nutzen sein, die eine fundierte Kenntnis über die aktuellen Entwicklungen im Bereich der Post-Quantenkryptografie suchen.

1.4 Struktur der Arbeit

Zunächst werden in den Grundlagen der Post-Quantenkryptografie theoretische Konzepte behandelt, einschließlich der Bedrohungen durch Quantencomputerangriffe und einem Überblick über den NIST-Standardisierungsprozess. (Grundlagen Post-Quantenkryptografie)

Eine detaillierte Analyse der von NIST für die Post-Quantenkryptografie ausgewählten Algorithmen wird im dritten Kapitel (Analyse der von NIST ausgewählten Post-Quantenalgorithmen) vorgenommen. Hierbei werden verschiedene Arten von Algorithmen, wie gitter- und codebasierte, untersucht. Das vierte Kapitel (TLS und die Integration von PQC) konzentriert sich auf die Integration von Post-Quantenkryptografie in das Transport Layer Security-Protokoll, wobei verschiedene Konzepte, Ansätze und die Bedeutung von hybrider Kryptografie diskutiert werden.

Die praktische Untersuchung der Open Quantum Safe Library wird im fünften Kapitel (Praktische Untersuchung der Open Quantum Safe Library) behandelt. Es umfasst die Architektur und das Design der Bibliothek, die Testumgebung, die Verfahren sowie eine Auswertung der Ergebnisse. Im sechsten Kapitel (Evaluation der Ergebnisse) werden diese Ergebnisse evaluiert, wobei der Schwerpunkt auf dem Speicherverbrauch, lokalen Tests und den Ergebnissen des OQS Testservers liegt.

Abschließend wird im letzten Kapitel (Fazit und Ausblick) ein Fazit gezogen und ein Ausblick auf zukünftige Entwicklungen im Bereich der Post-Quantenkryptografie gegeben. Ergänzt wird die Arbeit durch ein Literaturverzeichnis und Anhänge, die zusätzliche unterstützende Informationen und Materialien bereitstellen.

2 Grundlagen Post-Quantenkryptografie

Während herkömmliche kryptografische Algorithmen, wie z.B. RSA (Rivest–Shamir–Adleman), ECC (Elliptic Curve Cryptography) oder AES (Advanced Encryption Standard) auf Problemen basieren, die für klassische Computer schwer zu lösen sind, könnten diese von Quantencomputern in polynomialer Zeit gebrochen werden. [1] Die Grundidee der Post-Quantenkryptografie (PQC) ist es, sich auf mathematische Probleme zu stützen, die selbst für Quantencomputer schwer zu lösen sind.

In der Quantenwelt gibt es verschiedene Konzepte, die Angriffe auf klassische kryptografische Algorithmen ermöglichen. Ein solches Konzept ist der Quantenparallelismus, das einem Quantencomputer ermöglicht, viele verschiedene Berechnungspfade gleichzeitig auszuführen. Es ist eines der Schlüsselelemente, das Quantencomputer potenziell leistungsfähiger als klassische Computer in bestimmten Aufgaben macht.

Ein Quantencomputer arbeitet mit Qubits statt mit den üblichen Bits eines klassischen Computers. Ein Qubit kann sich nicht nur in einem Zustand von 0 oder 1 befinden, sondern auch in einer Überlagerung dieser Zustände, die Superposition. Dies bedeutet, dass es gleichzeitig Informationen über sowohl den Zustand 0 als auch den Zustand 1 tragen kann.[2]

[1] (Vasileios Mavroeidis, 2018)

[2] (Vlatko Vedral, 1998)

Wenn nun mehrere Qubits in Überlagerungszuständen verwendet werden, kann ein Quantencomputer in der Theorie eine exponentiell wachsende Anzahl von Berechnungen gleichzeitig durchführen. Zum Beispiel sind zwei Qubits äquivalent zu vier klassischen Bits, drei Qubits zu acht klassischen Bits und bereits 10 Qubits können die selben Berechnungen wie 1024 klassische Bits durchführen.[3] [4]

2.1 Bedrohung durch Angriffe mit Quantencomputer

Die Faktorisierung großer Zahlen spielt in vielen kryptografischen Systemen eine zentrale Rolle. Insbesondere wird das RSA-Verschlüsselungssystem, eines der am häufigsten verwendeten Public-Key-Systeme, durch die Schwierigkeit der Faktorisierung großer Zahlen abgesichert. Das Sicherheitsversprechen von RSA beruht darauf, dass es mit klassischen Algorithmen extrem zeitaufwendig ist, das Produkt zweier großer Primzahlen zu zerlegen, während es relativ einfach ist, diese Zahlen zu multiplizieren. Im RSA-Verfahren werden zwei große Primzahlen p und q heimlich gewählt und miteinander multipliziert, um $n = p * q$ zu erhalten. Das Produkt n wird Teil des öffentlichen Schlüssels. Der private Schlüssel hingegen ist unter anderem von p und q abhängig.

Die Sicherheit von RSA beruht darauf, dass es, wenn jemand eine mit RSA verschlüsselte Nachricht entschlüsseln möchte, ohne den privaten Schlüssel zu kennen, erforderlich ist, n in seine Faktoren p und q zu zerlegen. Mit aktuellen Algorithmen und herkömmlichen Computern ist dies extrem schwierig und zeitaufwendig, wenn n groß genug ist (z.B. mehrere hundert oder tausend Bit lang).[5]

Das Problem der Faktorisierung großer Zahlen wird als "schwer" betrachtet, weil alle bekannten klassischen Algorithmen exponentielle Zeit benötigen, um eine Zahl zu faktorisieren, die

[3] (Yoder, 2021)

[4] (Hartnett, 2019)

[5] (Zhengping Jay Luo, 2023)

das Produkt von zwei gleich großen Primzahlen ist. Dies hat zur bisher akzeptierten Annahme geführt, dass RSA, solange die Schlüssellängen groß genug sind, gegen Angriffe sicher ist.[6]

Das diskrete Logarithmus-Problem (DLP) ist ein weiteres fundamentales Problem, das in der Kryptografie benutzt wird. Das DLP dient als Grundlage für viele kryptografische Lösungen, insbesondere solche, die auf elliptischen Kurven basieren (ECC). Die Schwierigkeit des DLP sichert viele kryptografische Verfahren, einschließlich des Diffie-Hellman Schlüsselaustauschs und des Digital Signature Algorithm (DSA). In diesen Protokollen basiert die Sicherheit darauf, dass es einerseits zwar relativ einfach ist, Nachrichten in eine Richtung (z.B. das Generieren eines öffentlichen Schlüssels aus einem privaten) zu senden, es aber andererseits extrem schwierig ist, den Prozess umzukehren, ohne den privaten Schlüssel zu kennen.[7]

2.1.1 Shor's Algorithmus

Die bekannteste Bedrohung durch Quantencomputer für die Kryptografie ist Shor's Algorithmus. Dieser Algorithmus kann die Faktorisierung großer Zahlen und den diskreten Logarithmus in polynomialer Zeit durchführen. Traditionelle Verschlüsselungsschemata wie RSA und ECC, die auf der Schwierigkeit dieser mathematischen Probleme basieren, können durch Shor's Algorithmus gebrochen werden. [8] Shor's Algorithmus, der für die Faktorisierung großer Zahlen verwendet wird, nutzt den Quantenparallelismus, um mögliche Faktoren in einem Bruchteil der Zeit zu überprüfen, die ein klassischer Computer benötigen würde. [9]

[6] (Brown, 2008)

[7] (Albrecht Beutelspacher, 2010)

[8] (Shor, 1995)

[9] (Craig Gidney, 2021)

2.1.2 Grover's Algorithmus

Ein weiterer bekannter Quantenalgorithmus ist Grover's Algorithmus. Dieser kann eine unsortierte Datenbank in einer in Quadratwurzel reduzierter Zeit durchsuchen ($O(\sqrt{N})$), die ein klassischer Computer benötigen würde ($O(N)$). [10] [11]

Bei der Herangehensweise zur Überwindung der Sicherheitsbarriere würde ein Angreifer, der den Schlüssel nicht kennt, möglicherweise eine Brute-Force-Suche durchführen müssen, bei der er jeden möglichen Schlüssel ausprobiert, bis er den richtigen findet. Für einen AES-Schlüssel mit einer Länge von 128 Bits gibt es 2^{128} mögliche Schlüssel. Grover's Algorithmus kann die benötigte Suchzeit von 2^{128} auf etwa 2^{64} Schritte reduzieren, was immer noch eine enorme Anzahl ist, aber deutlich weniger als die Brute-Force-Suche.

Eine direkte Folge des Grover's Algorithmus für die symmetrische Kryptografie ist die Überlegung, die Schlüssellängen zu verdoppeln, um die gleiche Sicherheitsmarge gegenüber Quantenangriffen zu erhalten, die sie gegenüber klassischen Angriffen hat. Das bedeutet, dass ein 128-Bit-Schlüssel (der gegenüber klassischen Angriffen als sicher gilt) in einer Quantenwelt auf 256 Bits erhöht werden sollte, um gegenüber Quantenangriffen sicher zu sein. [12] [13]

Es ist wichtig zu beachten, dass, obwohl Grover's Algorithmus die Sicherheit von symmetrischer Kryptografie beeinträchtigt, der Algorithmus sie aber nicht "bricht" oder nutzlos macht. Im Gegensatz dazu hat Shor's Algorithmus, die Fähigkeit, bestimmte Public-Key-Kryptosysteme zu "brechen". Dementsprechend ist es auch eher von Bedeutung einen Ersatz für asymmetrische Kryptografie zu finden.

[10] (Jang, 2021)

[11] (Kyungbae Jang, 2022)

[12] (Kyungbae Jang, 2022)

[13] (Marre, 2018)

2.2 Überblick über den NIST-Standardisierungsprozess

Das National Institute of Standards and Technology (NIST) ist eine US-amerikanische Behörde, die sich unter anderem mit der Entwicklung von Standards und Technologien zur Verbesserung der Datensicherheit befasst. Eines ihrer bedeutendsten Projekte in den letzten Jahren ist der Post-Quantum-Cryptography (PQC) Standardisierungsprozess. Dieser Prozess wurde ins Leben gerufen, um kryptografische Algorithmen zu identifizieren, die auch in einer Ära der Quantencomputer sicher bleiben würden. [14]

Der PQC-Standardisierungsprozess begann in 2016 mit einem öffentlichen Aufruf von NIST zur Einreichung von Vorschlägen für post-quantenkryptografische Algorithmen.[15] In der ersten Runde wurden 69 verschiedene PQC-Algorithmen eingereicht.[16] Die eingereichten Algorithmen wurden anschließend einer ersten Prüfung unterzogen. Hierbei wurden sie in Bezug auf ihre Sicherheit, Effizienz und andere relevante Faktoren analysiert.

Nach der ersten Bewertung wurden die verbliebenen Kandidaten veröffentlicht, um Feedback und Kommentare aus der breiten Öffentlichkeit, insbesondere von Experten auf dem Gebiet, zu sammeln.

Der gesamte Prozess umfasste mehrere Runden der Einreichung, Bewertung, öffentlichen Kommentierung und weiteren Verfeinerung der Algorithmen, um die besten und sichersten Kandidaten zu identifizieren. [17]

Im Jahr 2022 wurden die ersten vier Algorithmen bekanntgegeben: Drei Signaturalgorithmen – CRYSTALS-Dilithium, FALCON und SPHINCS+ – sowie der Public-Key Verschlüsselungsalgorithmus: CRYSTAL-Kyber.[18] Weiterhin ist geplant, im Jahr 2024 Standards für die

[14] (NIST, 2023)

[15] (NIST, 2016)

[16] (NIST, 2017)

[17] (NIST, 2023)

[18] (NIST, 2022)

Algorithmen BIKE, Classic McEliece, HQC und anfänglich SIKE (siehe PQC in TLS: Konzepte und Ansätze4.2) festzulegen.[19]

NIST selektiert nicht einen einzigen Gewinner, sondern präferiert die Auswahl mehrerer Kandidaten. Bei der Auswahl wurde sorgfältig darauf geachtet, eine Vielzahl von alternativen Algorithmen zu berücksichtigen und zu erlauben, um Flexibilität zu gewährleisten, falls einige Methoden zukünftig als unsicher eingestuft werden oder Schwachstellen aufweisen sollten.

[19] (NIST, 2022)

3 Analyse der von NIST ausgewählten Post-Quantenalgorithmen

Key Encapsulation Methods (KEM) sind ein wichtiger Bestandteil der Kryptografie. Sie werden insbesondere in der Public-Key-Kryptografie verwendet, um Schlüssel sicher zu übertragen. Die grundlegende Idee hinter KEM ist: Anstatt einen geheimen Schlüssel direkt zu senden, wird er "eingekapselt" oder in eine Form gebracht, in der er sicher über unsichere Kanäle übertragen werden kann.

Die Performance von KEMs ist essentiell für eine Vielzahl von modernen kryptografischen Verfahren. Dies begründet sich hauptsächlich dadurch, dass KEMs in Protokollen Anwendung finden, welche eine rasche und effiziente Verschlüsselung sowie Entschlüsselung voraussetzen. Die Selektion eines geeigneten KEMs ist von diversen Parametern abhängig, dazu zählen die konkrete Anwendungsdomäne, die angestrebte Sicherheitsstufe sowie die zur Verfügung stehenden Ressourcen.[20]

NIST hat verschiedene Sicherheitsstufen für post-quantenkryptografische Systeme definiert. Verschiedene Parameter können unterschiedliche Sicherheitsniveaus bieten. In der Praxis werden diese Niveaus oft in Begriffen wie "128-bit Sicherheit" oder "256-bit Sicherheit" ausgedrückt. Das bezieht sich auf die geschätzte Menge an Arbeit, die notwendig ist, um das System zu brechen, gemessen in Bit-Operationen. Stufe 1 bezeichnet ein Sicherheitsniveau, bei dem jegliche Angriffe eine Rechenkomplexität aufweisen müssen, die zumindest der eines Angriffs auf eine Blockchiffre mit einem 128-Bit-Schlüssel entsprechen, wie zum Beispiel bei AES-128. Analog dazu bedeutet Stufe 3 ein Sicherheitsniveau von einem 192-Bit-Schlüssel, wie

[20] (Goutam Tamvada, 2022)

AES-192. Schließlich erfordert Stufe 5 ein Sicherheitsniveau, welches der Rechenkomplexität eines Angriffs auf eine Blockchiffre mit einem 256-Bit-Schlüssel gleichkommt.[21]

In der sich entwickelnden Welt der Post-Quantenkryptografie sind verschiedene KEMs von zentraler Bedeutung, da sie als resistent gegen mögliche Angriffe durch Quantencomputer gelten. Zu den beachtenswerten Systemen gehören hash-basierte[22], gitter-basierte[23] [24], code-basierte[25] und auf multivariaten polynomiellen Gleichungen[26] basierende Ansätze.

Die von NIST ausgewählten Algorithmen, einschließlich jener, die noch im Jahr 2024 bekannt gegeben werden sollen, basieren auf Gitter- oder Code-basierten Konzepten. Infolgedessen werden diese spezifischen kryptografischen Ansätze detaillierter betrachtet.

3.1 Gitterbasierte Algorithmen

Das grundlegende Prinzip der gitterbasierten Kryptografie stützt sich auf die Schwierigkeit bestimmter Probleme, die auf Gittern basieren. Ein Gitter ist eine Menge von Punkten im Raum, die durch eine Basis von Vektoren definiert wird. Die Schwierigkeit, bestimmte Vektoren in diesem Gitter zu finden oder Abstände zwischen ihnen genau zu bestimmen, bildet das Herzstück der Sicherheit von gitterbasierten kryptografischen Systemen.

Eines der bekanntesten dieser Probleme ist das "kürzeste Vektorproblem" (SVP), das das Finden des kürzesten Vektors in einem Gitter beinhaltet oder das Learning With Errors (LWE)

[21] (NIST, 2016)

[22] (Daniel J. Bernstein, 2019)

[23] (Regev, 2006)

[24] (Daniele Micciancio, 2009)

[25] (Raphael Overbeck, 2009)

[26] (Jintai Ding, 2009)

Problem, bei dem es im Wesentlichen darum geht, aus einer gegebenen Menge von linearen Gleichungen mit verrauschten Lösungen die ursprüngliche Lösung zu erraten. [27]

Ein wichtiges gemeinsames Merkmal beider Probleme ist ihre sogenannte "Worst-Case-Hardness". Das bedeutet, dass selbst das schwierigste Instanzproblem immer noch die Sicherheit des gesamten Systems gewährleistet. Das steht im Gegensatz zu vielen traditionellen kryptografischen Systemen, bei denen die Sicherheit oft auf Durchschnittswerten basiert. [28]

Zu den gitterbasierten KEM-Algorithmen gehört der bereits von NIST als Standard ausgewählte Kyber und die Signatur-Algorithmen Dilithium und Falcon. Kyber nutzt das MLWE (Module Learning with Errors) Problem, das Vektoren von Polynomen für die Berechnung der Schlüssel verwendet. Ein bemerkenswertes Merkmal von Kyber ist dabei seine Effizienz sowohl in Bezug auf die Rechenzeit als auch auf den Speicherbedarf. Deshalb wird es besonders attraktiv für Anwendungen, bei denen Ressourcenbeschränkungen eine Rolle spielen. [29]

Algorithmus	*Secret Key*	*Public Key*	*Output*	*Level*
Kyber-512	1632	800	768	1
Kyber-768	2400	1184	1088	3
Kyber-1024	3168	1568	1568	5

Tabelle 1:Die Größe des privaten Schlüssels, des öffentlichen Schlüssels, des verschlüsselten Outputs jeweils in Bytes und die behauptete Sicherheitsstufe in Bezug auf symmetrischen Algorithmen[30]

[27] (Daniele Micciancio, 2009)

[28] (Ajtai, 1996)

[29] (Roberto Avanzi, 2021)

[30] (Roberto Avanzi, 2021)

3.2 Codebasierte Algorithmen

Die codebasierte Kryptografie wurde erstmals in den 1970er Jahren eingeführt. Das bekannteste System in dieser Kategorie ist das McEliece-Kryptosystem, das 1978 von Robert McEliece vorgestellt wurde. Das grundlegende Prinzip besteht darin, die Schwierigkeit der Decodierung zufälliger linearen Codes als Basis für kryptografische Sicherheit zu verwenden.

Die Idee ist, dass es zwar einfach ist, eine Nachricht unter Verwendung eines bekannten Codierungsschemas zu kodieren (d.h. in ein Codewort zu übersetzen), aber extrem schwierig, sie ohne das entsprechende Wissen zu dekodieren. [31]

Zunächst wird ein zufälliger linearer Code (z.B. Goppa-Code oder MDPC-Code) gewählt, welcher als geheimer Schlüssel dient. Die dazugehörige Generatormatrix dieses Codes wird dann mit Hilfe zufällig gewählter invertierbarer Matrizen in eine "verfälschte" Form gebracht. Diese modifizierte Matrix wird als Teil des öffentlichen Schlüssels verwendet.

Der private Schlüssel besteht aus der ursprünglichen Generatormatrix des zufällig gewählten linearen Codes und den verwendeten zufälligen invertierbaren Matrizen, die zur Verfälschung eingesetzt wurden. Mit Kenntnis dieses Schlüssels kann eine Nachricht effizient dekodiert werden, indem man den Code nutzt und die Verfälschung mit den zufälligen Matrizen rückgängig macht.

Der öffentliche Schlüssel besteht aus der "verfälschten" Generatormatrix. Mit diesem Schlüssel können Nachrichten verschlüsselt werden. Ohne Kenntnis des privaten Schlüssels ist es jedoch sehr schwierig, die verschlüsselte Nachricht zu dekodieren. [32]

[31] (Raphael Overbeck, 2009)

[32] (Sendrier, 2017)

Die Algorithmen BIKE[33], HQC[34] und Classic McEliece, die sich noch im NIST-Standardisierungsprozess befinden, verwenden alle codebasierte Systeme.

Algorithmus	*Secret Key*	*Public Key*	*Output*	*Level*
BIKE-L1	5223	1541	1573	1
BIKE-L3	10105	3083	3115	3
BIKE-L5	16494	5122	5154	5
HQC-128	2289	2249	4481	1
HQC-192	4562	4522	9026	3
HQC-256	7285	7245	14469	5
C.M.-348864	6492	261120	96	1
C.M.-460896	13608	524160	156	3
C.M.-6688128	13932	1044992	208	5

Tabelle 2: Die Größe des privaten Schlüssels, des öffentlichen Schlüssels, des verschlüsselten Outputs jeweils in Bytes und die behauptete Sicherheitsstufe[35] [36]

[33] (Nicolas Aragon, 2022)

[34] (Carlos Aguilar Melchor, 2023)

[35] (Nicolas Aragon, 2022)

[36] (Carlos Aguilar Melchor, 2023)

4 TLS und die Integration von PQC

Das Transport Layer Security-Protokoll ist der Standard für sichere Kommunikation im Internet. Das Protokoll stützt sich auf eine Vielzahl verschiedener kryptografischer Verschlüsselungsmethoden. Folglich ist es in besonderem Maße den potenziellen Risiken ausgesetzt, die durch die Entwicklung von Quantencomputern entstehen.

4.1 Grundlagen des Transport Layer Security-Protokolls

Es schützt Daten, die zwischen zwei Systemen übertragen werden, vor Abhören und Manipulation. Dies wird durch den Einsatz von Verschlüsselungsalgorithmen erreicht. Die grundlegenden Konzepte und Komponenten von TLS sind das Handshake-Protokoll, die Verschlüsselung, Authentifizierung und Integrität. Die Sammlung der verwendeten Algorithmen in den Komponenten wird als „Cipher Suite“ bezeichnet.[37]

Bevor die Daten sicher übertragen werden können, müssen die beiden Kommunikationspartner (z.B. ein Webserver und Webbrowser) eine sichere Verbindung aufbauen. Dieser Prozess wird als "Handshake" bezeichnet. Während des Handshakes tauschen die beiden Parteien Zertifikate aus, um ihre Identität zu bestätigen, und sie vereinbaren, welche kryptografischen Algorithmen (z.B. für Verschlüsselung und Authentifizierung) verwendet werden sollen. Der Handshake verwendet eine Kombination aus asymmetrischer und symmetrischer Kryptografie.

Nachdem der Handshake abgeschlossen ist, werden die Daten verschlüsselt übertragen. Dies stellt sicher, dass Dritte, die möglicherweise den Datenverkehr abfangen, die Daten nicht lesen oder verändern können. TLS verwendet symmetrische Verschlüsselung für die Datenübertragung, wobei beide Parteien denselben Schlüssel verwenden.

---

[37] (T. Dierks, 2008)

Für die Authentifizierung verwendet TLS digitale Zertifikate, um die Identität der Kommunikationspartner zu bestätigen. Diese Zertifikate werden von vertrauenswürdigen Dritten, sogenannten Zertifizierungsstellen (Certificate Authorities, CAs), ausgestellt. Wenn beispielsweise ein Webbrowser eine Verbindung zu einem Webserver herstellt, präsentiert der Server sein Zertifikat, um zu beweisen, dass er tatsächlich der ist, für den er sich ausgibt.

Mit der Verschlüsselung stellt TLS auch sicher, dass die übertragenen Daten nicht verändert wurden. Dies wird durch den Einsatz von Message Authentication Codes (MACs) erreicht. Ein MAC ist ein kurzer Wert, der aus der Nachricht und einem geheimen Schlüssel berechnet wird. Der Empfänger kann den MAC überprüfen, um sicherzustellen, dass die Nachricht während der Übertragung nicht verändert wurde.[38]

Daraus ergeben sich verschiedene Angriffspunkte für einen Quantencomputer. Wenn der Server ein Zertifikat mit einem RSA- oder ECC-basierten öffentlichen Schlüssel sendet, könnte ein Angreifer mit einem Quantencomputer den zugehörigen privaten Schlüssel berechnen. Mit dem privaten Schlüssel könnte der Angreifer dann alle Nachrichten entschlüsseln, die mit dem öffentlichen Schlüssel codiert wurden.[39]

Die größte Gefahr besteht in der rückwirkenden Entschlüsselung. Selbst wenn der Angreifer während des Handshakes passiv bleibt und nur den Datenverkehr aufzeichnet, könnte er, mit Zugriff auf einen leistungsstarken Quantencomputer, den gesamten aufgezeichneten Datenverkehr rückwirkend entschlüsseln.[40] Dies ist besonders besorgniserregend für Daten, die über lange Zeiträume hinweg vertraulich bleiben müssen. In der Konsequenz bedeutet dies, dass viele Systeme bereits jetzt schon von solchen Angriffen betroffen sein könnten, denn wenn in der Kommunikation keine quantenresistenten Algorithmen verwendet worden sind, ist die sogenannte „Forward Secrecy“ auch in Gefahr. Das Problem im Kontext von Quantencomputern besteht darin, dass der initiale Handshake (bei dem der Sitzungsschlüssel ausgetauscht wird)

[38] (Rescorla, 2018)

[39] (Panos Kampanakis, 2018)

[40] (Mohamed Taoufiq Damir, 2022)

aufgezeichnet und später durch einen Quantencomputer gebrochen wird und damit Forward Secrecy keinen Schutz mehr bietet.[41]

4.2 PQC in TLS: Konzepte und Ansätze

Ein Ansatz zur Integration der PQC in das TLS-Protokoll besteht darin, sich ausschließlich auf PQC-Algorithmen zu beschränken. Dies würde bedeuten, dass in TLS Komponente, die keine quantenresistenten Algorithmen beinhalten, ausschließlich PQC-Algorithmen verwendet werden. Dies klingt theoretisch vielversprechend, sollte allerdings derzeit in der Praxis noch nicht so umgesetzt werden. So wurden mehr als 15 der bei NIST eingereichten PQC-Algorithmen bereits gebrochen oder bieten weniger Sicherheit, als zunächst behauptet wurde.[42] [43] [44] [45] Die PQC-Algorithmen befinden sich noch in einem frühen Entwicklungsstadium und sind weitgehend unerforscht. Daher könnte ein zu früher Umstieg auf ausschließlich PQC-Algorithmen schwerwiegende Konsequenzen haben. Ein bemerkenswertes Beispiel hierfür ist der SIKE-Algorithmus, der es zwar in die finale Runde des NIST-Standardisierungsprozesses geschafft hat, jedoch kurz nach seiner Veröffentlichung als "gebrochen" eingestuft wurde.[46] Dieser Angriff konnte sogar mithilfe eines herkömmlichen Computers innerhalb einer Stunde durchgeführt werden. [47]

Im Gegensatz dazu wurde z.B. die elliptische Kurvenkryptografie (ECC) im Jahr 1985 entwickelt und im Jahr 2006 im RFC von TLS eingeführt, wobei sie im Laufe der Jahre kontinuierlich verbessert wurde.[48] Dies unterstreicht zwei wichtige Erkenntnisse. Erstens ist es von entscheidender Bedeutung, den Übergang zur Post-Quantenkryptografie frühzeitig zu planen.

