Vzpostavitev šifrirnih ključev v post-kvantni kriptografiji

Marko Kompara, Marko Hölbl
Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru
marko.kompara@um.si, marko.holbl@um.si

Izvleček

Razvoj kvantnega računalnika napreduje tako hitro, da čeprav kriptografsko signifikanten kvantni računalnik še ne obstaja in ga po najbolj optimističnih napovedih lahko pričakujemo komaj čez kakšno desetletje, je že potrebna zamenjava določenih ranljivih kriptografskih gradnikov, ki se danes uporabljajo za varovanje podatkov. Ta prispevek je namenjen pregledu mehanizmov za inkapsulacijo ključa (KEM), ki so kvantno varni algoritmi za dogovor o šifrirnih ključih. V prispevku se bomo dotaknili nevarnosti kvantnih računalnikov za tradicionalno kriptografijo, predstavili splošno delovanje algoritmov KEM ter specifične rešitve, za katere bomo predstavili njihove varnostne lastnosti ter izvedli analizo učinkovitosti oz. hitrosti njihovega delovanja. Na koncu bomo predstavili še nekaj primerov uporabe algoritmov KEM oz. hibridnih rešitev, ki združujejo tradicionalno in post-kvantno kriptografijo v pogosto uporabljenih protokolih.

Ključne besede: KEM, mehanizem za inkapsulacijo ključa, post-kvantna kriptografija, analiza.

Establishing Encryption Keys with Post-Quantum Cryptography

Abstract

The development of the quantum computer is advancing so fast that, although a cryptographically significant quantum computer does not yet exist and, according to the most optimistic forecasts, can only be expected in about a decade, some of the vulnerable cryptographic building blocks used to protect data today need to be replaced now. This paper is an overview of key encapsulation mechanisms (KEMs), which are quantum-secure algorithms for key agreement. In the paper, we will discuss the threat quantum computers pose to traditional cryptography, present the general operation of KEM algorithms and specific algorithms for which we will present their security properties, and perform an analysis of their performance. Finally, we will present some examples of KEM algorithms or hybrid solutions that combine traditional and post-quantum cryptography in commonly used protocols.

Keywords: KEM, key encapsulation mechanism, post-quantum cryptography, analysis.

1 UVOD

Čeprav ideja kvantnega računalništva obstaja že več desetletij, je bila njihova realna implementacija dolgo vprašljiva. Vendar je, z velikimi vložki denarja in raziskovalnega truda, ta v zadnjih letih postala dosegljiva resničnost. Kvantni računalniki so problem za kibernetsko varnost od leta 1994 [1], ko je Peter Shor odkril kvantni algoritem (od takrat imenovan Shorov algoritem), ki bi lahko, ko bodo kvantni računalniki postali dovolj močni, razbil tradicionalno asimetrično kriptografijo oz. kriptografijo z javnim ključem (npr. RSA, ECC), ki je temelj sodobne kibernetske varnosti. To je botrovalo nastanku post-kvantne kriptografije (angl. Post-Quantum Cryptography), ki zahteva, da so kriptografski elementi (poudarek je predvsem na algoritmih javnega ključa) varni pred kriptoanalitičnimi napadi tradicionalnega računalnika in kvantnega računalnika. Medtem ko kvantni

računalniki, ki so dovolj zmogljivi, da bi izničili varnost tradicionalnih asimetričnih algoritmov, še niso na voljo, je strategija napada SNDL (angl. Store-Now Decrypt-Later) že zaskrbljujoča in jo je treba nasloviti. Z napadom NSDL napadalec shrani vse šifrirane podatke, ki jih lahko zajame sedaj, v upanju, da jih bo dešifriral, ko bodo kvantni računalniki postali dovolj močni in dostopni.