---

[41] (Surbhi Shaw, 2022)

[42] (Steinfeld, 2018)

[43] (Haoyu Li, 2021)

[44] (Lyubashevsky, 2019)

[45] (Beullens, 2022)

[46] (NIST, 2022)

[47] (Wouter Castryck, 2022)

[48] (Bodo Moeller, 2006)

Zweitens sollten Lösungen, die ausschließlich auf PQC-Algorithmen setzen, derzeit nicht eingesetzt werden. Bei dem hybriden Ansatz (nicht zu verwechseln mit der Kombination von symmetrische und asymmetrische Verschlüsselungstechniken) werden klassische und PQC-Algorithmen kombiniert.

4.3 Bedeutung und Anwendung von hybrider Kryptografie

Der Hauptvorteil des hybriden Ansatzes besteht darin, dass sie sowohl gegen klassische als auch gegen potenzielle zukünftige quantenkryptografische Angriffe resistent sind. Selbst wenn der post-quantenkryptografische Algorithmus in der Zukunft kompromittiert wird, bietet der klassische Algorithmus immer noch einen gewissen Schutz. Das Hauptziel eines hybriden Modus besteht darin, sicherzustellen, dass die gewünschte Sicherheitseigenschaft solange erhalten bleibt, bis mindestens eins der verwendeten Algorithmen noch sicher ist.[49]

Die Integration von quantensichere Algorithmen in TLS ist eine Herausforderung, da sie oft andere Anforderungen und Eigenschaften haben als klassische Algorithmen. Dafür muss das TLS-Protokoll geändert werden, um anzugeben, wie eine Kombination von Algorithmen ausgehandelt wird, wie diese kryptografisch kombiniert werden können, um hybride Modi des Schlüsselaustauschs oder der Authentifizierung zu unterstützen, wie die Backwads Compatibility gelöst wird und etc. [50] [51]

Die hybride Verschlüsselung ermöglicht eine schrittweise Integration von PQC-Verfahren. Unternehmen können schrittweise migrieren und so hierfür entstehende Kosten und Komplexität reduzieren. Einer der ersten Versuche mit CECPQ1 war von Google im Jahr 2016. CECPQ1 kombinierte den traditionellen elliptischen Kurven-Schlüsselaustausch (ECC) mit einem post-quantenkryptografischen Schlüsselaustausch (NewHope). Nach CECPQ1 hat Google weitere hybride Schlüsselaustauschexperimente durchgeführt, darunter CECPQ2, das andere post-

[49] (Diana Ghinea, 2022)

[50] (D. Stebila, 2023)

[51] (Mike Ounsworth, 2023)

quantenkryptografische Algorithmen verwendet (HRSS) und eine Alternative mit dem bereits erwähnten SIKE, der nicht mehr als sicher gilt. [52]

Festzuhalten ist, dass, CECPQ2 und die folgenden Experimente das Bewusstsein für die Bedeutung hybrider Kryptografie erhöht haben. Andere große Technologieunternehmen und Organisationen begannen, durch Untersuchungen mit ähnlichen Ansätzen sicherzustellen, dass ihre Systeme und Kommunikationskanäle auch in einer post-quanten Welt sicher sind. [53] [54]

Neben dem Schlüsselaustausch gibt es auch hybride Ansätze in anderen Bereichen der Kryptografie. Dies ist bei digitalen Signaturen zu beobachten.[55] Einige Algorithmen kombinieren klassische auf Public-Key-Infrastrukturen basierende digitale Signaturen mit Post-Quantensignaturen, um die Integrität und Authentizität von Nachrichten in einer post-quanten Umgebung zu gewährleisten. Jedoch haben PQC-Algorithmen oft größere Schlüsselgrößen als klassische Algorithmen mit der möglichen Folge einer Erhöhung des Bandbreitenverbrauchs. Zudem können PQC-Algorithmen langsamer sein und damit die Gesamtleistung des Systems beeinträchtigen. [56]

4.4 Position der NSA zu hybriden Algorithmen

Obwohl die NSA (National Security Agency) und andere behördennahe Organisationen von hybriden Verfahren abraten, sind diese derzeit die einzigen Methoden, die als sicher und rechtzeitig einsatzbereit gelten. [57]

[52] (Daniel J. Bernstein, 2021)

[53] (Mila Anastasova, 2022)

[54] (Bas Westerbaan, 2022)

[55] (Diana Ghinea, 2022)

[56] (Yu Li, 2023)

[57] (Alison Becker, 2021)

Laut verschiedenen veröffentlichten Emails hat sich die NSA für reine PQC-Lösungen entschieden.[58] Obwohl einige Benutzer bei der Anwendung hybrider Lösungen Sicherheitsbedenken haben könnten, sieht die NSA keinen Unterschied in der Sicherheit von hybriden und nicht-hybriden Lösungen. Es wird darauf hingewiesen, dass Implementierungsfehler in aktuellen kryptografischen Algorithmen häufig sind und aus dem Verdoppeln von Algorithmen ein größerer Fehlerbereich resultiert.

Diese Hinweise und Argumentationen sind in weitreichenden Gegenstimmen und Diskussionen in der Wissenschaft aufgegriffen.[59] Es gibt Bedenken bezüglich eines Vorstoßes der NSA, einerseits Gitter-Systeme einzusetzen, um gegen Quantencomputer zu schützen, und andererseits gleichzeitig ECC abzuschalten.

Stattdessen wird vorgeschlagen ECC in alle post-quanten Einsätze zu integrieren. Die Kritik am NIST-PQC-Prozess besteht insbesondere in der fehlenden Transparenz und den überraschenden kryptografischen Angriffen auf die von NIST ausgewählten Verfahren. Die NSA wird so aufgefordert, Details ihres Zugangs zu diesem Prozess offenzulegen. Dabei wird betont, dass Crypto Forum Research Group (CFRG), Internet Research Task Force (IRTF) und Internet Engineering Task Force (IETF) nicht verpflichtet sind, den Wünschen der NSA zu folgen. Sicherheit sollte das oberste Ziel sein. Es gibt weitere Vorschläge, dass die hybride Option für mindestens empfindliche Anwendungen in Betracht gezogen werden, bei denen die Kosten den Nutzen rechtfertigen und für Daten von geringem Wert und kurzer Dauer und einem geringen Budget sich eine einzelne nicht-hybride PQC-Algorithmus-Option lohnen könnte.[60]

[58] (Becker, 2021)

[59] (Bernstein, 2021)

[60] (Brown, 2021)

4.5 Bewertung der Eignung der Algorithmen für TLS

TLS erfordert schnelle und effiziente Algorithmen. Im Gegensatz zu BIKE, HQC oder Classic McEliece zeichnet sich Kyber durch relativ kleine Schlüsselgrößen aus. Dies erleichtert die Implementierung in Umgebungen mit begrenztem Speicher. Dadurch eignet sich Kyber als ein guter Kandidat für die Verwendung hybrider Methoden in TLS. Zudem ist Kyber bereits als Standard ausgewählt und die restlichen drei Algorithmen befinden sich noch im Auswahlverfahren.

In der Praxis wird die folgende hybride Methode vorgeschlagen, in der Nachrichten der hybridisierten Algorithmen verkettet und als ein einziger Wert übertragen werden. Es werden beide KEMs als Eingaben für eine Schlüsselableitungsfunktion (Key Derivation Function, KDF) verwendet, die die Verletzung beider Algorithmen erfordert, um auf den Klartext zuzugreifen. Die Konkatenation der Schlüssel wurde von NIST für die Schlüsselableitungsfunktion empfohlen.[61] Auf diese Weise wird die Sicherheit des Protokolls zu einem "Alles-oder-Nichts"-Protokoll, d. h. ein Angreifer müsste beide Algorithmen überwinden, um an den Klartext zu gelangen. [62] [63] Die Konkatenation (∘) von PQC mit einem klassischen Algorithmus würde wie folgt aussehen:

```
Hybrider Schlüssel = klassischer Schlüssel ∘ PQC Schlüssel
```

Dadurch erhöht sich zwar der Speicherbedarf der Algorithmen, aber der Mehrbedarf wird durch die zusätzliche Sicherheitsebene mit der hybriden Methode ausgeglichen (siehe Tabelle 3: Die Größenunterschiede der Kombinationen von PQC und klassischen Algorithmen.). Kyber, mit seiner vergleichsweise kleinen Schlüsselgröße, macht es nicht nur möglich, eine solche hybride

[61] (Elaine Barker, 2020)

[62] (Mike Ounsworth, 2023)

[63] (D. Stebila, 2023)

Methode effektiv umzusetzen, sondern auch in solchen Anwendungen zu integrieren, in denen der Speicherplatz ein knappes Gut ist, wie zum Beispiel in IoT-Geräten oder eingebetteten Systemen.[64]

Algorithmus	*Secret Key*	*Public Key*	*Output*
X25519_Kyber512	1664	832	800
X448_Kyber768	2465	1240	1144
P521_Kyber1024	3234	1701	1701
P256_HKDF_SHA256	32	65	65
P384_HKDF_SHA384	48	97	97
P521_HKDF_SHA512	66	133	133
X25519_HKDF_SHA256	32	32	32
X448_HKDF_SHA512	56	56	56

Tabelle 3: Die Größenunterschiede der Kombinationen von PQC und klassischen Algorithmen. [65]

Zudem bietet dieser Ansatz zukunftssichere Flexibilität. Sollte in der Zukunft einer der Algorithmen – sei es der klassische oder der PQC – als unsicher oder angreifbar eingestuft werden, könnte er einfach durch einen anderen, sichereren Algorithmus ersetzt werden, ohne das gesamte System zu überholen.

[64] (Adarsh Kumar, 2022)

[65] (Nakov, 2021)

5 Praktische Untersuchung der Open Quantum Safe Library

5.1 Architektur und Design der Open Quantum Safe Library

Das Open Quantum Safe (OQS) ist ein gemeinsames Projekt vieler Unternehmen und Universitäten, um ein Library für PQC-Lösungen zu finden. Es kooperiert eng mit NIST. Diese Bibliothek zielt darauf ab, Entwicklern die Möglichkeit zu geben, quantensichere Kryptografie in ihre Anwendungen zu integrieren, um sich gegen zukünftige Bedrohungen durch Quantencomputer zu schützen. [66] Die Open Quantum Safe Library ist modular aufgebaut und ermöglicht es Entwicklern, verschiedene quantensichere Algorithmen zu verwenden, je nach ihren spezifischen Anforderungen. Die Hauptkomponente der Bibliothek ist liboqs.[67] Dies ist der Hauptteil der Bibliothek und enthält die Implementierung der quantensicheren Algorithmen. Die liboqs ist in C geschrieben und bietet eine Schnittstelle, über die Anwendungen auf die Algorithmen zugreifen können. Die Bibliothek bietet unter anderem auch Algorithmen an, die bei NIST nicht ausgewählt worden sind. Zudem gibt es noch verschiedene Wrapper-Projekte, die es ermöglichen OQS in verschiedenen Programmiersprachen zu verwenden.

Darüber hinaus stellt OQS einen Testserver für die Algorithmen bereit. Er dient als Plattform für Kunden, um die Interoperabilität mit PQC-Algorithmen verbesserter Software und verschiedenen PQC-Algorithmen über verschiedene Ports zu testen. Die Website bietet auch detaillierte Informationen über die Spezifikationen, Vorbehalte, Testverfahren und eine Liste der

[66] (OQS, 2017)

[67] (liboqs, 2017)

unterstützten PQC-Signatur-/Schlüsselaustauschalgorithmen.[68] Es gibt eine breite Palette an, von OQS angepasste, Software und Protokolle wie z.B. QUIC, Wireshark, nginx, Chromium etc., die ebenfalls PQC Algorithmen unterstützen.

5.2 Testumgebung

Das verwendete System für die Durchführung der Test ist eine virtuelle Maschine mit Ubuntu 22.04.3 LTS als Betriebssystem, 12 GB RAM und ein Intel(R) Core(TM) i5-8250U. Die verwendeten Software und Librarys sind: OpenSSL 3.0.2, liboqs 0.8.0, oqsprovider 0.5.1, liboqs-python 0.8.0, Python 3.10.12, Docker 24.0.6, sowie die Docker Images openquantumsafe/curl und openquantumsafe/wireshark.

5.3 Verfahren

Die Untersuchung der Leistungsfähigkeit erfolgte mittels zweier zentraler Methoden. Erstens, die lokale Methodik, welche die Quantität der Handshake-Verbindungen innerhalb eines definierten Zeitrahmens erfasst. Zweitens, die Methode der OQS-Testserver, die sich auf die Messung der Dauer von Handshake-Vorgängen einzelner Verbindungen zu einem von OQS bereitgestellten Server konzentriert.

5.3.1 Lokale TLS-Handshake Performance

Zum Testen der TLS-Handshake Zeit wurde eine von der OQS-cURL Version bereitgestellte Messmethode, in Form des nachfolgenden Skripts, verwendet. Hierbei werden die kryptografischen Algorithmen des Open Quantum Safe (OQS) Projekts genutzt, um die Praktikabilität und Effizienz von PQC-Kryptografie in realen Kommunikationsszenarien zu testen.

[68] (OQS, 2023)

Zu Beginn wird eine kontrollierte Initialisierung der Algorithmen für den Schlüsselaustausch und die Signatur durchgeführt. Wenn ein Signaturalgorithmus (SIG_ALG) festgelegt ist, wird ein neuer privater Schlüssel und ein entsprechendes Zertifikats-Signatur-Anforderungsdatei (CSR) generiert (siehe Zeile 1). Diese CSR wird dann verwendet, um ein Serverzertifikat von einer vordefinierten Zertifizierungsstelle (CA) zu signieren (Zeile 2).

Ein Server wird mit dem neu generierten Zertifikat und privaten Schlüssel gestartet und ist so konfiguriert, dass er nur den spezifizierten KEM_ALG akzeptiert (Zeile 3). Es wird nur TLS1.3 unterstützt. Danach führt der Server eine Anzahl von TLS-Handshakes mit `openssl s_time` durch, basierend auf TEST_TIME. `openssl s_time` ist ein Tool zum Messen der SSL-Performance. Es wird in Verbindung mit einem laufenden `openssl s_server` verwendet, um Handshakes mit einem Server zu initiieren und die Zeit zu messen, die benötigt wird, um diese Handshakes abzuschließen (Zeile 4).

```
1    openssl req -new -newkey $SIG_ALG -keyout /opt/test/server.key -out
/opt/test/server.csr -nodes -subj "/CN=localhost"
2    openssl x509 -req -in /opt/test/server.csr -out /opt/test/server.crt
-CA CA.crt -CAkey CA.key -CAcreateserial -days 365
3    openssl s_server -cert /opt/test/server.crt -key /opt/test/server.key
-groups $DEFAULT_GROUPS -www -tls1_3 -accept localhost:4433&
4    openssl s_time -connect :4433 -new -time $TEST_TIME -verify 1 | grep
connections
```

Listing 1:Shellskript der wichtigsten Befehle für die lokalen Tests

Diese Methoden bietet eine einfache Möglichkeit, Handshakes schnell zu testen, ohne eine komplizierte Testinfrastruktur aufzubauen und durch die Verwendung von `s_time` erhalten Benutzer sofortiges Feedback zur Anzahl der erfolgreich abgeschlossenen Handshakes innerhalb des festgelegten Testzeitraums.

Allerdings ist diese Methode so eingerichtet, dass sie Tests lokal auf demselben System durchführt. Hier besteht das methodische Risiko, dass Netzwerklatenzen nicht berücksichtigt

werden, obwohl sie in realen Anwendungen von Bedeutung sind. Zudem werden andere Aspekte wie die Interaktion mit realen Client-Anwendungen nicht berücksichtigt. Um statistisch signifikante Ergebnisse zu erhalten, müssen mehrere Durchläufe durchgeführt werden, um durchschnittliche Werte und Standardabweichungen zu erhalten.

5.3.2 OQS Testserver

Um verlässlichere und realitätsnähere Daten zu generieren, wird auf einen Open Quantum Safe bereitgestellten Testserver zurückgegriffen. Die Adresse dieses Servers lautet `test.openquantumsafe.org`.[69] Er zeichnet sich durch die Verfügbarkeit von über 400 unterschiedlichen Ports aus, die jeweils verschiedene Kombinationen von PQC Algorithmen und traditionellen kryptografischen Algorithmen nutzen.

Zunächst beschreibt ein Skript die Durchführung täglicher Tests verschiedener Signaturalgorithmen und Kurven. Es gibt vordefinierte Listen von Signaturalgorithmen und Kurven, die getestet werden sollen. Eine Funktion namens `run_daily_tests` wird definiert. Sie öffnet und liest eine Datei namens `assignments.json`. Diese Datei beinhaltet alle verfügbaren Kombinationen und die dazugehörigen Ports. Es wird überprüft, ob der Signaturalgorithmus und die Kurve in den vordefinierten Listen vorhanden sind. Ist die Bedingung erfüllt, wird ein externes Skript mit den Parametern Port und Kurve aufgerufen.

Das externe Skript misst die durchschnittliche TLS-Handshake Zeit für die bestimmten Kurven. Zunächst wird eine URL basierend auf dem angegebenen Port erstellt. Das Skript startet dann einen Docker-Container mit der OQS-Version von cURL, der PQC-Algorithmen unterstützt. In einer Schleife von fünf Iterationen, führt es den Befehl `curl` im Docker-Container aus, um die TLS-Handshake-Zeit zu messen. Es berechnet die tatsächliche TLS-Handshake-Zeit basierend auf den Ausgaben von `curl`. Dabei werden die zwei Parameter `time_connect` und `time_appconnect` verwendet, welche mithilfe der cURL-Funktion `write-out` erfasst werden. Mit Hilfe dieser Parameter können verschiedene Zeiten des

[69] (OQS, 2023)

Handshakes einer TLS-Verbindung ermittelt werden. Die Zeit des TLS-Handshakes wird berechnet, indem von der `time_appconnect` von `time_connect` subtrahiert wird. Dieser berechnete Wert wird zur fortlaufenden Summe von `total_tls_handshake_time` addiert, die den gesamten Handshake über alle Durchläufe hinweg erfasst und daraus der Durchschnitt berechnet wird.

Die resultierende durchschnittliche TLS-Handshake-Zeiten werden schließlich in einer CSV-Datei gespeichert.

In einer Zeitspanne von zehn Tagen wurde dieser Test zweimal täglich durchgeführt, um möglichen Schwankungen der Netzwerklatenz und anderen Echtzeit-Varianzen entgegenzuwirken und eine möglichst präzise Messung zu gewährleisten. Durch das Wiederholen der Messungen über mehrere Tage hinweg können zufällige Abweichungen oder Anomalien ausgeglichen werden, die an einem bestimmten Tag auftreten könnten.

Fehlerhafte Version

Ursprünglich wurde der Parameter `time_pretransfer` verwendet. Die Nichtverwendung von `time_appconnect` lag daran, dass ein potenzielles Risiko der Unzuverlässigkeit in der Implementierung der OQS cURL-Version bestand. Obwohl `time_appconnect` im Idealfall die Zeit messen würde, die zum Herstellen der Verbindung zum Server und zum Abschluss des TLS-Handshakes benötigt wird, tauchten in dieser speziellen cURL-Version Inkonsistenzen in dieser Metrik auf. Der Parameter `time_appconnect` bezeichnet die Dauer, die benötigt wird, um die Verfahren des TLS-Handshakes vollständig abzuschließen. Dies ist der Zeitraum, in dem Sicherheitsprotokolle zwischen dem Client und dem Server ausgetauscht werden, um eine verschlüsselte Verbindung herzustellen. Der Parameter `time_total` stellt die Gesamtdauer der Verbindung dar. In Listing 2 hat `time_appconnect` einen größeren Wert als `time_total`, was bedeuten würde, dass das Handshake länger gedauert hat als die gesamte Verbindung und dies ist nicht möglich.[70]

[70] (curl, 2023)

```
1 docker run -v `pwd`:/ca -it openquantumsafe/curl curl --cacert /ca/CA.crt
https://test.openquantumsafe.org:6001 --curves bikel1 -s -w
"%{time_namelookup},%{time_connect},%{time_appconnect},%{time_pretrans-
fer},%{time_starttransfer},%{time_total}" -I

2 0.010298, 0,210697, 0.482655, 0.272001, 0.303177, 0.303287
```

Listing 2: Output der fehlerhafte cURL-Version

Wegen der fehlerhaften Version wurde der Ansatz gewählt, die TLS-Handshake-Zeit durch Kombination anderer verfügbarer Metriken zu berechnen. `time_pretransfer` stellt die gesamte Dauer dar, die vom Beginn der Anfrage bis zum unmittelbaren Beginn der eigentlichen Datenübertragung vergeht. Es war zu beachten, dass die Integration dieser Zeitspannen in Gesamtlatenzmessungen zu potenziellen Verzerrungen der Ergebnisse führen kann. Dennoch stellte diese Methode, angesichts der bekannten Probleme, die präziseste Herangehensweise zur Messung dar.

Nach Initiierung eines Github-Issues und einer Diskussion mit den Entwicklern von OQS wurde festgestellt, dass eine defekte cURL-Version zum Einsatz kam und daher die Ergebnisse von `time_appconnect` nicht korrekt waren. Weitere Untersuchungen ergaben, dass eine vorherige cURL-Version dieses Problem nicht aufwies. Diese Erkenntnisse wurden den Entwicklern mitgeteilt, woraufhin eine Korrekturmaßnahme durch einen Merge-Prozess eingeleitet wurde, der Fehler behoben und `time_appconnect`, wie ursprünglich geplant, verwendet wurde.

5.4 Ergebnisse

Die aus den Tests ermittelten Daten zeigen die durchschnittliche Zeit, die für das TLS-Handshake unter Verwendung spezifischer hybrider und reiner PQC-Algorithmen benötigt wird. Sie sind in Millisekunden angegeben.

In den Dokumenten Ecdsap256.csv (B.2.3), Rsa3072.csv (B.2.2), P384_dilithium3.csv (B.2.4) und Dilithium3.csv (B.2.1) wurden die Daten gespeichert. Jedes Dokument ist in drei Spalten gegliedert: das Datum der Durchführung der Messung, die Bezeichnung der verwendeten kryptografischen Kurve oder des Algorithmus und die durchschnittliche Zeit, die für die kryptografische Operation aufgebracht wurde. Dabei wurden klassische Algorithmen wie ECDSA und RSA sowie PQC-Algorithmen, wie Dilithium, in Betracht gezogen. In der einleitenden Sektion wurde darauf hingewiesen, dass der primäre Fokus dieser Arbeit nicht auf der Evaluierung von Signaturalgorithmen liegt. Nichtsdestotrotz wurden diverse Signaturalgorithmen in die Untersuchung integriert, um potenzielle Abweichungen zu identifizieren, die als Ausgangspunkt für nachfolgende wissenschaftliche Untersuchungen dienen können.

Die Auswahl der untersuchten Algorithmen umfasst sowohl reine PQC-Algorithmen und klassische Algorithmen als auch hybride Lösungen, die auf ihre Anwendungstauglichkeit in einer zukünftigen quantencomputergestützten Umgebung geprüft wurden. Hierbei wurden verschiedene Varianten und Parametergrößen erfasst, darunter solche mit der Bezeichnung `bikel1`, `bikel3`, `hqc128`, `hqc192`, `kyber512` und andere. Die dokumentierten Zeiten beziehen sich auf die gemessene Zeit, angefangen beim `ClientHello` des Clients bis zum `Finished` Nachricht.[71]

5.4.1 Lokale TLS-Handshake Performance

Für den lokalen Test wurde die Anzahl der erfolgreichen Handshakes in einer Sekunde verglichen. Es wurde festgestellt, dass unter den hybriden Algorithmen der Algorithmus `x25519_kyber512` durchgängig die beste Leistung erzielte, mit den höchsten durchschnittlichen TLS-Handshakes pro Sekunde mit allen vier Signaturalgorithmen. Die schlechteste Leistung unter den hybriden Algorithmen wurde vom Algorithmus `p521_bikel5` gezeigt, der in jeder Kombination die niedrigsten Werte aufwies.

Bei den nicht-hybriden Algorithmen wies `kyber768` die höchste Leistung mit den Signaturalgorithmen `dilithium3`, `ed448` und `ed25519` auf, während `kyber512` mit

[71] (Rescorla, 2018)

`p384_dilithium3` als bester nicht-hybrider Algorithmus identifiziert wurde. Der Algorithmus `bikel5` zeigte hingegen die geringste Leistung unter den nicht-hybriden Algorithmen in allen Kombinationen.

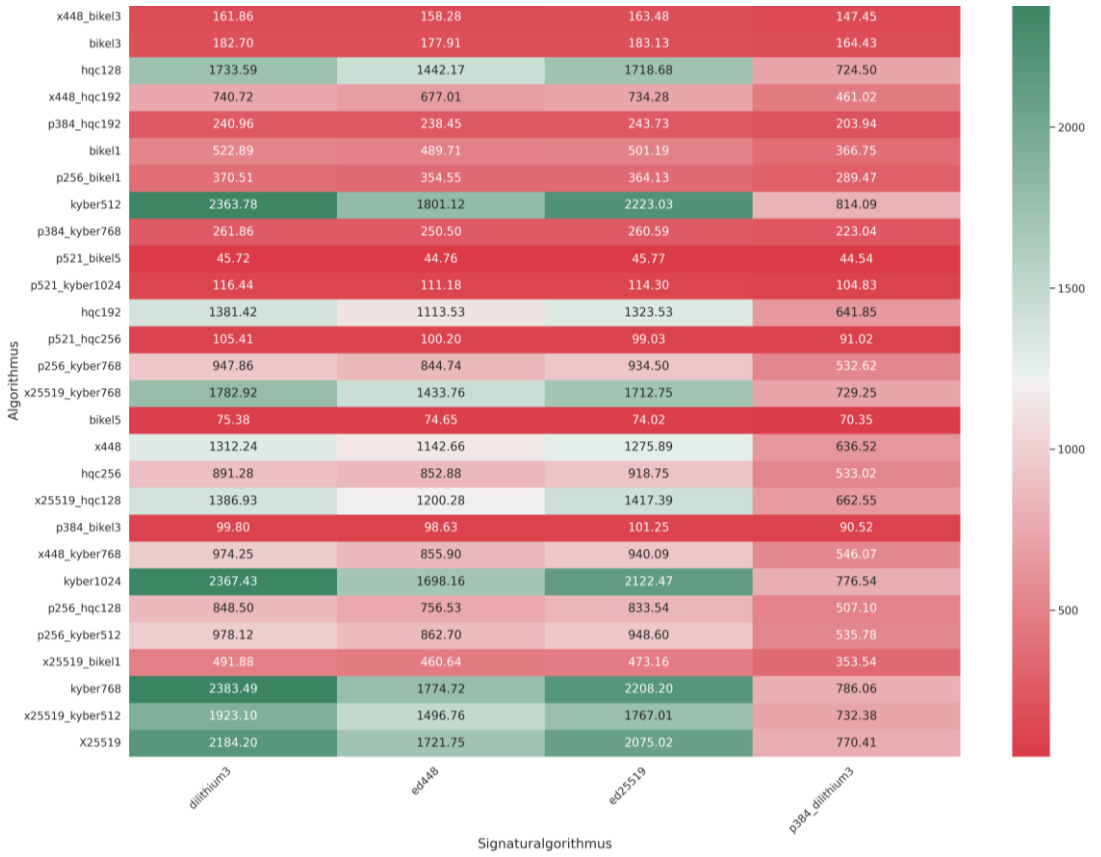

Abbildung 1: Heatmap der Anzahl der durchschnittlichen Handshakes pro Sekunde in verschiedenen Kombinationen

5.4.2 OQS Testserver

Im Rahmen der Untersuchung der TLS-Handshakezeit mit den RSA3072 und ECDSAP256 Signaturalgorithmus wurden insgesamt 864 Durchschnittswerte über einen Datensatz von 27 unterschiedlichen kryptografischen Algorithmen ermittelt. Dieses Spektrum umfasste jeweils 9 PQC-Algorithmen, 17 hybride Algorithmen sowie einen klassischen Algorithmus. Die Datenerhebung für diese Evaluation erstreckte sich über einen Zeitraum von 11 Tagen.

RSA3072

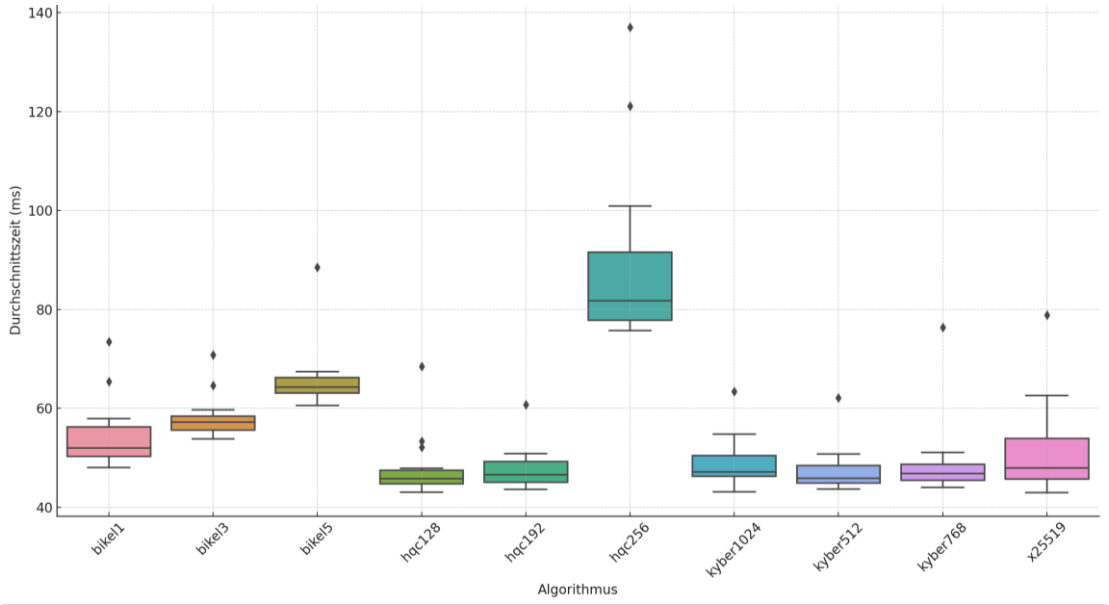

Abbildung 2: Boxplot der durchschnittlichen Zeiten für die PQC-Algorithmen und den klassischen Algorithmus mit RSA3072 als Signaturalgorithmus

Bei `bikel1` wurde eine durchschnittliche Ausführungszeit von 54,59 ms festgestellt. Die Beobachtungen reichten von einem Minimum von 48,06 ms bis zu einem Maximum von 73,48 ms. Der Algorithmus `bikel3` zeigte eine durchschnittliche Ausführungszeit von 58,12 ms. Die Werte bewegten sich zwischen 53,86 ms und 70,80 ms, was eine konsistentere Leistung als `bikel1` anzeigt. Für `bikel5` wurde eine höhere durchschnittliche Ausführungszeit von 65,76 ms festgestellt und hat mit 88,55 ms die höchste Maximalzeit in der BIKE-Kategorie.

Bei den HQC-Algorithmen haben `hqc128` und `hqc192` ähnliche durchschnittliche Ausführungszeiten von ca. 48 ms. `hqc256` unterschied sich mit einer deutlich höheren durchschnittlichen Ausführungszeit von 88,44 ms und einer beträchtlichen Standardabweichung von 17,56 ms, was eine wesentlich größere Inkonsistenz im Vergleich zu den anderen HQC-Varianten

zeigt. Dies spiegelt sich auch in der Spannweite der Ausführungszeiten wider, mit einem Minimum von 75,69 ms und einem auffallend hohen Maximum von 137,04 ms.

Alle Kyber-Varianten hatten eine sehr ähnliche durchschnittliche Zeit von ca. 47 bis 49 ms und relativ geringe Standardabweichungen.

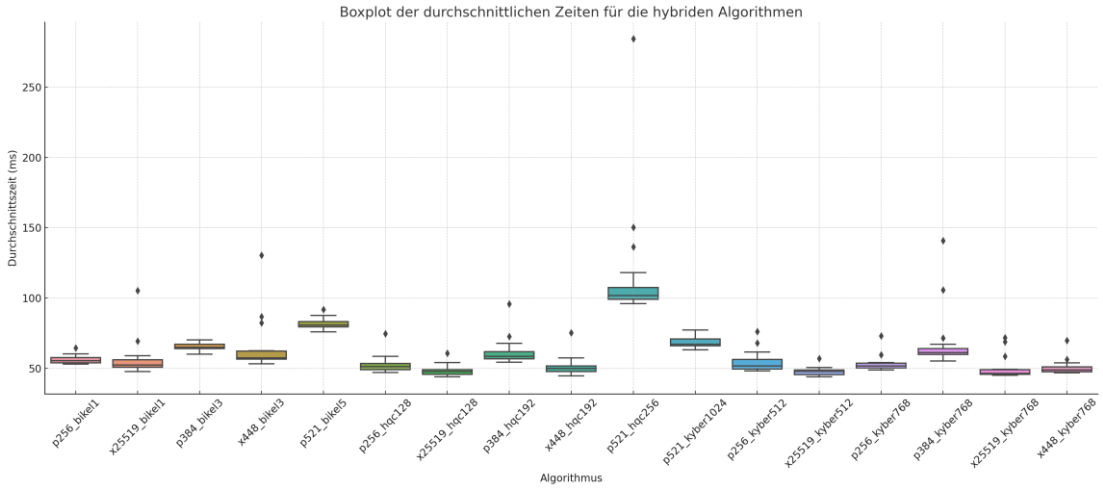

Abbildung 3: Boxplot der durchschnittlichen Zeiten für die hybriden Algorithmen mit RSA3072 als Signaturalgorithmus

Einige hybride Algorithmen wie `p256_bikel1` gewährleisten mit einer Durchschnittszeit von rund 56 ms und einer geringen Standardabweichung von 3 ms eine hohe Stabilität in der Messung. Im Kontrast dazu steht die ausgeprägte Inkonstanz von Algorithmen wie `p521_hqc256`, der mit einer Durchschnittszeit von über 118 ms und einer Standardabweichung von rund 47 ms eine signifikante Schwankungsbreite zeigt, dass ebenfalls durch Extremwerte von 75,69 ms bis zu 284,38 ms bestätigt wird.

Auch bei anderen Algorithmen wurde zahlreiche Messzeiten und Konsistenzniveaus festgestellt. `p384_hqc192` und `p384_kyber768` beispielsweise zeigten längere Ausführungszeiten und höhere Standardabweichungen. Andererseits wiesen die X25519-Algorithmen durchweg niedrige Durchschnittszeiten von etwa 48 ms und niedrige Standardabweichungen auf. Lediglich `x25519_bikel1` fällt mit einer etwas höheren Standardabweichung und einem Maximalwert von ca. 105 ms aus dem Rahmen.

Die X448-Algorithmen haben moderate Durchschnittszeiten, allerdings bei `x448_bikel3` mit einer hohen Standardabweichung, die auf größere Inkonsistenzen in der Ausführung hindeutet. Insgesamt zeichnet die Messung eine Bandbreite von unter 50 ms bis über 100 ms bei den Ausführungszeiten und bei den Standardabweichungen eine Varianz von sehr stabil (2 ms) bis hin zu sehr variabel (fast 50 ms) auf.

ECDSAP256

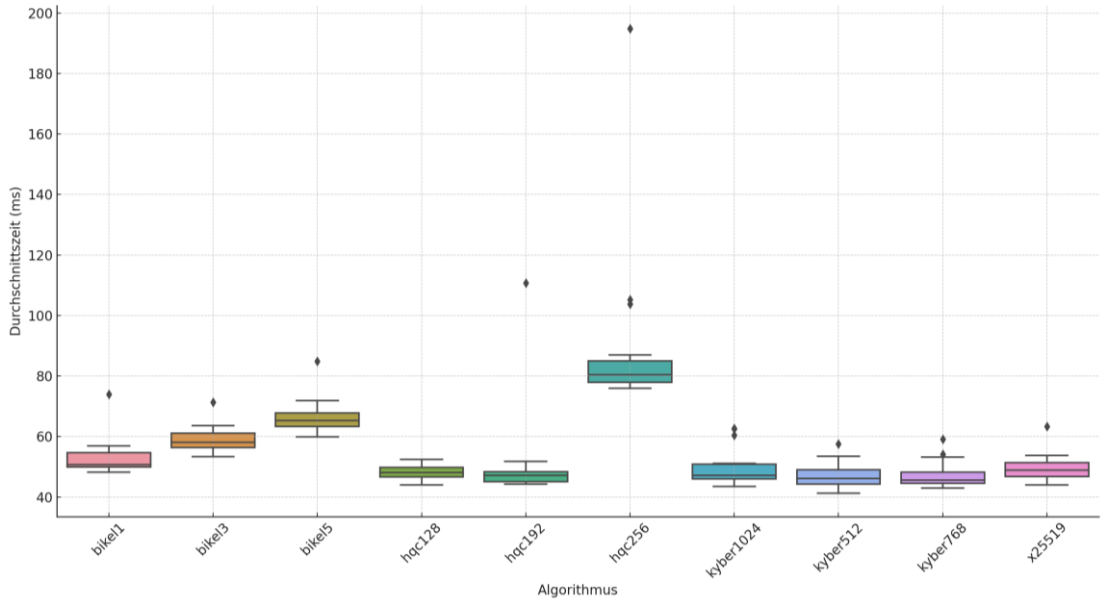

Abbildung 4: Boxplot der durchschnittlichen Zeiten für PQC-Algorithmen und den klassischen Algorithmus mit ESDSAP256 als Signaturalgorithmus

Bei den Handshakezeiten mit ECDSAP als Signaturalgorithmus hat `hqc128`, mit einer durchschnittlichen Zeit von 48.11 ms die geringste Zeit. Die Zeitspanne bleibt mit einem Maximum von nur 52.44 ms eng am Durchschnitt. Im Gegensatz dazu zeigt `hqc192` eine erheblich breitere Streuung, eine hohe Standardabweichung von 16.11 ms und eine Spitze von 110.79 ms im Maximum. `hqc256` hebt sich mit der größten durchschnittlichen Ausführungszeit von 89.97 ms ab und die Streubreite ist mit einem Maximum von fast 200 ms groß.

Bei den Kyber-Algorithmen bietet `kyber1024` eine ausgeglichene Leistung mit moderaten Zeitschwankungen, die zwischen 43.49 ms und 62.61 ms liegen. `kyber512` und `kyber768` sind mit etwas geringeren Durchschnittszeiten ähnlich mit engen Bereichen von Minimum zu Maximum.

`bikel1` zeigt eine stärkere Schwankung mit Zeiten, die bis zu 74.00 ms erreichen, während `bikel3` und `bikel5` durchweg höhere Durchschnittszeiten aufweisen, jedoch mit geringeren Abweichungen vom Median.

`x25519` zeigte eine durchschnittliche Ausführungszeit von 62.84 ms.

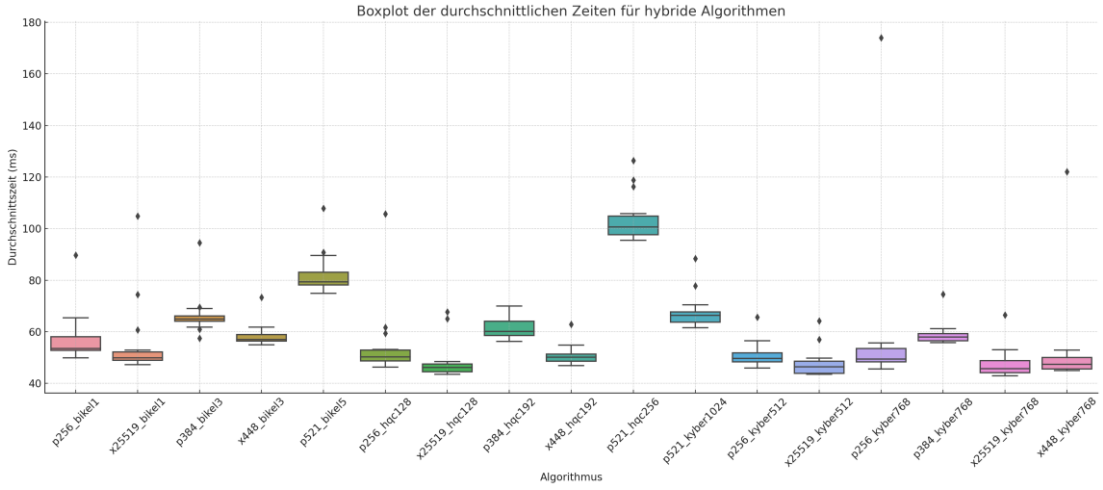

Abbildung 5: Boxplot der durchschnittlichen Zeiten für die hybriden Algorithmen mit ESDSAP256 als Signaturalgorithmus

Der Algorithmus `p256_kyber512` zeigt mit einer durchschnittlichen Ausführungszeit von 50,93 ms und einer Standardabweichung von 4,63 ms die effizienteste und stabilste Leistung unter den betrachteten P256-Hybriden. Bei `p256_hqc128` wird eine überdurchschnittliche Ausführungszeit von 54,55 ms verzeichnet. Auch `p256_bikel1` zeigt eine längere mittlere Ausführungszeit von 57,37 ms und eine größere Streuung mit einer Standardabweichung von 9,63 ms.

Für die P384-basierten Algorithmen wurde eine höhere durchschnittliche Zeit im Vergleich zu den P256-Algorithmen festgestellt, wobei `p384_kyber768` und `p384_hqc192` mittlere Zeiten von etwa 59 ms bieten. Der Algorithmus `p384_bikel3` wies jedoch eine höhere durchschnittliche Ausführungszeit von 66.43 ms auf.

P521-Algorithmen hatten die längsten durchschnittlichen Zeiten. `p521_kyber1024` und `p521_bikel5` zeigten mittlere Ausführungszeiten von 67.85 ms bzw. 82.49 ms, während `p521_hqc256` mit einer durchschnittlichen Zeit von 103.94 ms die längste Ausführungszeit aller analysierten Algorithmen aufwies.

Für die X25519-basierten Algorithmen konnte allgemein eine effizientere Leistung gemessen werden, wobei die durchschnittlichen Ausführungszeiten für `x25519_kyber768` und `x25519_kyber512` bei 47.60 ms bzw. 47.77 ms liegen, aber höhere Standardabweichungen.

Bei den X448-basierten Algorithmen wurde eine moderat höhere durchschnittliche Ausführungszeit festgestellt. Insbesondere bei `x448_hqc192` wurde eine Ausführungszeit von 50.85 ms beobachet. Der Algorithmus `x448_kyber768` zeigte eine Standardabweichung von 18.75 ms. `x448_bikel3` hingegen wies eine durchschnittliche Ausführungszeit von 58.38 ms auf.

Hybride und PQC-Signaturalgorithmen

Die Messung des TLS-Handshakes mit dem Dilithium3 Algorithmus, die 160 Beobachtungen umfasst, hat eine durchschnittliche Dauer 64,90 ms und eine Standardabweichung von 16,59 ms. Die Handshake-Zeiten variierten zwischen einem Minimum von 41,93 ms und einem Maximum von 132,06 ms. Bei den getesteten hybriden Algorithmen verzeichnete die Kurve `p384_hqc192` die höchste Durchschnittszeit, mit etwa 84,94 ms und `x448_kyber768` die niedrigste Zeit mit 47,22 ms. Bei den PQC-Algorithmen zeigte `kyber768` hingegen mit einem Durchschnitt von circa 46,32 ms die niedrigste Zeit und `hqc192` die höchste mit 79,07 ms.

Für die Messung des hybriden Signaturalgorithmus `p384_dilithium3`, welche 240 Beobachtungen beinhaltet, wurde eine durchschnittliche Handshake-Dauer von 77,65 ms

ermittelt. Das Spektrum der Handshake-Zeiten reichte von 51,52 ms bis zu 116,12 ms. Es fiel auf, dass bei den hybriden Algorithmen `p384_hqc192` mit einer durchschnittlichen Handshake-Zeit von rund 85,96 ms die längste Zeit benötigte und `x448_kyber768` die kürzeste Zeit mit 75,2 ms. Im Gegensatz dazu hatte bei den PQC-Algorithmen `kyber768`, mit etwa 72,42 ms die kürzeste durchschnittliche Zeit und `hqc192`, mit 79,01 ms, die längste Zeit.

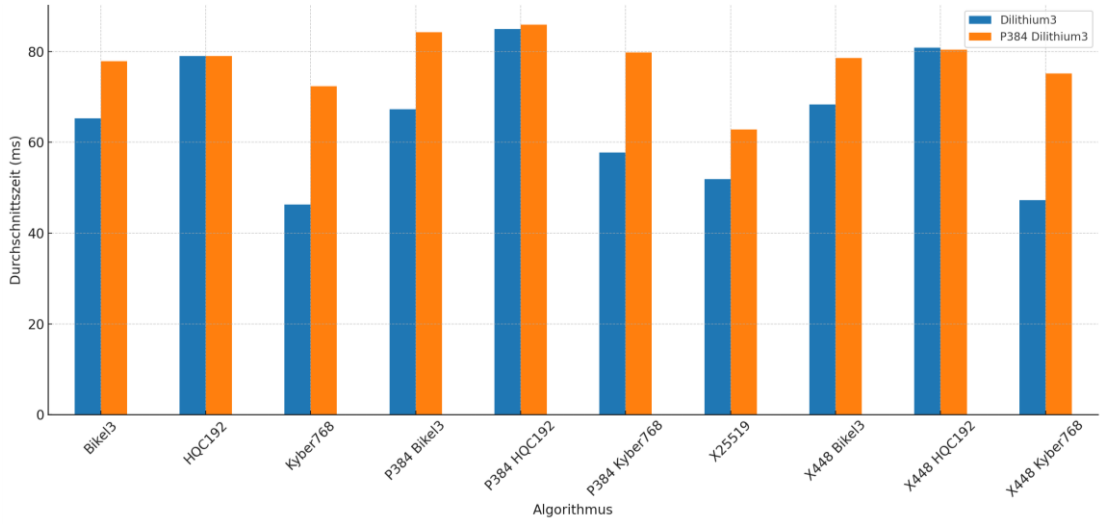

Abbildung 6: Durchschnittliche Handshakezeiten für hybride und PQC-Algorithmen

6 Evaluation der Ergebnisse

Die Evaluation geht von der Prämisse aus, dass die Entwicklung von Quantencomputern, die aktuelle Verschlüsselungsmethoden brechen können, wahrscheinlich ist, wodurch ein Wechsel zu PQC unumgänglich wird. Daher liegt der Fokus nicht auf dem Vergleich mit rein klassischen Algorithmen, sondern auf der Annahme, dass leistungsstarke Quantencomputer Realität werden. In diesem Kontext wird die Entscheidung zwischen hybridem Ansatz und der ausschließlich auf PQC beschränkten Alternative betrachtet.

6.1 Speicherverbrauch

Die Nutzung von kombinierten, also hybriden Methoden führt zwangsläufig dazu, dass mehr Speicherplatz benötigt wird. Falls mindestens PQC zum Einsatz kommt, wird durchgängig mehr Speicher als bei Verwendung klassischer Algorithmen benötigt. Die Daten in Tabelle 3 zeigen, dass die Schlüsselgrößen der klassischen Algorithmen an sich recht klein sind. Wenn man diese mit PQC kombiniert, steigt die Größe der Schlüssel bei den hybriden Systemen nur leicht an. Bei der Nutzung des TLS-Protokolls mit direktem Einfluss der Größe der Schlüssel auf die Handshake-Dauer und die Bandbreitennutzung sind diese erhöhten Anforderungen besonders bedeutsam. In Umgebungen mit beschränkten Ressourcen beeinträchtigt der zusätzliche Speicherbedarf für private Schlüssel, öffentliche Schlüssel und Outputs bei hybriden Algorithmen allerdings die Performance.

Die Zunahme in der Größe der Schlüssel ist bei reinem Kyber-Algorithmus im Vergleich zur Kombination aus Kyber und einem anderen Algorithmus (zum Beispiel `p256_kyber512`) nur klein. Bei `kyber512` erhöht sich die Größe des privaten Schlüssels um ungefähr 2% (von 1632 Bytes auf 1664 Bytes) und die des öffentlichen Schlüssels um etwa 8% (von 800 Bytes auf 865 Bytes), wenn man zu `p256_kyber512` wechselt. Kyber selbst zeichnet sich durch die kleinsten Schlüsselgrößen unter den ausgewählten Algorithmen aus. Daher ist der Anstieg der Schlüssellängen bei Kyber am deutlichsten. Auch bei `kyber768` und `kyber1024`, kombiniert mit Algorithmen, die den Sicherheitsstandards des NIST entsprechen (wie P384 oder

P521), ist der Anstieg ähnlich: etwa 2% bei privaten Schlüsseln und 8% bei öffentlichen Schlüsseln. Bei anderen Algorithmen ist der Anstieg noch geringer.

Im Vergleich dazu gibt es Algorithmen wie Classic McEliece, die viel größere Schlüssellängen aufweisen als die anderen. Die kleinste Variante von Classic McEliece, Classic-McEliece-348864, hat in der von OQS implementierten Version einen öffentlichen Schlüssel von 261120 Bytes. Diese Größe führte zu einem Ausschluss in der TLS-Version aus OQS.[72] Solche Algorithmen sind für Protokolle wie TLS wegen des Speicherverbrauchs nicht effizient genug. Trotz umfangreicher Tests und im Vergleich zu anderen PQC-Algorithmen hoher Sicherheit gilt Classic McEliece eher als letzte Option, falls alle anderen Algorithmen fehlschlagen.[73]

6.2 Lokale Tests

Bei der lokalen Durchschnittszeit für TLS-Handshakes ist der hybride Signaturalgorithmus mit etwa 447 Verbindungen pro Sekunde am langsamsten. Danach folgt ED448 mit 794 Verbindungen, ED25519 mit 920 Verbindungen und Dilithium3 ist mit 969 Verbindungen am schnellsten. Dies deutet darauf hin, dass reine PQC-Signaturalgorithmen, in dem Fall gitterbasierte, sogar schneller sein können als die aktuellen klassischen Algorithmen. Als hybride Methode sind die einhergehend mit Leistungseinbußen deutlich langsamer.