Nacionalni inštitut za standarde in tehnologijo iz ZDA - NIST (angl. National Institute of Standards and Technology) se že skoraj desetletje pripravlja na posledice prihajajočih kvantnih računalnikov. NIST se tudi pripravlja na ukinitev uporabe tradicionalnih asimetričnih algoritmov do leta 2035 [2] in večina Evropskih držav, vključno s Slovenijo je podala skupno izjavo o namenu prehoda na post-kvantno kriptografijo do konca leta 2030 [3]. NIST je zato že leta 2016 začel tekmovanje za standardizacijo post-kvantne kriptografije, da bi poiskal nove rešitve, ki bi nadomestile tradicionalno asimetrično kriptografijo, ki jo trenutno uporabljamo in ki bo razbita pod kriptografsko relevantnim kvantnim računalnikom (angl. Cryptographically-relevant quantum computer). NIST je že izbral in tudi standardiziral nekatere post-kvantne algoritme za elektronsko podpisovanje in izmenjavo ključev [4]. V tem članku se bomo osredotočili na algoritme za izmenjavo ključa.

Šifriranje javnega ključa – PKE (angl. Public Key Encryption), lahko šifrira poljubno sporočilo, medtem ko mehanizem za inkapsulacijo ključa - KEM (angl. Key Encapsulation Mechanism) inkapsulira naključno kratko skrivnost (javni in zasebni ključi se uporabljajo za inkapsulacijo in dekapsulacijo). PKE se lahko uporablja za izmenjavo ključev (ključ/skrivnost se ustvari ločeno in nato šifrira). Primer takšne uporabe je algoritem RSA. Algoritmi KEM so kot omejeni PKE algoritmi, ki pa imajo prednost lažjega dokazovanja varnosti, so učinkovitejši in ne potrebujejo bitnega zapolnjevanja (angl. Padding). Algoritme KEM je mogoče izdelati iz PKE, kar tudi drži za vse algoritme KEM omenjene v tem prispevku. Prvi del tega poročila bo posvečen pregledu pomembnih post-kvantnih algoritmov KEM, njihovi varnosti in učinkovitosti.

Kako zmogljiv mora biti kriptografsko relevanten kvantni računalnik (tj. dovolj zmogljiv, da lahko razbije tradicionalno asimetrično kriptografijo), se s časom spreminja. Leta 2015 so raziskovalci ocenili, da bi kvantni računalnik potreboval milijardo (10^9) kubitov, da bi razbil 2048-bitni RSA [5]. To se je leta 2021 zmanjšalo na samo 20 milijonov kubitov, kar bi trajalo približno osem ur. Trenutno sta dva kvantna računalnika z največ qubiti razvila IBM in Atom Computing s 1121 in 1180 kubiti. Vendar pa niso vsi kubiti enaki, koliko šuma in kako učinkovito je ta šum mogoče izničiti, močno vpliva na moč kvantnega računalnika. IBM načrtuje, da bo do leta 2033 imel superračunalnik s sto tisoč kubiti [6].

Post-kvantni algoritmi KEM, ki so najbolj znani in temeljito preizkušeni, so šli skozi NIST-ovo tekmovanje. Natečaj je vključeval tudi algoritme za elektronske/digitalne podpise, vendar se bomo v tem prispevku osredotočili na rešitve KEM. Čeprav so na voljo novi in standardizirani algoritmi elektronskega podpisa, je prehod infrastrukture javnih ključev – PKI (angl. Public Key Infrastructure) na uporabo post-kvantne kriptografije težji in bo trajal več časa, kot prehod algoritmov za dogovor o ključu. Preverjanje pristnosti je izjemno pomembno za preprečevanje napadov s posrednikom - MitM (angl. Man-in-the-Middle). Na srečo, za razliko od tradicionalnih rešitev za dogovor o ključu, ki trpijo zaradi strategije napada SNDL in jih je treba obravnavati čim prej, preverjanje pristnosti nima enakih težav. Ena od glavnih aplikacij, za katere se uporabljajo elektronski podpisi, je zagotavljanje kratkotrajnega overjanja (podpis se ustvari, kmalu zatem overi in nato se nikoli več ne uporabi). Primer tega je vzpostavljanje povezave TLS. Takšnih primerov uporabe ni potrebno takoj spremeniti v kvantno odporno obliko, ker sposobnost ponovnega ustvarjanja podpisov v prihodnosti s kvantnimi računalniki ne izničuje varnosti preverjanja pristnosti danes in današnji podpis v prihodnosti ne bo imel nobene vrednosti (ker je uporaben samo kratkoročno). To pomeni, da post-kvantno overjanje še ni tako tako nujno in jih bo treba algoritme spremeniti šele, ko bo na voljo kvantni računalnik (ali ko bo verjetnost, da takšen računalnik obstaja, dovolj visoka). Medtem so trenutne rešitve, ki temeljijo na RSA in ECC, varne in bolj priročne. Seveda, vse to, velja samo za kratkoročno preverjanje pristnosti, situacija je nekoliko bolj zapletena za primere dolgoročne uporabe (npr. uporaba elektronskih podpisov v pogodbah ali verigah blokov (angl. Blockchain)). Medtem ko so algoritmi post-kvantnega podpisa standardizirani s strani NIST (npr. ML-DSA, SLH-DSA in prihajajoči FN-DSA) in obstaja nekaj protokolov za njihovo izvajanje za namene

overjanja entitet v komunikaciji (npr. [7], [8], [9]), se trenutno ne sprejemajo množično, ker so kvantni certifikati veliko večji od tipičnih potrdil, kar povzroča slabo učinkovitost in težave z združljivostjo z nekaterimi omrežnimi napravami.

Zadnji del tega prispevka bo namenjen rešitvam, ki uporabljajo obravnavane post-kvantne algoritme KEM. Imamo že dolgo uveljavljene protokole (npr. TLS), ki uporabljajo tradicionalno asimetrično kriptografijo za izmenjavo ključev in nato uporabljajo simetrične algoritme za šifriranje vseh izmenjanih podatkov. Kot je že bilo omenjeno, kvantni računalniki ne izničijo varnosti simetrične kriptografije. Za prehod obstoječih protokolov na kvantno varne algoritme bi zato bilo treba spremeniti/zamenjati tradicionalne asimetrične algoritme, ki se uporablja za izmenjavo ključev (in algoritme za overjanje, ki pa niso del tega prispevka). Eden od načinov za to bi bil, da se ti algoritmi preprosto zamenjajo s post-kvantnimi KEM. Ker so algoritmi KEM odporni na napade tradicionalnih računalnikov in kvantnih računalnikov, bodo nastali protokoli varni. Vendar pa je varnostna skupnost zelo nenaklonjena temu, ker so algoritmi KEM relativno novi, niso bili preizkušeni v realnih okoljih in niso dolgoročno dokazali svoje varnosti. Strah je, da bi revolucionarni napad na njihovo izvajanje ali osnovni problem, na katerem temeljijo povzročil, da bi specifični algoritmi KEM čez noč postali nevarni. Ta strah ni več prisoten pri trenutno uporabljenih algoritmih, ker se uporabljajo že zelo dolgo in v tem času niso bile ugotovljene nobene pomembne pomanjkljivosti. Iskanje pravega trenutka za prehod iz tradicionalnih na nove metode vzpostavitve ključev je zato problematično. Previden pristop je združevanje tradicionalnih in algoritmov KEM na način, da je treba razbiti oba algoritma, da bi razkrili zaščiteno skrivnost (tj. izmenjano skupno skrivnost/ključ). Na ta način zaščiteni podatki ne bodo razkriti, ko kvantni računalniki postanejo dovolj močni, da razbijejo tradicionalne algoritme, ker bo post-kvantni KEM odporen, in če pride do težav z novejšim algoritmom KEM, bo robusten tradicionalni algoritem preprečil najhujše posledice (tj. ne bodo takoj zlomljivi, vsaj ne, dokler ne bo obstajal kriptografsko relevantni kvantni računalnik). Te vrste mehanizmov, ki uporabljajo obe vrsti KEM, imenujemo post-kvantni/tradicionalni hibridi, pogosto samo hibridi. Poleg večjega zaupanja v moč kombiniranih algoritmov (ker bi moral napadalec razbiti oba) hibridni pristop podpira tudi enostavno in postopno (različne storitve bodo imele možnost migracije ob različnih časih brez težav) migracijo na popolnoma post-kvantne rešitve, ko bo post-kvantna kriptografija dovolj zrela. Edina slabost hibridne sestave, je bolj zahtevno oz. počasnejše izvajanje. V zadnjem delu tega prispevka bo predstavljena konstrukcija in primeri takšnih hibridnih algoritmov KEM in protokolov, ki jih uporabljajo.