[72] (D. Stebila, 2023)

[73] (McEliece, 2023)

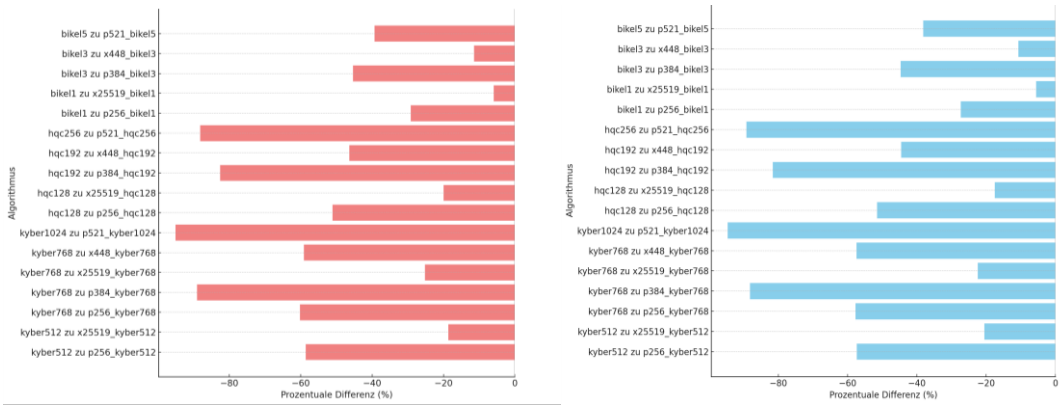

Abbildung 7: Prozentuale Differenz zwischen PQC- und Hybridvarianten mit den Signaturalgorithmen Dilithium3(rot) und ED25519(blau)

Bei der Betrachtung der prozentualen Differenzen zwischen verschiedenen PQC-Algorithmen und ihren hybriden Varianten werden Besonderheiten festgestellt. Die prozentualen Abweichungen zwischen den PQC und hybriden Algorithmen sind bei allen Signaturalgorithmen relativ konsistent. Diese Konsistenz in den Differenzen ist zu erwarten, da die Messungen des Schlüsselaustauschs stets unter Verwendung derselben Signaturalgorithmen durchgeführt werden.

Zudem wird beobachtet, dass bei den Kyber- und HQC-Varianten eine deutliche Leistungsminderung auftritt, wenn sie mit elliptischen Kurven wie P256, P384, P521, X25519 und X448 kombiniert werden. Insbesondere wird eine enorme Differenz von ca. -95% bei `kyber1024` zu `p521_kyber1024` und 89% bei `hqc256` zu `p521_hqc256` festgestellt.

Im Gegensatz dazu werden bei den BIKE-Varianten weniger Differenzen ermittelt, wie beispielsweise -6% bei `bike1` zu `x25519_bikel1` und -11% bei `bikel3` zu `x448_bikel3`. Auffällig ist, dass besonders hohe Differenzen auftreten, sobald die Algorithmen mit stärkeren Varianten von elliptischen Kurven kombiniert werden. Dies deutet in diese Kombinationen auf eine mögliche grundsätzliche Inkompatibilität oder auf erhöhte Rechenanforderungen hin.

Unter den reinen PQC-Algorithmen sticht `kyber512` mit 2223 Verbindungen pro Sekunde als leistungsstärkster heraus, dicht gefolgt von `kyber768` und `kyber1024`. Diese hohe Anzahl an Verbindungen deutet auf eine effiziente Performance hin. Bei den Hybrid-Kombinationen ragt `x25519_kyber512` mit 1767 Verbindungen pro Sekunde hervor. Trotz einer

Reduzierung ca. 21% im Vergleich zur reinen Kyber512-Version bleibt diese Kombination am leistungsstärksten, sogar unter den reinen PQC-Algorithmen wie `hqc128` mit 1718 Verbindungen pro Sekunde und `bikel1` mit 501 Verbindungen pro Sekunden.

Über die verschiedenen Algorithmen erweist sich als ein konstantes Muster, dass die Differenzen mit der Verwendung stärkerer elliptischer Kurven tendenziell zunehmen. Diese Erkenntnisse dürften für die Auswahl und Optimierung von kryptografischen Protokollen und Systemen von Bedeutung sein, da die Kombination dieser kryptografischen Algorithmen mit verschiedenen Varianten von elliptischen Kurven zu signifikanten Leistungsveränderungen führen kann. Die Entscheidung für die beste Kombination hängt letztlich von mehreren Faktoren ab, einschließlich der benötigten Sicherheitsstufe und der verfügbaren Systemressourcen.

6.3 OQS Testserver

RSA3072

Bei der Messung der TLS-Handshakezeit des OQS-Servers wurde ebenfalls festgestellt, dass die hybriden Algorithmen eine höhere durchschnittliche Gesamtzeit von 61,97 ms aufweisen. Diese Beobachtung könnte auf unterschiedliche Leistungseigenschaften der verschiedenen Kombinationen von Algorithmen hinweisen.

Im Gegensatz dazu wurden bei den PQC-Algorithmen konsistentere und im Durchschnitt niedrigere Handshake-Zeiten von 55,9 ms beobachtet. Die geringere Standardabweichung lässt auf eine stabilere Performance über verschiedene Implementierungen hinweg schließen. Diese Befunde deuten darauf hin, dass PQC-Methoden mit kürzeren und konsistenteren Handshake-Zeiten im Allgemeinen effizienter sind.

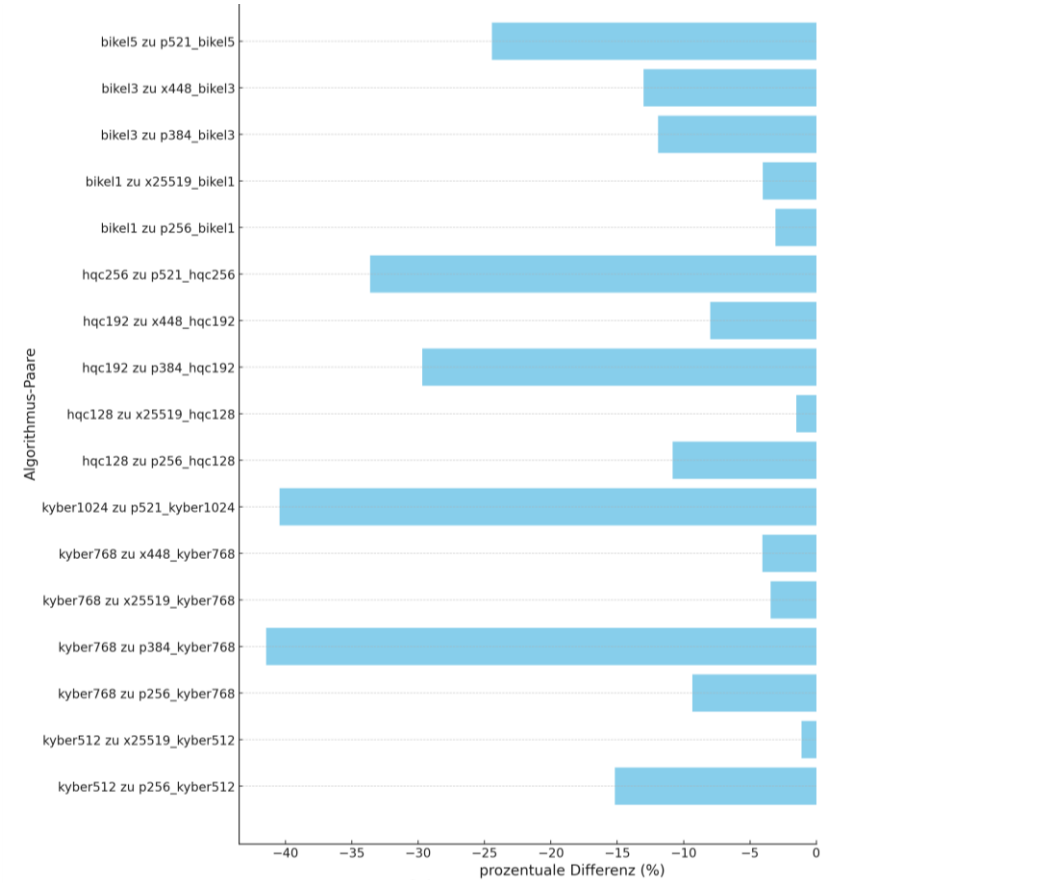

Abbildung 8: Prozentuale Differenz zwischen PQC- und Hybridvarianten mit den Signaturalgorithmen RSA3072

Es ist auffallend, dass mit zunehmendem Sicherheitsniveau auch die durchschnittliche Dauer des TLS-Handshakes ansteigt. Diese Beobachtung ist besonders ausgeprägt bei Algorithmen mit höheren Sicherheitsniveaus, während bei niedrigeren Sicherheitsniveaus die prozentuale Differenz geringer ist. Weiterhin zeigt sich, dass Kombinationen mit den elliptischen Kurven P256, P384 und P521 generell eine schlechtere Leistung aufweisen als die Kombinationen mit den Kurven X448 und X25519. Letztere weisen durchgängig eine geringere Differenz in der Handshake-Zeit auf, mit Ausnahme von `bikel1`, wo P256 führend ist.

Wie in Abbildung 3 dargestellt, weisen hybride Algorithmen der Stufe 1 und 3 die beste Performance auf. Bei Varianten der Stufe 3 hängt die Performance zusätzlich stark davon ab, mit

welchem klassischen Algorithmus sie kombiniert werden. Beispielsweise hat `kyber768` eine durchschnittliche TLS-Handshake-Zeit von 48,78 ms. Im Vergleich dazu liegt die Differenz zu `p384_kyber768` (69 ms) bei -41,45%, zu `x448_kyber768` (50,77 ms) bei -4,07% und zu `x25519_kyber768` (50,47 ms) sogar bei -3,45%.

ECDSAP256

Bei den Messungen mit ECDSAP256 als Signaturalgorithmus haben die PQC-Algorithmen ebenfalls eine durchschnittlich niedrigere Ausführungszeit von etwa 56,12 ms, während die hybriden Algorithmen durchschnittlich 60,15 ms benötigen. Dies deutet darauf hin, dass die PQC-Algorithmen geringfügig schneller sind als die hybriden Algorithmen. Der Unterschied hier ist ebenfalls relativ klein: beide Arten von Algorithmen sind in Bezug auf die Ausführungsgeschwindigkeit vergleichbar.

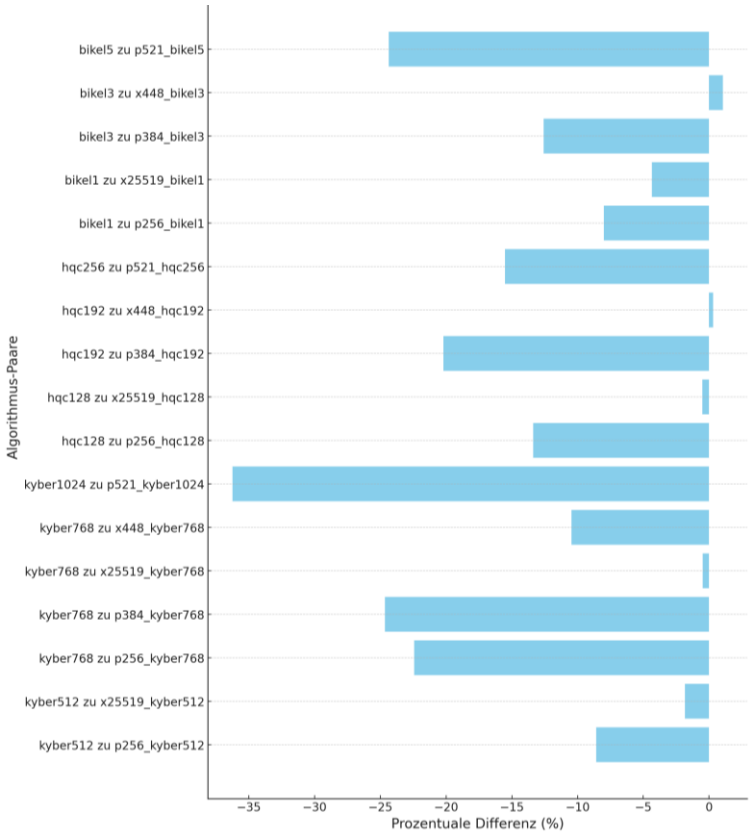

Abbildung 9: Prozentuale Differenz zwischen PQC- und Hybridvarianten mit den Signaturalgorithmen ECDSAP256

Des Weiteren zeigt sich erneut, dass die Kombination der elliptischen Kurven X25519 und X448 eine höhere Leistungsfähigkeit aufweist als die Kurven P256, P384 und P521. Die Level 1 und 3 Varianten haben wieder die geringsten Handshakezeiten und die geringste Differenz zu den reinen PQC-Algorithmen. Eine interessante Besonderheit bei der Kurve ECDSAP256 ist zu beobachten: Einige hybride Algorithmen weisen eine bessere Leistung auf als ihren reinen PQC-Varianten, was sich in einer positiven prozentualen Differenz zeigt. Bemerkenswert ist hierbei, dass diese positive Differenz stets in Kombination mit der Kurve X448 auftritt, beispielsweise von `hqc192` (50,99 ms) zu `x448_hqc192` (50,84 ms) mit einer Differenz von +0,3 % und von `bikel3` (58,99 ms) zu `x448_bikel3` (58,38 ms) mit einer Differenz von +1,04 %. Die Messungen sind möglicherweise nicht ausreichend oft durchgeführt wurden, um einen statistisch signifikanten Unterschied zu begründen.

Hybride und PQC-Signaturalgorithmen

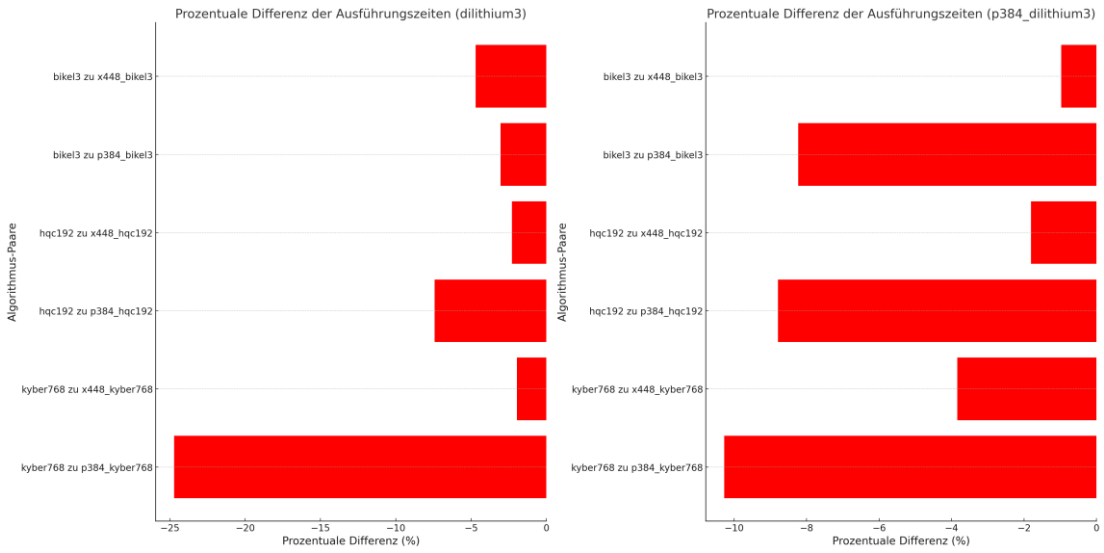

Abbildung 10: Prozentuale Differenz zwischen PQC- und Hybridvarianten mit den Signaturalgorithmen Dilithium3 (links) und P384_Dilithium3 (rechts)

Im Durchschnitt weisen die hybriden Algorithmen auch bei diesen Signaturalgorithmen höhere TLS-Handshakezeiten als reine PQC-Algorithmen. Die Auswahl der Kurve scheint erneut eine

Rolle bei der Performance zu spielen. So zeigen die Kombinationen mit X448 im Allgemeinen mit kürzeren und konsistenteren Handshake-Zeiten geringere Performance-Einbußen im Vergleich zu den P384 Varianten.

6.4 Diskussion

In der Diskussion ist die Effektivität und Praktikabilität hybrider Algorithmen die verbleibende zentrale Fragestellung. Wie bereits im Abschnitt 4.4, Position der NSA zu hybriden Algorithmen, hingewiesen, herrscht in der wissenschaftlichen Gemeinschaft eine große Diskrepanz hinsichtlich der Bewertung und Anwendung hybrider Methoden. In weitern von der IETF veröffentlichten E-Mails äußert Uri Blumenthal, ein Forscher am Lincoln Laboratory des MIT, seine Ansichten hierzu[74]:

In der Annahme der Produktion für Kryptografie relevante Quantencomputer in den nächsten zwei Jahrzehnte wird die Anwendung hybrider Lösungen in den meisten Fällen als, aus dem Englischen übersetzt, „unnötiger Ballast und Steigerung der Komplexität von Prozessen“ (Blumenthal, 2021) bezeichnet. In seiner Ausarbeitung identifiziert Blumenthal acht mögliche Szenarien, die auf der Grundlage von drei Schlüsselparametern formuliert werden: die Existenz von kryptografisch relevanten Quantencomputern (KRQC), die fortgesetzte Effektivität klassischer Algorithmen gegenüber herkömmlicher Kryptoanalyse und die Funktionsfähigkeit von PQC-Algorithmen.

[74] (Blumenthal, 2021)

Nr.	***KRQC vorhanden?***	***Verteidigung klassischer Angriffe***	***PQC-Algorithmen***	***Hybrid-Status***
1	Ja	Erfolgreich	Funktionsfähig	Nutzlos
2	Ja	Erfolgreich	Fehlgeschlagen	Nutzlos
3	Ja	Angreifbar	Funktionsfähig	Nutzlos
4	Ja	Angreifbar	Fehlgeschlagen	Nutzlos
5	Nein	Erfolgreich	Funktionsfähig	Nutzlos
6	Nein	Erfolgreich	Fehlgeschlagen	Hilfreich
7	Nein	Angreifbar	Funktionsfähig	Nutzlos
8	Nein	Angreifbar	Fehlgeschlagen	Nutzlos

Tabelle 4: Bewertung verschiedener Szenarien bezüglich hybriden Lösungen

Demnach bietet die hybride Methode in lediglich 12,5 % der Fälle einen Nutzen, während in 87,5 % der Fälle kein Vorteil hervorgeht. Diese Wahrscheinlichkeiten der Fälle sind jedoch nicht so gleichmäßig verteilt, wie es auf den ersten Blick scheinen mag. Insbesondere sind die Szenarien 4 und 8 äußerst unwahrscheinlich, da man vernünftigerweise annehmen kann, dass zumindest ein asymmetrischer Algorithmus Bestand haben wird. Generell sind die Fälle, in denen kein Quantencomputer existiert und die klassischen Algorithmen nicht mehr sicher sind, unwahrscheinlich. Dies begründet sich darin, dass diese Algorithmen bereits seit mehr als 25 Jahren existieren und eingesetzt werden. Selbst wenn jedoch alle diese Wahrscheinlichkeiten identisch wären, ist das in Abschnitt 4.2 beschriebene Szenario 6 bereits eingetreten. Ein leistungsfähiger Quantencomputer, der asymmetrische Kryptografie überwinden kann, existiert noch nicht, und gleichzeitig sind die klassischen Algorithmen zwar noch nicht kompromittiert, aber PQC-Algorithmen sind bereits als schwach oder gebrochen eingestuft worden. Ein

weiteres Beispiel ist der Fall des Rainbow-Algorithmus.[75] Dieser Algorithmus war einer der NIST-Finalisten für Signaturalgorithmen. Kurz nach seiner Ernennung als Finalist wurde jedoch bekannt, dass der Rainbow erfolgreich gebrochen wurde. Dies zeigt, dass es fatal wäre sich nur auf PQC-Algorithmen zu verlassen, die kontinuierliche Überprüfung und Verbesserung erfordern. Dieses Beispiel allein verdeutlicht, dass hybride Methoden durchaus sinnvoll sind.

Obwohl nicht detailliert erläutert wurde, was unter „unnötigem Ballast und der Steigerung der Komplexität von Prozessen" (Blumenthal, 2021) zu verstehen ist, untersucht diese Arbeit genau diese Aspekte. Dabei wurde hinterfragt, inwieweit Aspekte wie Speicherverbrauch oder Leistung tatsächlich als „Ballast" angesehen werden sollten. Bezüglich des Ressourcenverbrauchs wurde festgestellt, dass sowohl bei hybriden als auch bei reinen PQC-Algorithmen eine Zunahme des Ressourcenbedarfs unvermeidlich ist, gleichzeitig der Unterschied zwischen hybriden und reinen PQC-Algorithmen nicht besonders groß ist. Dies liegt daran, dass klassische Algorithmen insbesondere in Bezug auf den Speicherverbrauch nur einen minimalen zusätzlichen Overhead aufweisen. Somit ist der zusätzliche Ressourcenbedarf, der durch die Integration klassischer Algorithmen in hybride Systeme entsteht, vergleichsweise gering. Diese Redundanz in der Sicherheitsarchitektur kann als eine Absicherung gegen das Unbekannte betrachtet werden, insbesondere da der Zeitpunkt der tatsächlichen Verwendung von Quantencomputer noch ungewiss ist.

In Szenarien, in denen die Geschwindigkeit des Handshakes entscheidend ist, könnten reine PQC-Algorithmen bevorzugt werden. Allerdings sollten die Sicherheitsanforderungen und potenzielle zukünftige Bedrohungen durch Quantencomputer beachtet werden, was den Einsatz von hybriden Algorithmen rechtfertigen könnte.

Ein wichtiger Befund ist, dass die Leistung hybrider Algorithmen stark von der Wahl des klassischen Algorithmus abhängt. Hohe Leistungen wurden bei Kombinationen mit X448 und X25519 beobachtet, was darauf hindeutet, dass andere, nicht getestete Kombinationen möglicherweise noch bessere Ergebnisse für hybride Varianten liefern könnten.

[75] (Beullens, 2022)

In der Gesamtbetrachtung zeigten sich gitterbasierte Algorithmen, insbesondere Kyber, als die leistungsfähigsten. Diese Algorithmen wiesen kaum bis keine Leistungseinbußen bei steigenden Sicherheitsstufen auf. Lediglich die hybride Variante, speziell Kyber1024, verzeichnete signifikantere Leistungseinbußen, bot jedoch immer noch die kürzesten TLS-Handshake-Zeiten im Vergleich zu anderen hybriden Algorithmen desselben Sicherheitsniveaus. Die Messung hat ebenfalls gezeigt, dass andere untersuchte Algorithmen, wie BIKE oder HQC, in Bezug auf Leistungsfähigkeit gut mit klassischen Algorithmen konkurrieren können, und zwar sowohl in ihrer reinen als auch in ihrer hybriden Form. Diese Ergebnisse deuten darauf hin, dass sowohl BIKE als auch HQC, wenn sie in Kombination mit klassischen kryptografischen Methoden eingesetzt werden, effektiv funktionieren können. Sie bieten somit vielversprechende Alternativen im Bereich der PQC, sowohl im Kontext der aktuellen Sicherheitsanforderungen als auch in Vorbereitung auf die Ära der Quantencomputer.

Die ursprüngliche Frage, ob der Einsatz hybrider Algorithmen vorteilhaft ist oder ob ein direkter Übergang zu PQC sinnvoller wäre, lässt sich angesichts dieser Ergebnisse leichter beantworten. Aufgrund der in dieser Arbeit ermittelten Daten sollte die Implementierung hybrider Varianten keine wesentlichen Herausforderungen darstellen. Sowohl der Speicherbedarf als auch die zeitliche Leistungsfähigkeit sind vergleichsweise gut handhabbar. Für die meisten Anwendungsfälle sind Algorithmen der Sicherheitsstufen 1 und 3 ausreichend. Diese Stufen dürften für den Großteil der Anwendungen vollkommen genügen. Höhere Sicherheitsstufen, die in hybrider Variante schlechter performen, finden in den meisten Fällen keine Verwendung, da selbst klassische Algorithmen wie P521 erhebliche Leistungsprobleme aufweisen.[76] Zudem wurde 2017 in nur 2% der Fälle P521 verwendet.[77] Aktuellere Daten liegen nicht vor, aber es ist anzunehmen, dass diese Zahl auch gegenwärtig nicht wesentlich höher ist.

Wenn der erhöhte Zeit- und Speicherbedarf akzeptiert wird, ergibt sich aus hybriden Systemen ein erheblicher Mehrwert, der viele der aktuellen Bedenken ausräumen kann. Die Frage, ob Shors Algorithmus jemals praktisch eingesetzt werden kann oder ob Quantencomputer, die ihn nutzen können, existieren werden, ist für die Entscheidung über den Nutzen hybrider Systeme

[76] (Hannes Tschofenig, 2015)

[77] (Holmes, 2018)

nicht ausschlaggebend. Vielmehr sollte die Frage gestellt werden, was passiert, wenn PQC versagt. Diese Frage ist noch relativ neu und wenig erforscht.

Obwohl Techniken wie gitterbasierte und codebasierte Systeme bereits seit einiger Zeit bekannt sind, spielt die Art und Weise ihrer Implementierung eine wesentliche Rolle für ihre Zuverlässigkeit und Sicherheit. Die Implementierungen dieser Algorithmen sind oft noch nicht ausreichend getestet und können, wie das Beispiel des Rainbow-Algorithmus zeigt, fehleranfällig sein. Angesichts dieses Risikos sollten hybride Ansätze zumindest gründlich diskutiert werden, statt sie wegen behördlicher Vorgaben vorschnell zu verwerfen. Der potenzielle Nutzen eines hybriden Systems, das traditionelle und post-quantum-kryptografische Methoden kombiniert, könnte entscheidend sein, um die Sicherheit in einer unsicheren Zukunft der Quantencomputer zu gewährleisten.

Nichtdestotrotz müssen Trade-offs zwischen Speichergröße, Handshake-Latenz und langfristiger Sicherheit sorgfältig abgewogen werden. Die Entscheidung zwischen PQC und hybriden Methoden könnte trotzdem von spezifischen Anforderungen und dem Kontext abhängen. Während PQC-Methoden eher für Effizienz und vergleichsweise weniger Komplexität stehen, bieten hybride Methoden höhere Sicherheit.

7 Fazit und Ausblick

Durch die vergleichenden Tests, Messung nach Kriterien und Darstellung der Fachdiskussion und der messmethodischen Risiken ist die Effektivität und Praktikabilität hybrider Kryptografie-Algorithmen als einer Kombination aus klassischen und PQC-Algorithmen eingehend beschrieben. Im Ergebnis ist trotz der geäußerten Bedenken hinsichtlich des "unnötigen Ballasts" und der gesteigerten Komplexität, die hybride Lösungen mit sich bringen können aufgezeigt, dass hybride Methoden durchaus sinnvoll sind und sich in vielen Kontexten als vorteilhaft erweisen.

1 Eine voreilige Anwendung von ausschließlich PQC-Algorithmen könnte gravierende Sicherheitsrisiken für das Internet mit sich bringen.

2 PQC-Technologien sind derzeit noch nicht ausreichend stabil und getestet, um mit Sicherheit sagen zu können, dass sie den vielfältigen und sich ständig weiterentwickelnden Bedrohungen standhalten können.

3 Dies unterstreicht die Notwendigkeit einer gründlichen Überprüfung und kontinuierlichen Weiterentwicklung solcher kryptografischen Systeme, um potenzielle Schwächen zu identifizieren und zu beheben, bevor sie in breiterem Umfang eingesetzt werden.

4 Solange dieser Prozess andauert, bieten hybride Methoden, angesichts der rasanten Entwicklungen in der Quantentechnologie, eine effektive Lösung, um den Übergang in diese neue Ära sicher zu gestalten. Diese ermöglichen es, von den Stärken beider Technologien zu profitieren und gleichzeitig potenzielle Schwächen abzumildern. Sie stellen eine wichtige Brücke dar, um die Sicherheitslücke zu schließen, die sich aus dem Fortschritt der Quantentechnologie ergeben könnte, und tragen dazu bei, den Übergang zu einer quantensicheren Zukunft zu meistern.

7.1 Ausblick

Der Standardisierungsprozess des NISTs für PQC soll erst im Jahr 2024 abgeschlossen werden, gleichzeitig werden die weiteren Standards für die KEMs bekannt gegeben. Während dieser Phase könnten neue Schwachstellen in den getesteten Algorithmen aufgedeckt werden, was dazu führen könnte, dass NIST einen zuvor ausgeschiedenen Kandidaten wieder in Betracht zieht.

Ein weiteres potenzielles Einsatzgebiet für quantenresistente Algorithmen ist der Bereich der Blockchain und Kryptowährungen. Hierbei stellt sich die Frage nach der Sinnhaftigkeit hybrider Ansätze. Zwar setzen bereits Blockchains auf quantenresistente Algorithmen, aber deren Anfälligkeit für Fehler bleibt eine offene Frage nach dem Risiko für Sicherheit. Die Erforschung der Möglichkeiten und Auswirkungen hybrider Ansätze in diesem Bereich bleibt daher ein interessantes Forschungsgebiet.

Schließlich ist auch die Frage der Effizienz und Sinnhaftigkeit hybrider Signaturalgorithmen noch nicht abschließend geklärt. Dies bietet ein weiteres Forschungsfeld, um zu untersuchen, inwieweit diese Algorithmen effizient sind und ob der Einsatz hybrider Methoden vorteilhaft sein könnte.

Literaturverzeichnis

[1] Adarsh Kumar, C. O. S. S. G. R. B., 2022. *Securing the Future Internet of Things with Post-Quantum Cryptography*, s.l.: s.n.

[2] Ajtai, M., 1996. *Generating Hard Instances of the Short Basis,* s.l.: s.n.

[3] Albrecht Beutelspacher, H. B. N. T. S., 2010. Der diskrete Logarithmus, Diffie-Hellman-Schlüsselvereinbarung, ElGamal-Systeme. In: *Kryptografie in Theorie und Praxis* . s.l.:s.n., pp. 132-144.

[4] Alison Becker, R. G. D. N., 2021. *HYBRID DESIGNS.* [Online]
Available at: https://datatracker.ietf.org/meeting/112/materials/slides-112-lamps-hybrid-non-composite-multi-certificate-00
[Zugriff am 24 10 2023].

[5] Bas Westerbaan, C. D. R., 2022. *Defending against future threats: Cloudflare goes post-quantum.* [Online]
Available at: https://blog.cloudflare.com/post-quantum-for-all/
[Zugriff am 28 10 2023].

[6] Becker, A., 2021. *IETF.* [Online]
Available at:
https://mailarchive.ietf.org/arch/msg/spasm/McksDhejGgJJ6xG617FEWLbBq0k/
[Zugriff am 17 09 2023].

[7] Bernstein, D. J., 2021. *IETF.* [Online]
Available at: https://mailarchive.ietf.org/arch/msg/cfrg/T3XgKeJr4-PvmPrS5TwVNfW9t_w/
[Zugriff am 13 09 2023].

[8] Beullens, W., 2022. *Breaking Rainbow Takes a Weekend on a Laptop,* s.l.: s.n.

[9] Blumenthal, U., 2021. *IETF.* [Online]
Available at: https://mailarchive.ietf.org/arch/msg/cfrg/Kp_xKCCD6pPbAy2-lH1ygOMq4Y4/
[Zugriff am 05 11 2023].

[10] Bodo Moeller, N. B. ,. V. G. ,. S. B.-W. ,. C. H., 2006. *Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS),* s.l.: s.n.

[11] Brown, D., 2021. *IETF.* [Online]
Available at: https://mailarchive.ietf.org/arch/msg/cfrg/il8o1k2e-mQD-J2EB2BxOkolsXA/
[Zugriff am 18 09 2023].

[12] Brown, D. R. L., 2008. *Breaking RSA May Be As Difficult As Factoring.* [Online]
Available at: https://ia.cr/2005/380
[Zugriff am 22 10 2023].

[13] Carlos Aguilar Melchor, N. A. B. B. B. B.-C. D. D.-M. R., 2023. *Hamming Quasi-Cyclic,* s.l.: s.n.

[14] Craig Gidney, M. E., 2021. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum Journal,* Band 5, p. 433.

[15] curl, 2023. *Write out.* [Online]
Available at: https://everything.curl.dev/usingcurl/verbose/writeout
[Zugriff am 02 11 2023].

[16] D. Stebila, S. F. G. H., 2023. *Hybrid key exchange in TLS 1.3.* [Online]
Available at: https://datatracker.ietf.org/doc/html/draft-ietf-tls-hybrid-design-09
[Zugriff am 30 10 2023].

[17] Daniel J. Bernstein, A. H. K. S. R. N., 2019. *The SPHINCS+ Signature Framework.* [Online]

Available at: https://sphincs.org/data/sphincs+-paper.pdf
[Zugriff am 21 10 2023].

[18] Daniel J. Bernstein, B. B. B. M.-S. C. N. T., 2021. *OpenSSLNTRU: Faster post-quantum TLS key exchange,* s.l.: s.n.

[19] Daniele Micciancio, O. R., 2009. Lattice-based Cryptography. In: *Post-Quantum Cryptography.* s.l.:s.n., pp. 147-187.

[20] Diana Ghinea, F. K. J. P. J. C. S. K. R. M. J.-M. P. L. I. E. B., 2022. *Hybrid Post-Quantum Signatures in Hardware Security Keys,* s.l.: s.n.

[21] Dimitrios Sikeridis, P. K. D., 2020. *Post-Quantum Authentication in TLS 1.3,* s.l.: s.n.

[22] Elaine Barker, L. C. R. D., 2020. *NIST Special Publication 800-56C.* [Online]
Available at: https://doi.org/10.6028/NIST.SP.800-56Cr2
[Zugriff am 11 11 2023].

[23] Goutam Tamvada, S. C., 2022. *Deep dive into a post-quantum key encapsulation algorithm.* [Online]
Available at: https://blog.cloudflare.com/post-quantum-key-encapsulation/
[Zugriff am 15 10 2023].

[24] Hannes Tschofenig, M. P.-G., 2015. *Performance Investigations.* [Online]
Available at: https://www.ietf.org/proceedings/92/slides/slides-92-lwig-3.pdf
[Zugriff am 11 12 2023].

[25] Haoyu Li, R. L. Z. L. Y. P. X., 2021. *Ciphertext-Only Attacks Against Compact-LWE Submitted to NIST PQC Project.* [Online]
Available at: https://link.springer.com/article/10.1007/s11424-021-0042-3

[26] Hartnett, K., 2019. *A New Law to Describe Quantum Computing's Rise?.* [Online]
Available at: https://www.quantamagazine.org/does-nevens-law-describe-quantum-

computings-rise-20190618/
[Zugriff am 22 10 2023].

[27] Holmes, D., 2018. *THE 2017 TLS TELEMETRY REPORT.* [Online]
Available at: https://www.f5.com/pdf/products/2017_TLS_Telemetry_Report.pdf
[Zugriff am 25 10 2023].

[28] Jang, K.-B. a. S. G.-J. a. K. H.-J. a. S. H.-J., 2021. Grover on Simplified AES. *2021 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia),* pp. 1-4.

[29] Jintai Ding, B.-Y. Y., 2009. Multivariate Public Key Cryptography. In: *Post-Quantum Cryptography.* s.l.:s.n., pp. 193-234.

[30] Kyungbae Jang, A. B. ,. K. S. S. A. C., 2022. Quantum Analysis of AES.

[31] liboqs, 2017. *OPEN QUANTUM SAFE.* [Online]
Available at: https://openquantumsafe.org/liboqs/
[Zugriff am 10 11 2023].

[32] Lyubashevsky, V., 2019. *Official Comments (Round 2) - qTESLA.* [Online]
Available at: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/official-comments/qTESLA-round2-official-comment.pdf

[33] Marre, J., 2018. *Der Grover-Algorithmus und die Suche nach dem heiligen Gral.* [Online]
Available at: https://www.quantencomputer-info.de/quantencomputer/grover-algorithmus/
[Zugriff am 23 10 2023].

[34] McEliece, 2023. *Papers McEliece.* [Online]
Available at: https://classic.mceliece.org/papers.html
[Zugriff am 07 11 2023].

[35] Mike Ounsworth, J. G. M. P. J. K., 2023. *Composite Public and Private Keys For Use In Internet PKI.* [Online]
Available at: https://datatracker.ietf.org/doc/draft-ounsworth-pq-composite-keys/
[Zugriff am 29 10 2023].

[36] Mila Anastasova, P. K. J. M., 2022. *PQ-HPKE: Post-Quantum Hybrid Public Key Encryption,* s.l.: s.n.

[37] Mohamed Taoufiq Damir, T. M. S. R. a. V. N., 2022. *A Beyond-5G Authentication and Key Agreement Protocol.* [Online]
Available at: https://arxiv.org/pdf/2207.06144.pdf

[38] Nakov, S., 2021. *Elliptic Curve Cryptography (ECC).* [Online]
Available at: https://cryptobook.nakov.com/asymmetric-key-ciphers/elliptic-curve-cryptography-ecc
[Zugriff am 29 10 2023].

[39] Nicolas Aragon, T. G. L. B. A. M. B. M. P. B. S. R.-B. G. V. Z., 2022. *BIKE:Bit Flipping Key Encapsulation,* s.l.: s.n.

[40] NIST, 2016. *Submission Requirements and Evaluation Criteria.* [Online]
Available at: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf
[Zugriff am 21 10 2023].

[41] NIST, 2017. *Round 1 Submissions.* [Online]
Available at: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions
[Zugriff am 21 10 2023].

[42] NIST, 2022. *NIST Announces First Four Quantum-Resistant Cryptographic Algorithms.* [Online]
Available at: https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms
[Zugriff am 16 10 2023].

[43] NIST, 2022. *Round 4 Submissions.* [Online]
Available at: https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions
[Zugriff am 22 10 2023].

[44] NIST, 2022. *SIKE Foreword and Postscript.* [Online]
Available at: https://csrc.nist.gov/csrc/media/Projects/post-quantum-cryptography/documents/round-4/submissions/sike-team-note-insecure.pdf
[Zugriff am 2023].

[45] NIST, 2023. *NIST PQC.* [Online]
Available at: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization
[Zugriff am 23 10 2023].

[46] OQS, 2017. *OPEN QUANTUM SAFE.* [Online]
Available at: https://openquantumsafe.org/
[Zugriff am 01 11 2023].

[47] OQS, 2023. *Open Quantum Safe interop test server for quantum-safe cryptography.* [Online]
Available at: https://test.openquantumsafe.org/
[Zugriff am 10 11 2023].

[48] Panos Kampanakis, P. P. E. D. D. V. G., 2018. *The Viability of Post-Quantum X.509 Certificates.* [Online]
Available at: https://eprint.iacr.org/2018/063.pdf

[49] Peter Schwabe, D. S. T. W., 2020. *Post-quantum TLS without handshake signatures,* s.l.: s.n.

[50] Raphael Overbeck, N. S., 2009. Code-based cryptography. In: *Post-Quantum Cryptography.* s.l.:s.n., pp. 95-145.

[51] Regev, O., 2006. *Lattice-Based Cryptography.* [Online] Available at: https://link.springer.com/content/pdf/10.1007/11818175_8.pdf [Zugriff am 21 10 2023].

[52] Rescorla, E., 2018. *The Transport Layer Security (TLS) Protocol Version 1.3*, s.l.: Internet Engineering Task Force (IETF).

[53] Roberto Avanzi, J. B. L. D. E. K. T. L. V. L. J. M. S. P. S. G. S. D. S., 2021. *CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation,* s.l.: s.n.

[54] Sendrier, N., 2017. Code-Based Cryptography: State of the Art and Perspectives. *IEEE Security & Privacy,* Band 15, pp. 44-50.

[55] Shor, P. W., 1995. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.

[56] Steinfeld, R., 2018. *Official Comments - CFPKM.* [Online] Available at: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/official-comments/CFPKM-official-comment.pdf

[57] Surbhi Shaw, R. D., 2022. Post-quantum secure identity-based signature achieving forward secrecy. *Journal of Information Security and Applications,* Band 69.

[58] T. Dierks, E. R., 2008. *The Transport Layer Security (TLS) Protocol,* s.l.: Network Working Group.

[59] Vasileios Mavroeidis, K. V. M. D. Z. A. J., 2018. The Impact of Quantum Computing on Present. *International Journal of Advanced Computer Science and Applications,* pp. 405-414.

[60] Vlatko Vedral, M. B. P., 1998. Basics of Quantum Computation. *Prog.Quant.Electron.*

[61] Wouter Castryck, T. D., 2022. *An efficient key recovery attack on SIDH,* s.l.: s.n.

[62] Xavier Bonnetain, M. N.-P. A. S., 2019. Quantum Security Analysis of AES. *Transactions on Symmetric Cryptology*, pp. 55-93.

[63] Yoder, N., 2021. *Moore's Law of Moore's Law of Quantum Computing.* [Online]
Available at: https://nickyoder.com/moores-law-quantum-computer/ [Zugriff am 22 10 2023].

[64] Yu Li, L.-P. W., 2023. *Security analysis of the Classic McEliece, HQC and BIKE schemes in low memory*, s.l.: s.n.

[65] Zhengping Jay Luo, R. L. A. M., 2023. *Understanding the RSA algorithm.* [Online]
Available at: https://doi.org/10.48550/arXiv.2308.02785 [Zugriff am 22 10 2023].

A Python Codes

A.1 Lokale Tests

A.1.1 local.py

```
import subprocess
import csv
import re

SIG_ALG = ["ed25519", "ed448", "dilithium3", "p384_dilithium3"]
KEM_ALG = [
    "kyber512", "kyber768", "kyber1024", "p521_kyber1024",
"p256_kyber512",
    "x25519_kyber512", "p256_kyber768", "p384_kyber768",
"x25519_kyber768",
    "x448_kyber768", "bikel1", "bikel3", "bikel5", "p256_bikel1",
"x25519_bikel1",
    "p384_bikel3", "x448_bikel3", "p521_bikel5", "hqc128",
"hqc192", "hqc256",
    "p256_hqc128", "x25519_hqc128", "p384_hqc192", "x448_hqc192",
"p521_hqc256", "X25519", "x448"
]

def extract_avg_connections(output):
    match = re.search(r'(\d+\.\d+)\s+connections/user sec', out-
put)
    if match:
        return match.group(1)
    return None

for sig in SIG_ALG:
    results = []

    for kem in KEM_ALG:
        command = [
            "docker", "run",
            "-e", f"TEST_TIME=2",
            "-e", f"KEM_ALG={kem}",
            "-e", f"SIG_ALG={sig}",
            "-it", "oqs-curl",
```

```
            "perftest.sh"
        ]


        # Execute the command
        print(f"Running with SIG_ALG={sig} and KEM_ALG={kem}")
        result = subprocess.run(command, capture_output=True,
text=True)

        # Extract average connections per second
        avg_connections = extract_avg_connections(result.stdout)
        if avg_connections:
            results.append((kem, avg_connections))
        else:
            print(f"Error for SIG_ALG={sig} and KEM_ALG={kem}")
        print(result.stdout)
        avg_connections = extract_avg_connections(result.stdout)

    filename = f"{sig}.csv"
    with open(filename, 'a', newline='') as csvfile:
        csvwriter = csv.writer(csvfile)
        csvwriter.writerow(["KEM_ALG", "Avg_Connections_Per_Sec"])
        csvwriter.writerows(results)

    print(f"Results saved to {filename}")

print("tests completed.")
```

A.1.2 perftest.sh

```
#!/bin/sh
set -e

# Optionally set KEM to one defined in https://github.com/open-quantum-safe/oqs-provider#algorithms
if [ "x$KEM_ALG" == "x" ]; then
        export DEFAULT_GROUPS=kyber512
else
        export DEFAULT_GROUPS=$KEM_ALG
fi

# Optionally set SIG to one defined in https://github.com/open-quantum-safe/oqs-provider#algorithms
if [ "x$SIG_ALG" == "x" ]; then
        export SIG_ALG=dilithium2
fi

# Optionally set TEST_TIME
if [ "x$TEST_TIME" == "x" ]; then
        export TEST_TIME=100
fi

# Optionally set server certificate alg to one defined in https://github.com/open-quantum-safe/oqs-provider#algorithms
# The root CA's signature alg remains as set when building the image
if [ "x$SIG_ALG" != "x" ]; then
    cd /opt/oqssa/bin
    # generate new server CSR using pre-set CA.key & cert
    openssl req -new -newkey $SIG_ALG -keyout /opt/test/server.key -out /opt/test/server.csr -nodes -subj "/CN=localhost"
    if [ $? -ne 0 ]; then
       echo "Error generating keys - aborting."
       exit 1
    fi
    # generate server cert
    openssl x509 -req -in /opt/test/server.csr -out /opt/test/server.crt -CA CA.crt -CAkey CA.key -CAcreateserial -days 365
    if [ $? -ne 0 ]; then
```

```
        echo "Error generating cert - aborting."
        exit 1
    fi
fi

echo "Running $0 with SIG_ALG=$SIG_ALG and KEM_ALG=$KEM_ALG"
echo

# Start a TLS1.3 test server based on OpenSSL accepting only the specified
KEM_ALG
# The env var DEFAULT_GROUPS activates the required Group via the system
openssl.cnf:
# we put it on the command line to check for possible typos otherwise
silently discarded:
openssl s_server -cert /opt/test/server.crt -key /opt/test/server.key -
groups $DEFAULT_GROUPS -www -tls1_3 -accept localhost:4433&

# Give server time to come up first:
sleep 1

# Run handshakes for $TEST_TIME seconds
# The env var DEFAULT_GROUPS activates the required Group via the system
openssl.cnf:
openssl s_time -connect :4433 -new -time $TEST_TIME -verify 1 | grep con-
nections
```

A.2 OQS Testserver Tests

A.2.1 daily.py

```
import os
import subprocess
import json
import datetime

SIGNATURE_ALGORITHMS_TO_TEST = ["p384_dilithium3","dili-
thium3","rsa3072","esdsap256"]
CURVES_TO_TEST = ["kyber512", "kyber768",
```

```
"kyber1024","p521_kyber1024", "p256_kyber512", "x25519_kyber512",
"p256_kyber768", "p384_kyber768", "x25519_kyber768", "x448_ky-
ber768", "bikel1", "bikel3", "bikel5", "p256_bikel1", "x25519_bi-
kel1", "p384_bikel3", "x448_bikel3", "p521_bikel5", "hqc128",
"hqc192", "hqc256", "p256_hqc128", "x25519_hqc128", "p384_hqc192",
"x448_hqc192", "p521_hqc256"]

def run_daily_tests():
    print("starting the daily tests...")

    with open("assignments.json", 'r') as file:
        assignments = json.load(file)

    results_dict = {}

    for signature_algorithm, curve_to_port in assignments.items():
        if SIGNATURE_ALGORITHMS_TO_TEST and signature_algorithm
not in SIGNATURE_ALGORITHMS_TO_TEST:
            continue
        print(f"testing signature algorithm: {signature_algo-
rithm}")
        for curve, port in curve_to_port.items():
            if CURVES_TO_TEST and curve not in CURVES_TO_TEST:
                continue
            print(f"testing curve: {curve} on port: {port}")
            try:
                cmd = ["python3", "test60.py", str(port), curve]
# befehl in list splitten
                output = subprocess.check_output(cmd).decode('utf-
8')
                print(output)  # output
                # durshcnitt berechnen
                for line in output.split("\n"):
                    if "Average TLS handshake time" in line:
                        average_time = float(line.split(":")[-
1].split("ms")[0].strip())
                        if signature_algorithm not in re-
sults_dict:
                            results_dict[signature_algorithm] = []
                        results_dict[signature_algorithm].ap-
pend((datetime.datetime.now().strftime('%Y-%m-%d'), curve, aver-
age_time))
            except Exception as e:
                print(f"Error for curve: {curve} on port: {port}.
Error: {e}")

    for signature_algorithm, results in results_dict.items():
        with open(f"{signature_algorithm}.csv", "a") as file:
            if not os.path.exists(f"{signature_algorithm}.csv") or
os.stat(f"{signature_algorithm}.csv").st_size == 0:
                file.write("Date,Curve,Average Time (ms)\n")
            for result in results:
```

```
                file.write(",".join(map(str, result)) + "\n")

    print("daily tests finish")

if __name__ == "__main__":
    run_daily_tests()
```

A.2.2 test60.py

```
import subprocess
import shlex
import argparse
import csv
import os
import json
import time
from datetime import datetime

with open("assignments.json", 'r') as file:
    assignments = json.load(file)

def measure_tls_handshake_with_docker_curl(port, curve,
ca_cert_path="/home/oqs/test"):
    url = f"https://test.openquantumsafe.org:{port}"

    # o time_appconnect
    cmd = (f'curl --cacert /ca/CA.crt {url} --curves {curve} '
           f'-o /dev/null -s -w "%{{time_connect}},%{{time_appcon-
nect}}" -I')

    total_tls_handshake_time = 0
    num_successes = 0
    # docker in detached mode starten sonst zu viele container
    container_id = subprocess.check_output(["docker", "run", "-d",
"--rm", "-v", ca_cert_path + ":/ca", "oqs-curl", "tail", "-f",
"/dev/null"]).decode().strip()

    for i in range(1, 6):
        try:
            output = subprocess.check_output(["docker", "exec",
container_id] + shlex.split(cmd)).decode('utf-8')
            time_connect, time_appconnect = map(float, out-
put.split(','))
            tls_handshake_time = (time_appconnect - time_connect)
* 1000
            total_tls_handshake_time += tls_handshake_time
            num_successes += 1
            print(f"Iteration {i}: TLS Handshake time for {url}
```

```
using curve {curve}: {tls_handshake_time:.2f} ms")

            time.sleep(3)  # delay 3 sekudnen
        except subprocess.CalledProcessError as e:
            print(f"Iteration {i}: Error docker: {e}")

    subprocess.call(["docker", "stop", container_id])

    if num_successes > 0:
        average_tls_handshake_time = total_tls_handshake_time /
num_successes

        signature_algorithm = None
        for algo, ports in assignments.items():
            if port in ports.values():
                signature_algorithm = algo
                break

        directory_path = f"./{signature_algorithm}"
        if not os.path.exists(directory_path):
            os.makedirs(directory_path)

        filename = f"{directory_path}/{curve}.csv"

        with open(filename, 'a', newline='') as csvfile:
            fieldnames = ['date', 'average_handshake_time']
            writer = csv.DictWriter(csvfile, fieldnames=field-
names)

            if csvfile.tell() == 0:
                writer.writeheader()

            writer.writerow({
                'date': datetime.now().strftime('%Y-%m-%d'),
                'average_handshake_time': average_tls_hand-
shake_time
            })
        print(f"Average TLS Handshake time for {url} using
{curve} over {num_successes} times: {average_tls_hand-
shake_time:.2f} ms")
        print(f"output saved to {filename}")
    else:
        print(f"No  handshakes for {url} using curve {curve}.")

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Measure Average
TLS Handshake with Dockerized OQS curl over 10 trials and save re-
sults to CSV")
    parser.add_argument("port", type=int, help="Port number of the
target server.")
    parser.add_argument("curve", type=str, help="Curve to use for
the TLS handshake.")
```

```
    args = parser.parse_args()
    measure_tls_handshake_with_docker_curl(args.port, args.curve)
```