2 POST-KVANTNI MEHANIZMI ZA INKAPSULACIJO KLJUČA (KEM)

KEM kandidati na NIST tekmovanju so bili predloženi novembra 2017 in od takrat so kandidati opravili strogo varnostno testiranje in testiranje učinkovitosti v več ciklih v katerih so se algoritmi tudi nekoliko spreminjali. Trenutno edini kandidat, ki je bil standardiziran je ML-KEM. ML-KEM je standardizirana različica (z nekaj zelo majhnimi spremembami) algoritma CRYSTALS-Kyber (v nadaljevanju samo Kyber). Kyber je bil izbran za standardizacijo s strani NIST, ker je bila to (po njihovem mnenju) najboljša rešitev, predvsem iz vidika učinkovitosti. NIST zatem odprl četrti krog tekmovanja za izbiro in standardizacijo enega ali več dodatnih algoritmov KEM [10].

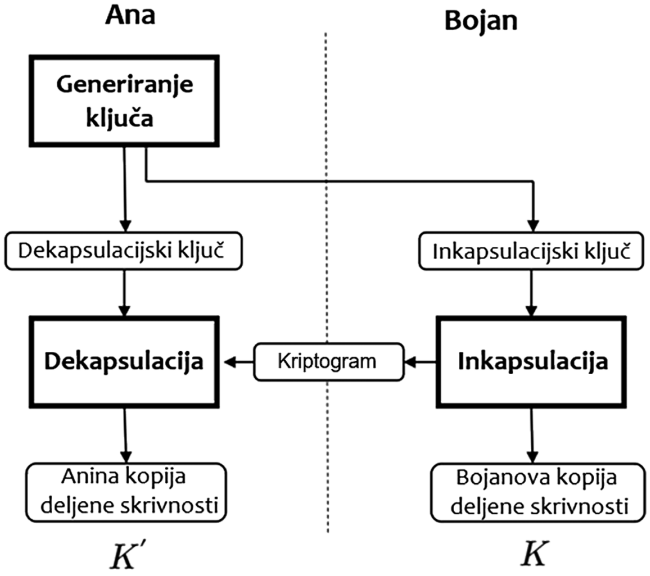
Prvi in za zdaj edini od algoritmov, ki so se uvrstili v četrti krog, za katerega je NIST napovedal standardizacijo je HQC [11]. Algoritem sicer še ni bil standardiziran, zato se bomo v tem prispevku še sklicevali na nestandardiziran HQC. Dva dodatna algoritma, ki jih bomo vključili v ta pregled sta, še vedno kandidata za standardizacijo (NIST ni napovedal, ali ima namen standardizirati še kakšno rešitev). To sta Classic McElice in BIKE. Edini algoritem, ki je vključen v četrti krog NIST tekmovanja, ki ga ne vključujemo v pregled je algoritem SIKE, ker je bil problem na podlagi katerega zagotavlja varno delovanje (tj. Izogeneza supersingularne eliptične krivulje) od začetka četrtega kroga dokazano razbit. Zadnji algoritem, o katerem bomo razpravljali tudi v tem prispevku je FrodoKEM, ki se kljub temu, da je bil del tekmovanja NIST, ni prebil v četrti krog. Čeprav NIST ni izbral algoritma FrodoKEM, ga nemški BSI [12] in francoski ANSSI [13] podpirata. Tabela 1 navaja algoritme KEM, obravnavane v tem dokumentu, skupaj z njihovim statusom (kandidati četrtega kroga tekmovanja NIST so še vedno lahko standardizirani s strani NIST) in iz katerega tipa post-kvantne kriptografije izhajajo.