B Ergebnisse

B.1 Lokale Tests

B.1.1 Dilithium3.csv

KEM_ALG	*Avg_Connections_Per_Sec*
kyber512	2390.24
kyber768	2352.42
kyber1024	2430.17
p521_kyber1024	109.52
p256_kyber512	949.59
x25519_kyber512	2000.93
p256_kyber768	976.3
p384_kyber768	260.19
x25519_kyber768	1794.44
x448_kyber768	974.8
bikel1	510.17
bikel3	182.86
bikel5	76.47
p256_bikel1	386.84
x25519_bikel1	481.37
p384_bikel3	101.53
x448_bikel3	167.65
p521_bikel5	43.79
hqc128	1619.35
hqc192	1314.73
hqc256	748.46

p256_hqc128	895.65
x25519_hqc128	1005.08
p384_hqc192	241.23
x448_hqc192	668.5
p521_hqc256	127.93
X25519	1814.17
x448	1130.17
kyber512	2152.55
kyber768	2227.87
kyber1024	1998.35
p521_kyber1024	118.69
p256_kyber512	970.9
x25519_kyber512	1835.25
p256_kyber768	896.3
p384_kyber768	298.17
x25519_kyber768	1818.6
x448_kyber768	1005.56
bikel1	526.67
bikel3	184.71
bikel5	72.96
p256_bikel1	357.52
x25519_bikel1	497.01
p384_bikel3	102.86
x448_bikel3	165.43
p521_bikel5	48.57
hqc128	1784.66
hqc192	1371.03
hqc256	849.31
p256_hqc128	830.34
x25519_hqc128	1374.26
p384_hqc192	237.6
x448_hqc192	738.02
p521_hqc256	97.32
X25519	2092.44
x448	1319.53
kyber512	2390
kyber768	2469.37
kyber1024	2347.22
p521_kyber1024	108.2
p256_kyber512	993.5
x25519_kyber512	2029.46

p256_kyber768	978.03
p384_kyber768	268.7
x25519_kyber768	1761.94
x448_kyber768	1003.28
bikel1	514.94
bikel3	183.24
bikel5	70.83
p256_bikel1	374.66
x25519_bikel1	491.23
p384_bikel3	100
x448_bikel3	167.61
p521_bikel5	44.67
hqc128	1761.19
hqc192	1374.83
hqc256	962.14
p256_hqc128	848.28
x25519_hqc128	1436.88
p384_hqc192	240.16
x448_hqc192	719.23
p521_hqc256	103.74
X25519	2257.69
x448	1349.21
kyber512	2307.32
kyber768	2172.87
kyber1024	2436.3
p521_kyber1024	126.73
p256_kyber512	999.25
x25519_kyber512	2017.21
p256_kyber768	976.07
p384_kyber768	250
x25519_kyber768	2005.22
x448_kyber768	960.8
bikel1	530.14
bikel3	181.5
bikel5	73.75
p256_bikel1	362.82
x25519_bikel1	490.8
p384_bikel3	100.79
x448_bikel3	161.33
p521_bikel5	45.32
hqc128	1725.34

hqc192	1316.88
hqc256	866.45
p256_hqc128	830.6
x25519_hqc128	1432.61
p384_hqc192	239.37
x448_hqc192	736.43
p521_hqc256	95.12
X25519	2264.29
x448	1347.29
kyber512	2324.35
kyber768	2737.11
kyber1024	2275.89
p521_kyber1024	119.47
p256_kyber512	896.99
x25519_kyber512	1920
p256_kyber768	922.06
p384_kyber768	244.26
x25519_kyber768	1740.37
x448_kyber768	989.68
bikel1	520.93
bikel3	185.16
bikel5	78.05
p256_bikel1	368.79
x25519_bikel1	486.39
p384_bikel3	95.49
x448_bikel3	158.08
p521_bikel5	44.78
hqc128	1648.91
hqc192	1415.33
hqc256	909.46
p256_hqc128	867.63
x25519_hqc128	1519.31
p384_hqc192	236.28
x448_hqc192	766.41
p521_hqc256	114.53
X25519	2238.79
x448	1412.07
kyber512	2329.03
kyber768	2234.96
kyber1024	2572.97
p521_kyber1024	115.65

p256_kyber512	1032.09
x25519_kyber512	1908.13
p256_kyber768	968.55
p384_kyber768	266.38
x25519_kyber768	1929.75
x448_kyber768	948.67
bikel1	530.68
bikel3	182.84
bikel5	75.97
p256_bikel1	351.95
x25519_bikel1	493.94
p384_bikel3	90.32
x448_bikel3	164.94
p521_bikel5	45.45
hqc128	1752.86
hqc192	1357.64
hqc256	908.63
p256_hqc128	842.75
x25519_hqc128	1369.57
p384_hqc192	242.31
x448_hqc192	738.35
p521_hqc256	88.18
X25519	2254.42
x448	1320.8
kyber512	2468.89
kyber768	2305.41
kyber1024	2387.7
p521_kyber1024	122.52
p256_kyber512	970.15
x25519_kyber512	1844.44
p256_kyber768	979.17
p384_kyber768	266.39
x25519_kyber768	1491.37
x448_kyber768	1025.24
bikel1	535.8
bikel3	175.88
bikel5	77.85
p256_bikel1	400.7
x25519_bikel1	499.41
p384_bikel3	101.56
x448_bikel3	152.2

p521_bikel5	45.52
hqc128	1685.53
hqc192	1420.57
hqc256	936.81
p256_hqc128	817.91
x25519_hqc128	1517.36
p384_hqc192	242.74
x448_hqc192	808.8
p521_hqc256	114.91
X25519	2311.48
x448	1346.72
kyber512	2547.9
kyber768	2567.92
kyber1024	2490.83
p521_kyber1024	110.71
p256_kyber512	1012.5
x25519_kyber512	1829.37
p256_kyber768	886.36
p384_kyber768	240.8
x25519_kyber768	1721.64
x448_kyber768	885.94
bikel1	513.81
bikel3	185.38
bikel5	77.16
p256_bikel1	360.78
x25519_bikel1	494.88
p384_bikel3	105.83
x448_bikel3	157.67
p521_bikel5	47.69
hqc128	1890.84
hqc192	1480.34
hqc256	948.98
p256_hqc128	854.86
x25519_hqc128	1440.41
p384_hqc192	248
x448_hqc192	750
p521_hqc256	101.53
X25519	2240.34
x448	1272.13

B.1.2 Ed448.csv

KEM_ALG	*Avg_Connections_Per_Sec*
kyber512	1729.82
kyber768	1599.28
kyber1024	1704.67
p521_kyber1024	105.56
p256_kyber512	878.63
x25519_kyber512	1534.85
p256_kyber768	782.55
p384_kyber768	246.28
x25519_kyber768	1455.38
x448_kyber768	797.06
bikel1	471.28
bikel3	178.95
bikel5	77.3
p256_bikel1	351.27
x25519_bikel1	457.14
p384_bikel3	97.78
x448_bikel3	156.78
p521_bikel5	43.54
hqc128	1518.25
hqc192	1133.95
hqc256	856.25
p256_hqc128	758.62
x25519_hqc128	1203.42
p384_hqc192	245.08
x448_hqc192	681.43
p521_hqc256	103.48
X25519	1688.62
x448	1080.73
kyber512	1830.4
kyber768	1751.52
kyber1024	1611.68
p521_kyber1024	108.47
p256_kyber512	872.66
x25519_kyber512	1510.29
p256_kyber768	882.73
p384_kyber768	254.55
x25519_kyber768	1407.69

x448_kyber768	866.67
bikel1	480.11
bikel3	176.84
bikel5	73.17
p256_bikel1	364.52
x25519_bikel1	452.57
p384_bikel3	97.87
x448_bikel3	152.29
p521_bikel5	44.37
hqc128	1390.07
hqc192	1110.19
hqc256	847.71
p256_hqc128	778
x25519_hqc128	1220.95
p384_hqc192	240.34
x448_hqc192	676.09
p521_hqc256	91.82
X25519	1827.43
x448	1149.04
kyber512	1739.53
kyber768	1820.33
kyber1024	1693.55
p521_kyber1024	111.67
p256_kyber512	869.66
x25519_kyber512	1535.71
p256_kyber768	858.57
p384_kyber768	227.12
x25519_kyber768	1448.2
x448_kyber768	851.82
bikel1	493.49
bikel3	180.77
bikel5	74.7
p256_bikel1	365.54
x25519_bikel1	451.45
p384_bikel3	103.7
x448_bikel3	158.02
p521_bikel5	44.93
hqc128	1542.54
hqc192	1115.82
hqc256	875.26
p256_hqc128	753.9

x25519_hqc128	1208.78
p384_hqc192	227.78
x448_hqc192	644.68
p521_hqc256	105.31
X25519	1672.5
x448	1194.96
kyber512	1733.57
kyber768	1949.18
kyber1024	1694.4
p521_kyber1024	115.45
p256_kyber512	816.79
x25519_kyber512	1433.82
p256_kyber768	849.3
p384_kyber768	258.47
x25519_kyber768	1498.76
x448_kyber768	838.24
bikel1	494.61
bikel3	175.74
bikel5	73.42
p256_bikel1	368.53
x25519_bikel1	460.82
p384_bikel3	103.79
x448_bikel3	157.69
p521_bikel5	45.7
hqc128	1316.78
hqc192	1070.2
hqc256	891.89
p256_hqc128	739.47
x25519_hqc128	1217.56
p384_hqc192	236.67
x448_hqc192	702.4
p521_hqc256	98.25
X25519	1648
x448	1198.26
kyber512	1830.16
kyber768	1758.78
kyber1024	1773.6
p521_kyber1024	104.31
p256_kyber512	848.57
x25519_kyber512	1572.93
p256_kyber768	850.72

p384_kyber768	248.41
x25519_kyber768	1460.14
x448_kyber768	858.65
bikel1	499.43
bikel3	180.9
bikel5	79.62
p256_bikel1	342.41
x25519_bikel1	441.07
p384_bikel3	100.74
x448_bikel3	161.21
p521_bikel5	46.38
hqc128	1497.67
hqc192	1073.01
hqc256	850.93
p256_hqc128	754.19
x25519_hqc128	1256.46
p384_hqc192	236.07
x448_hqc192	690.71
p521_hqc256	90.09
X25519	1647.69
x448	1148.33
kyber512	1914.05
kyber768	1733.07
kyber1024	1601.47
p521_kyber1024	120.72
p256_kyber512	856.95
x25519_kyber512	1510.95
p256_kyber768	836.3
p384_kyber768	249.61
x25519_kyber768	1398.48
x448_kyber768	871.43
bikel1	491.06
bikel3	178.53
bikel5	74.12
p256_bikel1	348.52
x25519_bikel1	473.37
p384_bikel3	98.53
x448_bikel3	161.88
p521_bikel5	44.22
hqc128	1434.97
hqc192	1117.83

hqc256	827.45
p256_hqc128	772.19
x25519_hqc128	1088.89
p384_hqc192	247.93
x448_hqc192	632.37
p521_hqc256	105.45
X25519	1778.45
x448	1119.01
kyber512	1791.94
kyber768	1703.15
kyber1024	1739.55
p521_kyber1024	111.61
p256_kyber512	843.55
x25519_kyber512	1337.69
p256_kyber768	880.71
p384_kyber768	252.03
x25519_kyber768	1443.08
x448_kyber768	852.99
bikel1	483.62
bikel3	176.14
bikel5	67.26
p256_bikel1	350.69
x25519_bikel1	480.98
p384_bikel3	92.03
x448_bikel3	156.41
p521_bikel5	43.45
hqc128	1554.61
hqc192	1113.04
hqc256	798.7
p256_hqc128	743.24
x25519_hqc128	1248.63
p384_hqc192	245.16
x448_hqc192	705.19
p521_hqc256	104.63
X25519	1770.49
x448	1120.16
kyber512	1839.53
kyber768	1882.46
kyber1024	1766.4
p521_kyber1024	111.65
p256_kyber512	914.81

x25519_kyber512	1537.8
p256_kyber768	817.02
p384_kyber768	267.54
x25519_kyber768	1358.33
x448_kyber768	910.32
bikel1	504.09
bikel3	175.45
bikel5	77.64
p256_bikel1	344.94
x25519_bikel1	467.72
p384_bikel3	94.57
x448_bikel3	161.94
p521_bikel5	45.45
hqc128	1282.47
hqc192	1174.19
hqc256	874.83
p256_hqc128	752.63
x25519_hqc128	1157.53
p384_hqc192	228.57
x448_hqc192	683.22
p521_hqc256	102.61
X25519	1740.83
x448	1130.82

B.1.3 Ed25519.csv

KEM_ALG	***Avg_Connections_Per_Sec***
kyber512	2256.25
kyber768	2254.81
kyber1024	2193.98
p521_kyber1024	118.26
p256_kyber512	962.41
x25519_kyber512	1931.45
p256_kyber768	950.36
p384_kyber768	268.7
x25519_kyber768	1681.95
x448_kyber768	947.29
bikel1	498.18
bikel3	183.72
bikel5	74.69

p256_bikel1	360.48
x25519_bikel1	477.19
p384_bikel3	111.03
x448_bikel3	166.44
p521_bikel5	46.43
hqc128	1831.62
hqc192	1374.17
hqc256	905.56
p256_hqc128	865.22
x25519_hqc128	1392.31
p384_hqc192	226.89
x448_hqc192	726.72
p521_hqc256	96.61
X25519	2137.21
x448	1257.81
kyber512	2214.38
kyber768	2336.72
kyber1024	2090.85
p521_kyber1024	113.73
p256_kyber512	916.15
x25519_kyber512	1800
p256_kyber768	921.32
p384_kyber768	268.27
x25519_kyber768	1621.28
x448_kyber768	949.61
bikel1	511.04
bikel3	181.5
bikel5	69.74
p256_bikel1	354.86
x25519_bikel1	466.46
p384_bikel3	99.25
x448_bikel3	165.1
p521_bikel5	48.94
hqc128	1682.07
hqc192	1365.77
hqc256	900
p256_hqc128	798.64
x25519_hqc128	1445
p384_hqc192	257.76
x448_hqc192	745.19
p521_hqc256	104.35

X25519	2088.62
x448	1263.03
kyber512	2217.05
kyber768	2255.26
kyber1024	2210.57
p521_kyber1024	115.52
p256_kyber512	889.71
x25519_kyber512	1786.5
p256_kyber768	927.54
p384_kyber768	257.14
x25519_kyber768	1725
x448_kyber768	922.05
bikel1	502.47
bikel3	181.82
bikel5	74.4
p256_bikel1	360
x25519_bikel1	477.3
p384_bikel3	103.7
x448_bikel3	169.08
p521_bikel5	43.33
hqc128	1658.54
hqc192	1277.02
hqc256	929.61
p256_hqc128	820.55
x25519_hqc128	1434.97
p384_hqc192	236.36
x448_hqc192	726.56
p521_hqc256	97.48
X25519	2443.1
x448	1267.52
kyber512	2220.59
kyber768	2055.03
kyber1024	2013.79
p521_kyber1024	112.15
p256_kyber512	965.67
x25519_kyber512	1801.53
p256_kyber768	929.27
p384_kyber768	250
x25519_kyber768	1658.45
x448_kyber768	929.46
bikel1	500.61

bikel3	187.13
bikel5	73.62
p256_bikel1	362.75
x25519_bikel1	486.71
p384_bikel3	99.28
x448_bikel3	159.49
p521_bikel5	45.71
hqc128	1650.93
hqc192	1274.5
hqc256	910.96
p256_hqc128	866.42
x25519_hqc128	1390.07
p384_hqc192	246.83
x448_hqc192	727.46
p521_hqc256	88.07
X25519	1932.85
x448	1351.26
kyber512	2273.29
kyber768	2193.53
kyber1024	2112.95
p521_kyber1024	131.09
p256_kyber512	967.63
x25519_kyber512	1755.15
p256_kyber768	871.21
p384_kyber768	253.28
x25519_kyber768	1674.29
x448_kyber768	945.31
bikel1	503.23
bikel3	178.82
bikel5	74.25
p256_bikel1	362.58
x25519_bikel1	463.01
p384_bikel3	108.13
x448_bikel3	158.64
p521_bikel5	45.64
hqc128	1748.1
hqc192	1341.22
hqc256	935.33
p256_hqc128	819.87
x25519_hqc128	1402.65
p384_hqc192	248.76

x448_hqc192	724.63
p521_hqc256	103.88
X25519	1876.06
x448	1329.27
kyber512	2331.9
kyber768	2300
kyber1024	2022.45
p521_kyber1024	110.53
p256_kyber512	980.15
x25519_kyber512	1713.18
p256_kyber768	951.85
p384_kyber768	238.1
x25519_kyber768	1780.49
x448_kyber768	950.4
bikel1	497.7
bikel3	184.75
bikel5	76.1
p256_bikel1	374.21
x25519_bikel1	474.55
p384_bikel3	97.06
x448_bikel3	162
p521_bikel5	46.76
hqc128	1713.29
hqc192	1256.86
hqc256	895
p256_hqc128	840.77
x25519_hqc128	1413.08
p384_hqc192	279.82
x448_hqc192	743.85
p521_hqc256	98.04
X25519	1999.26
x448	1188.46
kyber512	2185.06
kyber768	2143.7
kyber1024	2067.72
p521_kyber1024	106.09
p256_kyber512	967.91
x25519_kyber512	1599.33
p256_kyber768	985.93
p384_kyber768	266.67
x25519_kyber768	1743.9

x448_kyber768	934.21
bikel1	490.12
bikel3	183.7
bikel5	75
p256_bikel1	384.42
x25519_bikel1	466.28
p384_bikel3	95.65
x448_bikel3	163.16
p521_bikel5	43.07
hqc128	1756.05
hqc192	1391.67
hqc256	906.04
p256_hqc128	790.73
x25519_hqc128	1404.7
p384_hqc192	219.84
x448_hqc192	738.52
p521_hqc256	96.64
X25519	2133.33
x448	1282.03
kyber512	2085.71
kyber768	2126.55
kyber1024	2267.48
p521_kyber1024	107.02
p256_kyber512	939.16
x25519_kyber512	1748.92
p256_kyber768	938.52
p384_kyber768	282.57
x25519_kyber768	1816.67
x448_kyber768	942.4
bikel1	506.17
bikel3	183.62
bikel5	74.39
p256_bikel1	353.75
x25519_bikel1	473.81
p384_bikel3	95.93
x448_bikel3	163.96
p521_bikel5	46.26
hqc128	1708.86
hqc192	1307.04
hqc256	967.46
p256_hqc128	866.15

x25519_hqc128	1456.34
p384_hqc192	233.61
x448_hqc192	741.32
p521_hqc256	107.21
X25519	1989.71
x448	1267.74

B.1.4 P384_dilithium3.csv

KEM_ALG	***Avg_Connections_Per_Sec***
kyber512	699.13
kyber768	668.1
kyber1024	701.74
p521_kyber1024	123.28
p256_kyber512	519.08
x25519_kyber512	654.84
p256_kyber768	437.3
p384_kyber768	226.05
x25519_kyber768	730.58
x448_kyber768	565
bikel1	374.07
bikel3	164.61
bikel5	77.01
p256_bikel1	290.28
x25519_bikel1	353.25
p384_bikel3	90.71
x448_bikel3	147.98
p521_bikel5	43.45
hqc128	726.61
hqc192	617.97
hqc256	549.3
p256_hqc128	519.86
x25519_hqc128	675
p384_hqc192	215.38
x448_hqc192	450
p521_hqc256	81.17
X25519	760.98
x448	616.39

kyber512	822.61
kyber768	801.32
kyber1024	758.06
p521_kyber1024	103.51
p256_kyber512	539.23
x25519_kyber512	715.6
p256_kyber768	548.12
p384_kyber768	220.47
x25519_kyber768	714.78
x448_kyber768	546.62
bikel1	374.21
bikel3	163.13
bikel5	76.05
p256_bikel1	287.57
x25519_bikel1	355.8
p384_bikel3	95.56
x448_bikel3	151.1
p521_bikel5	44.3
hqc128	709.85
hqc192	673.64
hqc256	526.85
p256_hqc128	505.8
x25519_hqc128	680.31
p384_hqc192	198.5
x448_hqc192	455.78
p521_hqc256	85.48
X25519	798.26
x448	677.59
kyber512	815.38
kyber768	801.69
kyber1024	800
p521_kyber1024	100.82
p256_kyber512	554.01
x25519_kyber512	758.2
p256_kyber768	532.33
p384_kyber768	222.13
x25519_kyber768	770.34
x448_kyber768	535.88
bikel1	365.95
bikel3	161.26
bikel5	69.51

p256_bikel1	294.81
x25519_bikel1	351.4
p384_bikel3	92.03
x448_bikel3	146.37
p521_bikel5	46.15
hqc128	687.6
hqc192	634.31
hqc256	541.13
p256_hqc128	490
x25519_hqc128	650
p384_hqc192	210
x448_hqc192	466.9
p521_hqc256	95.2
X25519	834.82
x448	631.3
kyber512	871.53
kyber768	782.2
kyber1024	800.81
p521_kyber1024	97.52
p256_kyber512	553.38
x25519_kyber512	744.35
p256_kyber768	537.5
p384_kyber768	211.57
x25519_kyber768	737.7
x448_kyber768	554.76
bikel1	362.15
bikel3	162.5
bikel5	63.91
p256_bikel1	293.84
x25519_bikel1	345.95
p384_bikel3	87.94
x448_bikel3	145.05
p521_bikel5	46.9
hqc128	720.16
hqc192	631.39
hqc256	547.92
p256_hqc128	521.74
x25519_hqc128	658.12
p384_hqc192	205.22
x448_hqc192	458.22
p521_hqc256	84.55

X25519	755.37
x448	624.22
kyber512	863.46
kyber768	765.49
kyber1024	776.67
p521_kyber1024	101.72
p256_kyber512	534.62
x25519_kyber512	719.51
p256_kyber768	568.18
p384_kyber768	222.69
x25519_kyber768	699.07
x448_kyber768	513.97
bikel1	370.74
bikel3	165.76
bikel5	74.6
p256_bikel1	290.62
x25519_bikel1	351.4
p384_bikel3	96.4
x448_bikel3	148.02
p521_bikel5	44.67
hqc128	734.78
hqc192	639.71
hqc256	522.39
p256_hqc128	517.78
x25519_hqc128	661.72
p384_hqc192	219.08
x448_hqc192	461.27
p521_hqc256	101.68
X25519	721.54
x448	637.5
kyber512	802.5
kyber768	786.86
kyber1024	802.46
p521_kyber1024	94.07
p256_kyber512	536.17
x25519_kyber512	760.17
p256_kyber768	547.58
p384_kyber768	262.93
x25519_kyber768	718.75
x448_kyber768	553.49
bikel1	368.13

bikel3	171.12
bikel5	66.86
p256_bikel1	292.35
x25519_bikel1	361.24
p384_bikel3	93.66
x448_bikel3	150
p521_bikel5	42.67
hqc128	747.93
hqc192	669.14
hqc256	535.71
p256_hqc128	504.41
x25519_hqc128	652.71
p384_hqc192	200.76
x448_hqc192	452.99
p521_hqc256	84.07
X25519	757.69
x448	615.38
kyber512	795.83
kyber768	823.42
kyber1024	778.81
p521_kyber1024	110.92
p256_kyber512	532.59
x25519_kyber512	774.11
p256_kyber768	542.22
p384_kyber768	205.69
x25519_kyber768	712.4
x448_kyber768	540.46
bikel1	354.89
bikel3	161.62
bikel5	69.71
p256_bikel1	286.18
x25519_bikel1	358.1
p384_bikel3	83.08
x448_bikel3	145.51
p521_bikel5	43.41
hqc128	752.5
hqc192	628.46
hqc256	519.87
p256_hqc128	496.38
x25519_hqc128	659.38
p384_hqc192	199.24

x448_hqc192	458.09
p521_hqc256	103.39
X25519	794.92
x448	625
kyber512	842.24
kyber768	859.42
kyber1024	793.75
p521_kyber1024	106.84
p256_kyber512	517.16
x25519_kyber512	732.26
p256_kyber768	547.69
p384_kyber768	212.8
x25519_kyber768	750.41
x448_kyber768	558.39
bikel1	363.83
bikel3	165.41
bikel5	65.13
p256_bikel1	280.12
x25519_bikel1	351.18
p384_bikel3	84.78
x448_bikel3	145.56
p521_bikel5	44.78
hqc128	716.54
hqc192	640.16
hqc256	520.97
p256_hqc128	500.84
x25519_hqc128	663.16
p384_hqc192	183.33
x448_hqc192	484.89
p521_hqc256	92.62
X25519	739.67
x448	664.75

B.2 OQS Testserver Ergebnisse

B.2.1 Dilithium3.csv

Date	*Curve*	*Average Time (ms)*
18.10.2023	bikel3	60.27
18.10.2023	hqc192	73.48
18.10.2023	kyber768	43.79
18.10.2023	p384_bikel3	65.72
18.10.2023	x448_bikel3	57.12
18.10.2023	p384_hqc192	84.99
18.10.2023	x448_hqc192	79.1
18.10.2023	p384_kyber768	56.6
18.10.2023	x25519	45.03
18.10.2023	x448_kyber768	46
19.10.2023	bikel3	57.81
19.10.2023	hqc192	110.26
19.10.2023	kyber768	43.84
19.10.2023	p384_bikel3	62.61
19.10.2023	x448_bikel3	55.65
19.10.2023	p384_hqc192	83.88
19.10.2023	x448_hqc192	81.03
19.10.2023	p384_kyber768	54.58
19.10.2023	x448_kyber768	47.93
19.10.2023	x25519	71.41
20.10.2023	bikel3	58.9
20.10.2023	hqc192	76.47
20.10.2023	kyber768	47.21
20.10.2023	p384_bikel3	63.68
20.10.2023	x448_bikel3	56.72
20.10.2023	p384_hqc192	85.58
20.10.2023	x448_hqc192	77.99
20.10.2023	p384_kyber768	57.34
20.10.2023	x448_kyber768	48.47
20.10.2023	x25519	47.02
20.10.2023	bikel3	54.46
20.10.2023	hqc192	75.52
20.10.2023	kyber768	44.61
20.10.2023	p384_bikel3	62.81
20.10.2023	x448_bikel3	54.04

20.10.2023p384_hqc192	79.84
20.10.2023x448_hqc192	78.95
20.10.2023p384_kyber768	57.05
20.10.2023x448_kyber768	47.09
21.10.2023x25519	46.2
21.10.2023bikel3	62.83
21.10.2023hqc192	75.26
21.10.2023kyber768	44.75
21.10.2023p384_bikel3	61.77
21.10.2023x448_bikel3	55.81
21.10.2023p384_hqc192	82.53
21.10.2023x448_hqc192	82.82
21.10.2023p384_kyber768	58.56
21.10.2023x448_kyber768	47.69
21.10.2023x25519	49.72
22.10.2023bikel3	58.09
22.10.2023hqc192	74.48
22.10.2023kyber768	45.39
22.10.2023p384_bikel3	64.97
22.10.2023x448_bikel3	58.23
22.10.2023p384_hqc192	82.44
22.10.2023x448_hqc192	72.45
22.10.2023p384_kyber768	57.74
22.10.2023x448_kyber768	44.97
22.10.2023x25519	47.09
22.10.2023bikel3	64
22.10.2023hqc192	84.26
22.10.2023kyber768	50.95
22.10.2023p384_bikel3	65.55
22.10.2023x448_bikel3	68.41
22.10.2023p384_hqc192	93.78
22.10.2023x448_hqc192	86.58
22.10.2023p384_kyber768	61.97
22.10.2023x448_kyber768	47.78
23.10.2023x25519	51.46
23.10.2023bikel3	63.84
23.10.2023hqc192	92.43
23.10.2023kyber768	45.36
23.10.2023p384_bikel3	68.36
23.10.2023x448_bikel3	61.05
23.10.2023p384_hqc192	102.44

23.10.2023x448_hqc192	90.77
23.10.2023p384_kyber768	62.86
23.10.2023x448_kyber768	55.65
23.10.2023x25519	45.99
23.10.2023bikel3	53.37
23.10.2023hqc192	74.6
23.10.2023kyber768	41.93
23.10.2023p384_bikel3	61.14
23.10.2023x448_bikel3	56.15
23.10.2023p384_hqc192	83.22
23.10.2023x448_hqc192	74.62
23.10.2023p384_kyber768	55.87
23.10.2023x448_kyber768	46.12
23.10.2023x25519	59.18
24.10.2023bikel3	59.9
24.10.2023hqc192	72.76
24.10.2023kyber768	46.38
24.10.2023p384_bikel3	63.5
24.10.2023x448_bikel3	58.41
24.10.2023p384_hqc192	82.31
24.10.2023x448_hqc192	77.78
24.10.2023p384_kyber768	58.86
24.10.2023x448_kyber768	46.09
24.10.2023x25519	46.73
25.10.2023bikel3	59.26
25.10.2023hqc192	76.73
25.10.2023kyber768	46.22
25.10.2023p384_bikel3	61.66
25.10.2023x448_bikel3	56.33
25.10.2023p384_hqc192	85.64
25.10.2023x448_hqc192	76.55
25.10.2023p384_kyber768	56.15
25.10.2023x448_kyber768	47.98
25.10.2023x25519	46.61
25.10.2023bikel3	77.1
25.10.2023hqc192	86.17
25.10.2023kyber768	49.02
25.10.2023p384_bikel3	63.29
25.10.2023x448_bikel3	57.01
25.10.2023p384_hqc192	89.36
25.10.2023x448_hqc192	81.5

25.10.2023p384_kyber768	56.21
25.10.2023x448_kyber768	48.9
25.10.2023x25519	50.8
26.10.2023bikel3	56.9
26.10.2023hqc192	72.27
26.10.2023kyber768	42.83
26.10.2023p384_bikel3	61.86
26.10.2023x448_bikel3	132.06
26.10.2023p384_hqc192	79.66
26.10.2023x448_hqc192	87.99
26.10.2023p384_kyber768	55.62
26.10.2023x448_kyber768	44.89
26.10.2023x25519	45.05
29.10.2023bikel3	74.54
29.10.2023hqc192	73.33
29.10.2023kyber768	43.81
29.10.2023p384_bikel3	80.1
29.10.2023x448_bikel3	76.62
29.10.2023p384_hqc192	78.81
29.10.2023x448_hqc192	74.25
29.10.2023p384_kyber768	62.37
29.10.2023x448_kyber768	45.45
29.10.2023x25519	48.24
29.10.2023bikel3	106.87
29.10.2023hqc192	72.85
29.10.2023kyber768	62.18
29.10.2023p384_bikel3	84.78
29.10.2023x448_bikel3	113.23
29.10.2023p384_hqc192	84.17
29.10.2023x448_hqc192	94.87
29.10.2023p384_kyber768	53.21
29.10.2023x448_kyber768	45.52
29.10.2023x25519	58.04
30.10.2023bikel3	76.56
30.10.2023hqc192	74.23
30.10.2023kyber768	42.78
30.10.2023p384_bikel3	84.67
30.10.2023x448_bikel3	77
30.10.2023p384_hqc192	80.44
30.10.2023x448_hqc192	76.89
30.10.2023p384_kyber768	59.2

30.10.2023x448_kyber768	45.02
30.10.2023x25519	72.01

B.2.2 Rsa3072.csv

Date	*Curve*	*Average Time (ms)*
18.10.2023	bikel1	51.98
18.10.2023	bikel3	57.28
18.10.2023	bikel5	65.14
18.10.2023	hqc128	46.1
18.10.2023	hqc192	45.37
18.10.2023	hqc256	93.62
18.10.2023	kyber1024	49.52
18.10.2023	kyber512	46.68
18.10.2023	kyber768	45.22
18.10.2023	p256_bikel1	56.46
18.10.2023	x25519_bikel1	52.39
18.10.2023	p384_bikel3	64.21
18.10.2023	x448_bikel3	56.64
18.10.2023	p521_bikel5	80.02
18.10.2023	p256_hqc128	51.31
18.10.2023	x25519_hqc128	47.8
18.10.2023	p384_hqc192	58.78
18.10.2023	x448_hqc192	47.89
18.10.2023	p521_hqc256	96.16
18.10.2023	p521_kyber1024	73.41
18.10.2023	p256_kyber512	54.21
18.10.2023	x25519_kyber512	49.75
18.10.2023	p256_kyber768	51.04
18.10.2023	p384_kyber768	60.81
18.10.2023	x25519_kyber768	46.21
18.10.2023	x448_kyber768	46.87
18.10.2023	x25519	52.12
19.10.2023	bikel1	52.06
19.10.2023	bikel3	57.12
19.10.2023	bikel5	66.12
19.10.2023	hqc128	45.47
19.10.2023	hqc192	47.73

19.10.2023hqc256	121.14
19.10.2023kyber1024	47.56
19.10.2023kyber512	44.82
19.10.2023kyber768	45
19.10.2023p256_bikel1	53.8
19.10.2023x25519_bikel1	50.98
19.10.2023p384_bikel3	62.14
19.10.2023x448_bikel3	130.56
19.10.2023p521_bikel5	78.86
19.10.2023p256_hqc128	48.37
19.10.2023x25519_hqc128	46.74
19.10.2023p384_hqc192	58.41
19.10.2023x448_hqc192	51.56
19.10.2023p521_hqc256	102.27
19.10.2023p521_kyber1024	66.37
19.10.2023p256_kyber512	58.01
19.10.2023x25519_kyber512	45.42
19.10.2023p256_kyber768	52.94
19.10.2023p384_kyber768	60.61
19.10.2023x25519_kyber768	45.15
19.10.2023x448_kyber768	47.03
19.10.2023x25519	45.88
20.10.2023bikel1	55.95
20.10.2023bikel3	58.37
20.10.2023bikel5	62.59
20.10.2023hqc128	44.79
20.10.2023hqc192	50.84
20.10.2023hqc256	80.61
20.10.2023kyber1024	46.38
20.10.2023kyber512	45.17
20.10.2023kyber768	46.36
20.10.2023p256_bikel1	57.32
20.10.2023x25519_bikel1	49.36
20.10.2023p384_bikel3	67.8
20.10.2023x448_bikel3	56.01
20.10.2023p521_bikel5	81.27
20.10.2023p256_hqc128	49.66
20.10.2023x25519_hqc128	47.59
20.10.2023p384_hqc192	57.49
20.10.2023x448_hqc192	48.37
20.10.2023p521_hqc256	101.82

20.10.2023p521_kyber1024	67.16
20.10.2023p256_kyber512	50.3
20.10.2023x25519_kyber512	47.47
20.10.2023p256_kyber768	50.32
20.10.2023p384_kyber768	61.13
20.10.2023x25519_kyber768	48.14
20.10.2023x448_kyber768	50.95
20.10.2023x25519	50.4
20.10.2023bikel1	51.71
20.10.2023bikel3	56.59
20.10.2023bikel5	65.51
20.10.2023hqc128	46.13
20.10.2023hqc192	45.07
20.10.2023hqc256	82.04
20.10.2023kyber1024	46.2
20.10.2023kyber512	44.8
20.10.2023kyber768	46.09
20.10.2023p256_bikel1	54.1
20.10.2023x25519_bikel1	51.47
20.10.2023p384_bikel3	65.4
20.10.2023x448_bikel3	55.73
20.10.2023p521_bikel5	80.1
20.10.2023p256_hqc128	52.14
20.10.2023x25519_hqc128	47.77
20.10.2023p384_hqc192	57.03
20.10.2023x448_hqc192	49.81
20.10.2023p521_hqc256	100.76
20.10.2023p521_kyber1024	67.1
20.10.2023p256_kyber512	49.25
20.10.2023x25519_kyber512	47.64
20.10.2023p256_kyber768	49.22
20.10.2023p384_kyber768	140.92
20.10.2023x25519_kyber768	45.43
20.10.2023x448_kyber768	51.34
20.10.2023x25519	55.43
21.10.2023bikel1	57.9
21.10.2023bikel3	57.79
21.10.2023bikel5	67.1
21.10.2023hqc128	47.88
21.10.2023hqc192	48.06
21.10.2023hqc256	137.04

21.10.2023kyber1024	53.29
21.10.2023kyber512	48.17
21.10.2023kyber768	49.1
21.10.2023p256_bikel1	58.11
21.10.2023x25519_bikel1	59.06
21.10.2023p384_bikel3	70.33
21.10.2023x448_bikel3	62.48
21.10.2023p521_bikel5	91.9
21.10.2023p256_hqc128	55.52
21.10.2023x25519_hqc128	54.23
21.10.2023p384_hqc192	63.59
21.10.2023x448_hqc192	50.94
21.10.2023p521_hqc256	103.92
21.10.2023p521_kyber1024	72.29
21.10.2023p256_kyber512	52.11
21.10.2023x25519_kyber512	48.38
21.10.2023p256_kyber768	51.96
21.10.2023p384_kyber768	105.76
21.10.2023x25519_kyber768	46.7
21.10.2023x448_kyber768	49.55
21.10.2023x25519	44.01
22.10.2023bikel1	73.48
22.10.2023bikel3	57.82
22.10.2023bikel5	63.11
22.10.2023hqc128	47.29
22.10.2023hqc192	48.29
22.10.2023hqc256	82.4
22.10.2023kyber1024	54.76
22.10.2023kyber512	50.76
22.10.2023kyber768	46.9
22.10.2023p256_bikel1	60.21
22.10.2023x25519_bikel1	55.66
22.10.2023p384_bikel3	68.23
22.10.2023x448_bikel3	61.64
22.10.2023p521_bikel5	81.5
22.10.2023p256_hqc128	52.85
22.10.2023x25519_hqc128	52.54
22.10.2023p384_hqc192	61.3
22.10.2023x448_hqc192	56.38
22.10.2023p521_hqc256	100.21
22.10.2023p521_kyber1024	77.37

22.10.2023p256_kyber512	51.44
22.10.2023x25519_kyber512	57.03
22.10.2023p256_kyber768	52.95
22.10.2023p384_kyber768	67.18
22.10.2023x25519_kyber768	68.87
22.10.2023x448_kyber768	50.33
22.10.2023x25519	45.76
22.10.2023bikel1	65.44
22.10.2023bikel3	70.8
22.10.2023bikel5	88.55
22.10.2023hqc128	53.33
22.10.2023hqc192	60.79
22.10.2023hqc256	100.89
22.10.2023kyber1024	63.45
22.10.2023kyber512	62.12
22.10.2023kyber768	76.36
22.10.2023p256_bikel1	64.56
22.10.2023x25519_bikel1	54.22
22.10.2023p384_bikel3	67.06
22.10.2023x448_bikel3	82.34
22.10.2023p521_bikel5	83.17
22.10.2023p256_hqc128	58.53
22.10.2023x25519_hqc128	60.83
22.10.2023p384_hqc192	72.63
22.10.2023x448_hqc192	75.29
22.10.2023p521_hqc256	136.37
22.10.2023p521_kyber1024	76.11
22.10.2023p256_kyber512	67.99
22.10.2023x25519_kyber512	50.56
22.10.2023p256_kyber768	59.67
22.10.2023p384_kyber768	61.17
22.10.2023x25519_kyber768	58.63
22.10.2023x448_kyber768	69.73
22.10.2023x25519	78.88
23.10.2023bikel1	55.77
23.10.2023bikel3	64.64
23.10.2023bikel5	66.53
23.10.2023hqc128	52.17
23.10.2023hqc192	50.37
23.10.2023hqc256	90.88
23.10.2023kyber1024	50.16

23.10.2023kyber512	49.4
23.10.2023kyber768	51.09
23.10.2023p256_bikel1	58.28
23.10.2023x25519_bikel1	57.55
23.10.2023p384_bikel3	64.34
23.10.2023x448_bikel3	62.11
23.10.2023p521_bikel5	83.6
23.10.2023p256_hqc128	58.5
23.10.2023x25519_hqc128	51.99
23.10.2023p384_hqc192	67.88
23.10.2023x448_hqc192	57.48
23.10.2023p521_hqc256	118.06
23.10.2023p521_kyber1024	70.53
23.10.2023p256_kyber512	55.74
23.10.2023x25519_kyber512	48.6
23.10.2023p256_kyber768	54.25
23.10.2023p384_kyber768	71.48
23.10.2023x25519_kyber768	48.97
23.10.2023x448_kyber768	53.83
23.10.2023x25519	62.61
23.10.2023bikel1	49.94
23.10.2023bikel3	53.86
23.10.2023bikel5	61.33
23.10.2023hqc128	44.49
23.10.2023hqc192	45.02
23.10.2023hqc256	75.69
23.10.2023kyber1024	46.91
23.10.2023kyber512	44.92
23.10.2023kyber768	45.52
23.10.2023p256_bikel1	55.01
23.10.2023x25519_bikel1	50.38
23.10.2023p384_bikel3	60.14
23.10.2023x448_bikel3	57.06
23.10.2023p521_bikel5	76.08
23.10.2023p256_hqc128	47.47
23.10.2023x25519_hqc128	44
23.10.2023p384_hqc192	56.31
23.10.2023x448_hqc192	45.97
23.10.2023p521_hqc256	97.52
23.10.2023p521_kyber1024	67.67
23.10.2023p256_kyber512	48.14

23.10.2023x25519_kyber512	43.91
23.10.2023p256_kyber768	48.84
23.10.2023p384_kyber768	56.83
23.10.2023x25519_kyber768	45.39
23.10.2023x448_kyber768	48.05
23.10.2023x25519	60.39
24.10.2023bikel1	54.28
24.10.2023bikel3	58.67
24.10.2023bikel5	63.11
24.10.2023hqc128	45.78
24.10.2023hqc192	49.26
24.10.2023hqc256	77.84
24.10.2023kyber1024	47.4
24.10.2023kyber512	46.16
24.10.2023kyber768	48.6
24.10.2023p256_bikel1	54.99
24.10.2023x25519_bikel1	105.39
24.10.2023p384_bikel3	64.55
24.10.2023x448_bikel3	57.67
24.10.2023p521_bikel5	85.48
24.10.2023p256_hqc128	50.97
24.10.2023x25519_hqc128	47.8
24.10.2023p384_hqc192	57.96
24.10.2023x448_hqc192	50.08
24.10.2023p521_hqc256	284.38
24.10.2023p521_kyber1024	66.89
24.10.2023p256_kyber512	52.25
24.10.2023x25519_kyber512	48.03
24.10.2023p256_kyber768	48.93
24.10.2023p384_kyber768	60.32
24.10.2023x25519_kyber768	49.33
24.10.2023x448_kyber768	49.03
24.10.2023x25519	42.95
25.10.2023bikel1	50.09
25.10.2023bikel3	55.61
25.10.2023bikel5	67.47
25.10.2023hqc128	45.85
25.10.2023hqc192	45.5
25.10.2023hqc256	81.55
25.10.2023kyber1024	44.55
25.10.2023kyber512	45.42

25.10.2023kyber768	47.15
25.10.2023p256_bikel1	54.53
25.10.2023x25519_bikel1	52.8
25.10.2023p384_bikel3	64.26
25.10.2023x448_bikel3	56.91
25.10.2023p521_bikel5	80.31
25.10.2023p256_hqc128	49.33
25.10.2023x25519_hqc128	47.41
25.10.2023p384_hqc192	60
25.10.2023x448_hqc192	49.37
25.10.2023p521_hqc256	101.62
25.10.2023p521_kyber1024	67.2
25.10.2023p256_kyber512	50.6
25.10.2023x25519_kyber512	46.97
25.10.2023p256_kyber768	53.56
25.10.2023p384_kyber768	62.6
25.10.2023x25519_kyber768	46.33
25.10.2023x448_kyber768	48.83
25.10.2023x25519	45.6
25.10.2023bikel1	57.03
25.10.2023bikel3	59.67
25.10.2023bikel5	64.79
25.10.2023hqc128	68.45
25.10.2023hqc192	49.22
25.10.2023hqc256	82.55
25.10.2023kyber1024	51.41
25.10.2023kyber512	50.74
25.10.2023kyber768	48.97
25.10.2023p256_bikel1	56
25.10.2023x25519_bikel1	52.26
25.10.2023p384_bikel3	67.51
25.10.2023x448_bikel3	58.96
25.10.2023p521_bikel5	87.69
25.10.2023p256_hqc128	51.91
25.10.2023x25519_hqc128	48.24
25.10.2023p384_hqc192	58.99
25.10.2023x448_hqc192	51.39
25.10.2023p521_hqc256	103.01
25.10.2023p521_kyber1024	69.54
25.10.2023p256_kyber512	61.73
25.10.2023x25519_kyber512	50.64

25.10.2023p256_kyber768	53.98
25.10.2023p384_kyber768	63.06
25.10.2023x25519_kyber768	48.91
25.10.2023x448_kyber768	56.33
25.10.2023x25519	53.42
26.10.2023bikel1	48.65
26.10.2023bikel3	55.56
26.10.2023bikel5	63.08
26.10.2023hqc128	43.06
26.10.2023hqc192	43.6
26.10.2023hqc256	75.7
26.10.2023kyber1024	46.6
26.10.2023kyber512	43.82
26.10.2023kyber768	47.9
26.10.2023p256_bikel1	53.1
26.10.2023x25519_bikel1	51.95
26.10.2023p384_bikel3	65.18
26.10.2023x448_bikel3	55.63
26.10.2023p521_bikel5	79.45
26.10.2023p256_hqc128	51.39
26.10.2023x25519_hqc128	44.06
26.10.2023p384_hqc192	55.02
26.10.2023x448_hqc192	47.28
26.10.2023p521_hqc256	97.42
26.10.2023p521_kyber1024	64.72
26.10.2023p256_kyber512	49.19
26.10.2023x25519_kyber512	45.03
26.10.2023p256_kyber768	49.98
26.10.2023p384_kyber768	56.63
26.10.2023x25519_kyber768	45.96
26.10.2023x448_kyber768	47.68
26.10.2023x25519	45.82
29.10.2023bikel1	50.72
29.10.2023bikel3	55.32
29.10.2023bikel5	60.58
29.10.2023hqc128	44.54
29.10.2023hqc192	44.8
29.10.2023hqc256	76.99
29.10.2023kyber1024	43.76
29.10.2023kyber512	45.59
29.10.2023kyber768	44.03

29.10.2023p256_bikel1	53.29
29.10.2023x25519_bikel1	48.1
29.10.2023p384_bikel3	60.62
29.10.2023x448_bikel3	53.23
29.10.2023p521_bikel5	82.71
29.10.2023p256_hqc128	47.04
29.10.2023x25519_hqc128	43.94
29.10.2023p384_hqc192	54.97
29.10.2023x448_hqc192	44.58
29.10.2023p521_hqc256	97.39
29.10.2023p521_kyber1024	63.2
29.10.2023p256_kyber512	49.6
29.10.2023x25519_kyber512	48.46
29.10.2023p256_kyber768	51.65
29.10.2023p384_kyber768	58.67
29.10.2023x25519_kyber768	46.16
29.10.2023x448_kyber768	48.13
29.10.2023x25519	49.12
29.10.2023bikel1	50.37
29.10.2023bikel3	54.16
29.10.2023bikel5	63.41
29.10.2023hqc128	44.97
29.10.2023hqc192	44.47
29.10.2023hqc256	78.46
29.10.2023kyber1024	43.15
29.10.2023kyber512	47.35
29.10.2023kyber768	46.83
29.10.2023p256_bikel1	53.03
29.10.2023x25519_bikel1	69.39
29.10.2023p384_bikel3	63.64
29.10.2023x448_bikel3	86.69
29.10.2023p521_bikel5	77.82
29.10.2023p256_hqc128	74.75
29.10.2023x25519_hqc128	45.98
29.10.2023p384_hqc192	95.79
29.10.2023x448_hqc192	46.08
29.10.2023p521_hqc256	150.21
29.10.2023p521_kyber1024	64.19
29.10.2023p256_kyber512	76.18
29.10.2023x25519_kyber512	44.4
29.10.2023p256_kyber768	73.08

29.10.2023p384_kyber768	55.19
29.10.2023x25519_kyber768	71.75
29.10.2023x448_kyber768	47.43
29.10.2023x25519	46.82
30.10.2023bikel1	48.06
30.10.2023bikel3	56.73
30.10.2023bikel5	63.73
30.10.2023hqc128	43.99
30.10.2023hqc192	45.33
30.10.2023hqc256	77.7
30.10.2023kyber1024	46.26
30.10.2023kyber512	43.69
30.10.2023kyber768	45.42
30.10.2023p256_bikel1	57.43
30.10.2023x25519_bikel1	47.75
30.10.2023p384_bikel3	65.39
30.10.2023x448_bikel3	57.42
30.10.2023p521_bikel5	79.35
30.10.2023p256_hqc128	47.35
30.10.2023x25519_hqc128	44.98
30.10.2023p384_hqc192	54.44
30.10.2023x448_hqc192	52.33
30.10.2023p521_hqc256	99.64
30.10.2023p521_kyber1024	63.63
30.10.2023p256_kyber512	48.09
30.10.2023x25519_kyber512	45.68
30.10.2023p256_kyber768	51.01
30.10.2023p384_kyber768	61.68
30.10.2023x25519_kyber768	45.54
30.10.2023x448_kyber768	47.21
30.10.2023x25519	45.09

B.2.3 Ecdsap256.csv

Date	*Curve*	*Average Time (ms)*
18.10.2023	bikel1	50.57
18.10.2023	bikel3	53.37
18.10.2023	bikel5	65.15

18.10.2023hqc128	46.81
18.10.2023hqc192	47.51
18.10.2023hqc256	80.38
18.10.2023kyber1024	44.13
18.10.2023kyber512	44.65
18.10.2023kyber768	45.22
18.10.2023p256_bikel1	53.53
18.10.2023x25519_bikel1	49.79
18.10.2023p384_bikel3	69.52
18.10.2023x448_bikel3	56.82
18.10.2023p521_bikel5	81.26
18.10.2023p256_hqc128	47.54
18.10.2023x25519_hqc128	45.7
18.10.2023p384_hqc192	67.22
18.10.2023x448_hqc192	50.82
18.10.2023p521_hqc256	101.11
18.10.2023p521_kyber1024	66.77
18.10.2023p256_kyber512	47.87
18.10.2023x25519_kyber512	48.73
18.10.2023p256_kyber768	49.43
18.10.2023p384_kyber768	57.71
18.10.2023x25519_kyber768	45.45
18.10.2023x448_kyber768	48.63
18.10.2023x25519	50.04
19.10.2023bikel1	54.5
19.10.2023bikel3	59.12
19.10.2023bikel5	71.8
19.10.2023hqc128	50.22
19.10.2023hqc192	46.51
19.10.2023hqc256	194.81
19.10.2023kyber1024	46.73
19.10.2023kyber512	43.36
19.10.2023kyber768	45.67
19.10.2023p256_bikel1	53.2
19.10.2023x25519_bikel1	104.84
19.10.2023p384_bikel3	65.32
19.10.2023x448_bikel3	57.14
19.10.2023p521_bikel5	79.21
19.10.2023p256_hqc128	50.79
19.10.2023x25519_hqc128	46.28
19.10.2023p384_hqc192	57.28

19.10.2023x448_hqc192	51.07
19.10.2023p521_hqc256	99.85
19.10.2023p521_kyber1024	65.11
19.10.2023p256_kyber512	49.74
19.10.2023x25519_kyber512	47.33
19.10.2023p256_kyber768	47.91
19.10.2023p384_kyber768	57.88
19.10.2023x25519_kyber768	45.79
19.10.2023x448_kyber768	122.04
19.10.2023x25519	47.09
20.10.2023bikel1	51.82
20.10.2023bikel3	57.58
20.10.2023bikel5	62.07
20.10.2023hqc128	47.9
20.10.2023hqc192	47.69
20.10.2023hqc256	81.12
20.10.2023kyber1024	46.07
20.10.2023kyber512	44.81
20.10.2023kyber768	44.64
20.10.2023p256_bikel1	52.62
20.10.2023x25519_bikel1	48.87
20.10.2023p384_bikel3	65.07
20.10.2023x448_bikel3	56.75
20.10.2023p521_bikel5	78.34
20.10.2023p256_hqc128	48.83
20.10.2023x25519_hqc128	46.48
20.10.2023p384_hqc192	62.59
20.10.2023x448_hqc192	48.84
20.10.2023p521_hqc256	100.18
20.10.2023p521_kyber1024	67.06
20.10.2023p256_kyber512	49.81
20.10.2023x25519_kyber512	46.56
20.10.2023p256_kyber768	173.94
20.10.2023p384_kyber768	59.06
20.10.2023x25519_kyber768	48.99
20.10.2023x448_kyber768	46.19
20.10.2023x25519	43.95
20.10.2023bikel1	50.09
20.10.2023bikel3	57.19
20.10.2023bikel5	65.38
20.10.2023hqc128	48.3

20.10.2023hqc192	45.17
20.10.2023hqc256	80.45
20.10.2023kyber1024	49.26
20.10.2023kyber512	46.27
20.10.2023kyber768	44.34
20.10.2023p256_bikel1	52.91
20.10.2023x25519_bikel1	50.01
20.10.2023p384_bikel3	60.92
20.10.2023x448_bikel3	59.39
20.10.2023p521_bikel5	80.58
20.10.2023p256_hqc128	46.32
20.10.2023x25519_hqc128	45.96
20.10.2023p384_hqc192	65.65
20.10.2023x448_hqc192	47.5
20.10.2023p521_hqc256	118.75
20.10.2023p521_kyber1024	70.45
20.10.2023p256_kyber512	49.66
20.10.2023x25519_kyber512	44.49
20.10.2023p256_kyber768	55.1
20.10.2023p384_kyber768	58.09
20.10.2023x25519_kyber768	45.53
20.10.2023x448_kyber768	45.37
20.10.2023x25519	53.4
21.10.2023bikel1	54.5
21.10.2023bikel3	62.26
21.10.2023bikel5	67.62
21.10.2023hqc128	49.38
21.10.2023hqc192	47.1
21.10.2023hqc256	86.96
21.10.2023kyber1024	62.59
21.10.2023kyber512	48.93
21.10.2023kyber768	53.24
21.10.2023p256_bikel1	57.82
21.10.2023x25519_bikel1	74.48
21.10.2023p384_bikel3	68.99
21.10.2023x448_bikel3	61.82
21.10.2023p521_bikel5	89.59
21.10.2023p256_hqc128	52.85
21.10.2023x25519_hqc128	47.68
21.10.2023p384_hqc192	59.71
21.10.2023x448_hqc192	52.1

21.10.2023p521_hqc256	105.78
21.10.2023p521_kyber1024	65.9
21.10.2023p256_kyber512	56.48
21.10.2023x25519_kyber512	49.77
21.10.2023p256_kyber768	53.16
21.10.2023p384_kyber768	60.01
21.10.2023x25519_kyber768	48.28
21.10.2023x448_kyber768	52.96
21.10.2023x25519	46.19
22.10.2023bikel1	56.9
22.10.2023bikel3	61.1
22.10.2023bikel5	66.3
22.10.2023hqc128	47.72
22.10.2023hqc192	49.19
22.10.2023hqc256	80.4
22.10.2023kyber1024	51.15
22.10.2023kyber512	45.85
22.10.2023kyber768	59.09
22.10.2023p256_bikel1	52.95
22.10.2023x25519_bikel1	52.03
22.10.2023p384_bikel3	61.78
22.10.2023x448_bikel3	57.17
22.10.2023p521_bikel5	107.88
22.10.2023p256_hqc128	59.35
22.10.2023x25519_hqc128	65.03
22.10.2023p384_hqc192	64.2
22.10.2023x448_hqc192	50.78
22.10.2023p521_hqc256	101.92
22.10.2023p521_kyber1024	65.9
22.10.2023p256_kyber512	50.53
22.10.2023x25519_kyber512	43.55
22.10.2023p256_kyber768	48.65
22.10.2023p384_kyber768	55.81
22.10.2023x25519_kyber768	66.54
22.10.2023x448_kyber768	48.49
22.10.2023x25519	51.32
22.10.2023bikel1	74
22.10.2023bikel3	71.34
22.10.2023bikel5	84.86
22.10.2023hqc128	52.44
22.10.2023hqc192	110.79

22.10.2023hqc256	105.26
22.10.2023kyber1024	62.61
22.10.2023kyber512	53.49
22.10.2023kyber768	54.1
22.10.2023p256_bikel1	65.46
22.10.2023x25519_bikel1	60.7
22.10.2023p384_bikel3	94.5
22.10.2023x448_bikel3	73.38
22.10.2023p521_bikel5	90.73
22.10.2023p256_hqc128	61.64
22.10.2023x25519_hqc128	67.68
22.10.2023p384_hqc192	69.95
22.10.2023x448_hqc192	62.87
22.10.2023p521_hqc256	126.36
22.10.2023p521_kyber1024	88.36
22.10.2023p256_kyber512	65.61
22.10.2023x25519_kyber512	64.22
22.10.2023p256_kyber768	55.63
22.10.2023p384_kyber768	74.5
22.10.2023x25519_kyber768	53.06
22.10.2023x448_kyber768	51.49
22.10.2023x25519	46.78
23.10.2023bikel1	56
23.10.2023bikel3	63.54
23.10.2023bikel5	68.19
23.10.2023hqc128	50.27
23.10.2023hqc192	51.42
23.10.2023hqc256	103.72
23.10.2023kyber1024	48.23
23.10.2023kyber512	57.56
23.10.2023kyber768	52.1
23.10.2023p256_bikel1	63.17
23.10.2023x25519_bikel1	51.12
23.10.2023p384_bikel3	65.82
23.10.2023x448_bikel3	58.72
23.10.2023p521_bikel5	88.79
23.10.2023p256_hqc128	53.17
23.10.2023x25519_hqc128	48.49
23.10.2023p384_hqc192	63.99
23.10.2023x448_hqc192	53.56
23.10.2023p521_hqc256	116.29

23.10.2023p521_kyber1024	77.81
23.10.2023p256_kyber512	52.22
23.10.2023x25519_kyber512	48.51
23.10.2023p256_kyber768	53.34
23.10.2023p384_kyber768	60.86
23.10.2023x25519_kyber768	48.85
23.10.2023x448_kyber768	49.85
23.10.2023x25519	46.51
23.10.2023bikel1	50.42
23.10.2023bikel3	54.93
23.10.2023bikel5	59.85
23.10.2023hqc128	46.64
23.10.2023hqc192	44.21
23.10.2023hqc256	75.98
23.10.2023kyber1024	45.12
23.10.2023kyber512	49.13
23.10.2023kyber768	47
23.10.2023p256_bikel1	49.88
23.10.2023x25519_bikel1	49.82
23.10.2023p384_bikel3	62.76
23.10.2023x448_bikel3	56.54
23.10.2023p521_bikel5	78.08
23.10.2023p256_hqc128	47.32
23.10.2023x25519_hqc128	43.89
23.10.2023p384_hqc192	58.25
23.10.2023x448_hqc192	48
23.10.2023p521_hqc256	96.89
23.10.2023p521_kyber1024	63.54
23.10.2023p256_kyber512	51.63
23.10.2023x25519_kyber512	43.95
23.10.2023p256_kyber768	45.6
23.10.2023p384_kyber768	55.81
23.10.2023x25519_kyber768	42.89
23.10.2023x448_kyber768	45.2
23.10.2023x25519	49.02
24.10.2023bikel1	50.19
24.10.2023bikel3	57.35
24.10.2023bikel5	66.31
24.10.2023hqc128	48.33
24.10.2023hqc192	47.17
24.10.2023hqc256	76.33

24.10.2023kyber1024	45.77
24.10.2023kyber512	49.08
24.10.2023kyber768	45.92
24.10.2023p256_bikel1	54.33
24.10.2023x25519_bikel1	51.02
24.10.2023p384_bikel3	64.6
24.10.2023x448_bikel3	58.81
24.10.2023p521_bikel5	79.48
24.10.2023p256_hqc128	51.98
24.10.2023x25519_hqc128	47.4
24.10.2023p384_hqc192	58.62
24.10.2023x448_hqc192	49.86
24.10.2023p521_hqc256	99.72
24.10.2023p521_kyber1024	63.02
24.10.2023p256_kyber512	49.7
24.10.2023x25519_kyber512	46.32
24.10.2023p256_kyber768	49.47
24.10.2023p384_kyber768	56.49
24.10.2023x25519_kyber768	46.66
24.10.2023x448_kyber768	47.74
24.10.2023x25519	63.27
25.10.2023bikel1	48.24
25.10.2023bikel3	56.6
25.10.2023bikel5	63.76
25.10.2023hqc128	44.53
25.10.2023hqc192	48.03
25.10.2023hqc256	79.69
25.10.2023kyber1024	46.84
25.10.2023kyber512	43.17
25.10.2023kyber768	43.34
25.10.2023p256_bikel1	56.71
25.10.2023x25519_bikel1	48.44
25.10.2023p384_bikel3	65.06
25.10.2023x448_bikel3	54.93
25.10.2023p521_bikel5	79.87
25.10.2023p256_hqc128	50.45
25.10.2023x25519_hqc128	46.11
25.10.2023p384_hqc192	60.47
25.10.2023x448_hqc192	48.71
25.10.2023p521_hqc256	97.61
25.10.2023p521_kyber1024	67.31

25.10.2023p256_kyber512	48.41
25.10.2023x25519_kyber512	44.45
25.10.2023p256_kyber768	49.6
25.10.2023p384_kyber768	56.35
25.10.2023x25519_kyber768	44.33
25.10.2023x448_kyber768	46.19
25.10.2023x25519	48.63
25.10.2023bikel1	55.1
25.10.2023bikel3	58.43
25.10.2023bikel5	68.15
25.10.2023hqc128	49.64
25.10.2023hqc192	51.71
25.10.2023hqc256	84.27
25.10.2023kyber1024	50.66
25.10.2023kyber512	48.07
25.10.2023kyber768	46.21
25.10.2023p256_bikel1	58.88
25.10.2023x25519_bikel1	48.92
25.10.2023p384_bikel3	66.9
25.10.2023x448_bikel3	56.87
25.10.2023p521_bikel5	78.13
25.10.2023p256_hqc128	50.11
25.10.2023x25519_hqc128	47.01
25.10.2023p384_hqc192	61.09
25.10.2023x448_hqc192	54.88
25.10.2023p521_hqc256	104.61
25.10.2023p521_kyber1024	68.31
25.10.2023p256_kyber512	52.64
25.10.2023x25519_kyber512	48.21
25.10.2023p256_kyber768	53.92
25.10.2023p384_kyber768	61.23
25.10.2023x25519_kyber768	49.87
25.10.2023x448_kyber768	50.33
25.10.2023x25519	53.7
26.10.2023bikel1	49.13
26.10.2023bikel3	59.53
26.10.2023bikel5	62.02
26.10.2023hqc128	45.75
26.10.2023hqc192	44.52
26.10.2023hqc256	78.15
26.10.2023kyber1024	47.46

26.10.2023kyber512	41.22
26.10.2023kyber768	45.52
26.10.2023p256_bikel1	51.78
26.10.2023x25519_bikel1	49.37
26.10.2023p384_bikel3	64.62
26.10.2023x448_bikel3	59.37
26.10.2023p521_bikel5	79.05
26.10.2023p256_hqc128	48.39
26.10.2023x25519_hqc128	44.04
26.10.2023p384_hqc192	58.64
26.10.2023x448_hqc192	50.33
26.10.2023p521_hqc256	97.53
26.10.2023p521_kyber1024	63.31
26.10.2023p256_kyber512	47.7
26.10.2023x25519_kyber512	43.63
26.10.2023p256_kyber768	48.53
26.10.2023p384_kyber768	56.46
26.10.2023x25519_kyber768	43.54
26.10.2023x448_kyber768	45.66
26.10.2023x25519	44.07
29.10.2023bikel1	50.76
29.10.2023bikel3	61.29
29.10.2023bikel5	64.63
29.10.2023hqc128	51.21
29.10.2023hqc192	45.24
29.10.2023hqc256	77.97
29.10.2023kyber1024	43.49
29.10.2023kyber512	46.62
29.10.2023kyber768	44.56
29.10.2023p256_bikel1	53.62
29.10.2023x25519_bikel1	47.25
29.10.2023p384_bikel3	57.52
29.10.2023x448_bikel3	55.08
29.10.2023p521_bikel5	75.89
29.10.2023p256_hqc128	49.55
29.10.2023x25519_hqc128	44.62
29.10.2023p384_hqc192	59.71
29.10.2023x448_hqc192	47.46
29.10.2023p521_hqc256	95.45
29.10.2023p521_kyber1024	67.45
29.10.2023p256_kyber512	48.77

29.10.2023x25519_kyber512	43.64
29.10.2023p256_kyber768	47.53
29.10.2023p384_kyber768	56.96
29.10.2023x25519_kyber768	44.15
29.10.2023x448_kyber768	47.14
29.10.2023x25519	51.44
29.10.2023bikel1	48.35
29.10.2023bikel3	55.88
29.10.2023bikel5	60.98
29.10.2023hqc128	43.98
29.10.2023hqc192	44.69
29.10.2023hqc256	76.15
29.10.2023kyber1024	60.37
29.10.2023kyber512	43.25
29.10.2023kyber768	43.01
29.10.2023p256_bikel1	89.72
29.10.2023x25519_bikel1	47.33
29.10.2023p384_bikel3	64.76
29.10.2023x448_bikel3	55.16
29.10.2023p521_bikel5	74.92
29.10.2023p256_hqc128	105.72
29.10.2023x25519_hqc128	43.55
29.10.2023p384_hqc192	56.29
29.10.2023x448_hqc192	46.89
29.10.2023p521_hqc256	96.68
29.10.2023p521_kyber1024	61.52
29.10.2023p256_kyber512	45.9
29.10.2023x25519_kyber512	43.95
29.10.2023p256_kyber768	47.26
29.10.2023p384_kyber768	58.5
29.10.2023x25519_kyber768	43.96
29.10.2023x448_kyber768	45.07
29.10.2023x25519	50.61
30.10.2023bikel1	49.35
30.10.2023bikel3	54.45
30.10.2023bikel5	64.2
30.10.2023hqc128	46.68
30.10.2023hqc192	45
30.10.2023hqc256	77.94
30.10.2023kyber1024	46.28
30.10.2023kyber512	45

30.10.2023kyber768	43.96
30.10.2023p256_bikel1	51.27
30.10.2023x25519_bikel1	52.87
30.10.2023p384_bikel3	64.67
30.10.2023x448_bikel3	56.15
30.10.2023p521_bikel5	78.06
30.10.2023p256_hqc128	48.75
30.10.2023x25519_hqc128	43.9
30.10.2023p384_hqc192	57.23
30.10.2023x448_hqc192	49.85
30.10.2023p521_hqc256	104.34
30.10.2023p521_kyber1024	63.75
30.10.2023p256_kyber512	48.14
30.10.2023x25519_kyber512	56.98
30.10.2023p256_kyber768	48.74
30.10.2023p384_kyber768	58.94
30.10.2023x25519_kyber768	43.65
30.10.2023x448_kyber768	44.97
30.10.2023x25519	47.8

B.2.4 P384_dilithium3.csv

Date	*Curve*	*Average Time (ms)*
04.11.2023	bikel3	81.38
04.11.2023	hqc192	82.01
04.11.2023	kyber768	74.16
04.11.2023	p384_bikel3	84.94
04.11.2023	x448_bikel3	76.83
04.11.2023	p384_hqc192	84.94
04.11.2023	x448_hqc192	81.13
04.11.2023	p384_kyber768	71.88
04.11.2023	x448_kyber768	76.18
04.11.2023	x25519	54.95
04.11.2023	bikel3	78.47
04.11.2023	hqc192	78.61
04.11.2023	kyber768	76.65
04.11.2023	p384_bikel3	83.99
04.11.2023	x448_bikel3	77.94

04.11.2023p384_hqc192	86.78
04.11.2023x448_hqc192	79.37
04.11.2023p384_kyber768	61.14
04.11.2023x448_kyber768	72.23
04.11.2023x25519	59.71
04.11.2023bikel3	73.74
04.11.2023hqc192	78.92
04.11.2023kyber768	73.27
04.11.2023p384_bikel3	83.42
04.11.2023x448_bikel3	77.99
04.11.2023p384_hqc192	84.72
04.11.2023x448_hqc192	79.37
04.11.2023p384_kyber768	77.48
04.11.2023x448_kyber768	75.15
04.11.2023x25519	61.02
04.11.2023bikel3	75.13
04.11.2023hqc192	78.01
04.11.2023kyber768	74.64
04.11.2023p384_bikel3	86.49
04.11.2023x448_bikel3	77.63
04.11.2023p384_hqc192	86.08
04.11.2023x448_hqc192	78.64
04.11.2023p384_kyber768	81.01
04.11.2023x448_kyber768	61.96
04.11.2023x25519	52.92
04.11.2023bikel3	76.92
04.11.2023hqc192	76.74
04.11.2023kyber768	69.06
04.11.2023p384_bikel3	83.81
04.11.2023x448_bikel3	77.78
04.11.2023p384_hqc192	83.99
04.11.2023x448_hqc192	79.49
04.11.2023p384_kyber768	80.37
04.11.2023x448_kyber768	77.24
04.11.2023x25519	54.59
04.11.2023bikel3	75.59
04.11.2023hqc192	77.88
04.11.2023kyber768	69.74
04.11.2023p384_bikel3	85.11
04.11.2023x448_bikel3	81.41
04.11.2023p384_hqc192	85.65

04.11.2023x448_hqc192	81.32
04.11.2023p384_kyber768	116.12
04.11.2023x448_kyber768	75.87
04.11.2023x25519	64.93
04.11.2023bikel3	76.69
04.11.2023hqc192	78.45
04.11.2023kyber768	68.59
04.11.2023p384_bikel3	81.97
04.11.2023x448_bikel3	79.58
04.11.2023p384_hqc192	82.56
04.11.2023x448_hqc192	79.73
04.11.2023p384_kyber768	81.84
04.11.2023x448_kyber768	75.49
04.11.2023x25519	69.91
04.11.2023bikel3	75.84
04.11.2023hqc192	78.46
04.11.2023kyber768	73.21
04.11.2023p384_bikel3	84.13
04.11.2023x448_bikel3	77.82
04.11.2023p384_hqc192	83.79
04.11.2023x448_hqc192	80.84
04.11.2023p384_kyber768	77.95
04.11.2023x448_kyber768	74.82
04.11.2023x25519	61.4
04.11.2023bikel3	76.73
04.11.2023hqc192	77.9
04.11.2023kyber768	67.62
04.11.2023p384_bikel3	82.97
04.11.2023x448_bikel3	78.06
04.11.2023p384_hqc192	82.85
04.11.2023x448_hqc192	80.59
04.11.2023p384_kyber768	77.07
04.11.2023x448_kyber768	74.22
04.11.2023x25519	57.8
04.11.2023bikel3	75.63
04.11.2023hqc192	78.04
04.11.2023kyber768	75.97
04.11.2023p384_bikel3	84.63
04.11.2023x448_bikel3	79.85
04.11.2023p384_hqc192	88.48
04.11.2023x448_hqc192	81.42

04.11.2023p384_kyber768	78.94
04.11.2023x448_kyber768	77.83
04.11.2023x25519	74.01
04.11.2023bikel3	82.2
04.11.2023hqc192	79.48
04.11.2023kyber768	74.28
04.11.2023p384_bikel3	82.85
04.11.2023x448_bikel3	79.92
04.11.2023p384_hqc192	88.2
04.11.2023x448_hqc192	79.47
04.11.2023p384_kyber768	81.64
04.11.2023x448_kyber768	75.41
04.11.2023x25519	62.95
04.11.2023bikel3	74.68
04.11.2023hqc192	76.73
04.11.2023kyber768	74.98
04.11.2023p384_bikel3	85.23
04.11.2023x448_bikel3	76.29
04.11.2023p384_hqc192	85.79
04.11.2023x448_hqc192	79.5
04.11.2023p384_kyber768	72.63
04.11.2023x448_kyber768	74.45
04.11.2023x25519	61.05
04.11.2023bikel3	77.17
04.11.2023hqc192	76.44
04.11.2023kyber768	62.96
04.11.2023p384_bikel3	82.26
04.11.2023x448_bikel3	79.08
04.11.2023p384_hqc192	85.93
04.11.2023x448_hqc192	79.17
04.11.2023p384_kyber768	70.78
04.11.2023x448_kyber768	75.56
04.11.2023x25519	67.21
04.11.2023bikel3	81.72
04.11.2023hqc192	78.88
04.11.2023kyber768	63.63
04.11.2023p384_bikel3	84.63
04.11.2023x448_bikel3	75.69
04.11.2023p384_hqc192	86.59
04.11.2023x448_hqc192	79.28
04.11.2023p384_kyber768	83.33

04.11.2023x448_kyber768	75.57
04.11.2023x25519	54.25
04.11.2023bikel3	77.1
04.11.2023hqc192	78.47
04.11.2023kyber768	73.81
04.11.2023p384_bikel3	82.42
05.11.2023x448_bikel3	76.38
05.11.2023p384_hqc192	89.14
05.11.2023x448_hqc192	80.24
05.11.2023p384_kyber768	81.08
05.11.2023x448_kyber768	72.96
05.11.2023x25519	51.52
05.11.2023bikel3	76.88
05.11.2023hqc192	80.56
05.11.2023kyber768	69.8
05.11.2023p384_bikel3	84.53
05.11.2023x448_bikel3	78.46
05.11.2023p384_hqc192	85.37
05.11.2023x448_hqc192	79.21
05.11.2023p384_kyber768	81.42
05.11.2023x448_kyber768	69.45
05.11.2023x25519	59.85
05.11.2023bikel3	84.65
05.11.2023hqc192	79.14
05.11.2023kyber768	78
05.11.2023p384_bikel3	87.13
05.11.2023x448_bikel3	81.57
05.11.2023p384_hqc192	89.08
05.11.2023x448_hqc192	83.5
05.11.2023p384_kyber768	85.17
05.11.2023x448_kyber768	81.33
05.11.2023x25519	66.97
05.11.2023bikel3	77.01
05.11.2023hqc192	80.19
05.11.2023kyber768	76.24
05.11.2023p384_bikel3	85.65
05.11.2023x448_bikel3	78.09
05.11.2023p384_hqc192	86.92
05.11.2023x448_hqc192	81.27
05.11.2023p384_kyber768	83.78
05.11.2023x448_kyber768	78.9

05.11.2023x25519	62.53
05.11.2023bikel3	77.75
05.11.2023hqc192	80.94
05.11.2023kyber768	74.87
05.11.2023p384_bikel3	83.62
05.11.2023x448_bikel3	77.97
05.11.2023p384_hqc192	84.86
05.11.2023x448_hqc192	81.25
05.11.2023p384_kyber768	83.11
05.11.2023x448_kyber768	69.88
05.11.2023x25519	60.91
05.11.2023bikel3	76.4
05.11.2023hqc192	76.83
05.11.2023kyber768	70.66
05.11.2023p384_bikel3	82.32
05.11.2023x448_bikel3	79.95
05.11.2023p384_hqc192	84.18
05.11.2023x448_hqc192	80.25
05.11.2023p384_kyber768	78.33
05.11.2023x448_kyber768	74.98
05.11.2023x25519	68.1
05.11.2023bikel3	77.62
05.11.2023hqc192	78.86
05.11.2023kyber768	70.55
05.11.2023p384_bikel3	84.26
05.11.2023x448_bikel3	77.53
05.11.2023p384_hqc192	88.12
05.11.2023x448_hqc192	78.71
05.11.2023p384_kyber768	75.81
05.11.2023x448_kyber768	80.28
05.11.2023x25519	70.43
05.11.2023bikel3	79.51
05.11.2023hqc192	79.88
05.11.2023kyber768	75.04
05.11.2023p384_bikel3	82.89
05.11.2023x448_bikel3	78.96
05.11.2023p384_hqc192	84.04
05.11.2023x448_hqc192	80.66
05.11.2023p384_kyber768	73.95
05.11.2023x448_kyber768	75.18
05.11.2023x25519	68.9

05.11.2023bikel3	80.73
05.11.2023hqc192	78.06
05.11.2023kyber768	74.47
05.11.2023p384_bikel3	87.49
05.11.2023x448_bikel3	79.85
05.11.2023p384_hqc192	87.51
05.11.2023x448_hqc192	83.36
05.11.2023p384_kyber768	83.34
05.11.2023x448_kyber768	76.06
05.11.2023x25519	69.4
05.11.2023bikel3	79.01
05.11.2023hqc192	86.8
05.11.2023kyber768	75.77
05.11.2023p384_bikel3	85.65
05.11.2023x448_bikel3	82.13
05.11.2023p384_hqc192	87.38
05.11.2023x448_hqc192	82.85
05.11.2023p384_kyber768	78.35
05.11.2023x448_kyber768	83.77
05.11.2023x25519	72.92

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

_____	_____	_____
Ort	Datum	Unterschrift im Original