Tabela 1: **Algoritmi KEM, obravnavani v tem prispevku.**

Algoritem KEM	Stanje	Vrsta
ML-KEM [14]	Standardiziran	Mreža (angl. Lattice)
HQC [15]	Izbran za standardizacijo	Koda (angl. Code)
Classic McEliece [16]	Kandidat v četrtem krogu	Koda (angl. Code)
FrodoKEM [17]	Nadomestni kandidat tretjega kroga	Mreža (angl. Lattice)
BIKE [18]	Kandidat v četrtem krogu	Koda (angl. Code)

Vredno je omeniti še algoritma Saber in NTRU, ki sta se prav tako uvrstila v tretji krog tekmovanja in sta se zelo dobro odrezala. NIST se je odločil, da jih ne bo standardiziral (enako kot FrodoKEM), ker temeljijo na isti osnovni strukturi kot Kyber/ML-KEM (tj. strukturirana mreža). NIST meni, da je Kyber najboljša rešitev od treh in ne želijo standardizirati več algoritmov, ki temeljijo na istem problemu, ker bi ranljivost v tem osnovnem problemu lahko povzročila razbitje vseh takšnih algoritmov. Iz istega razloga želi NIST standardizirati vsaj še en algoritem (to bo HQC), ki sloni na drugačnem osnovnem problemu. To zagotavlja, da če se v prihodnosti najde rešitev za problem mrež (zaradi česar bo ML-KEM neuporaben), bo rešitev preprost prehod na alternativen standardiziran KEM (Saber in NTRU ne bi bila dobra izbira, ker bi bila najverjetneje razbita istočasno). Hkrati se, zaradi preprostosti in lažje združljivosti, nabor standardiziranih algoritmov ohranja majhen (tj. se ne standardizira več algoritmov kot je potrebno). Glede na to, da Saber in NTRU ne bosta standardizirana in trenutno ni znakov, da imata podporo drugje, nista bila vključena v ta prispevek. FrodoKEM ni bil izbran s strani NIST iz istega razloga, vendar je FrodoKEM z varnostnega vidika bolj konzervativna rešitev (ne uporablja strukturiranih mrež, ampak mreže, ki niso strukturirane), zato ga nekateri smatrajo, kot dobro alternativo za ML-KEM.

Vsi algoritmi KEM so sestavljeni iz treh splošnih funkcij: generiranje ključev, inkapsulacija in dekapsulacija. Generiranje ključev nima vhodnih parametrov in ustvari zasebni (imenovan dekapsulacijski ključ) in javni ključ (imenovan inkapsulacijski ključ). Javni ključ je poslan entiteti s katero želimo izmenjati skrivnost. Inkapsulacijo izvede prejemnik javnega ključa. Inkapsulacija uporabi javni ključ kot vhod, ustvari naključni skrivni ključ (ali skupno skrivnost, iz katere je mogoče ustvariti ključ) in vrne ključ/skrivnost in njeno inkapsulirano vrednost - tj. kriptogram (angl. Ciphertext). Kriptogram se pošlje nazaj imetniku zasebnega ključa (tj. entiteti, ki je prvotno ustvarila par ključev). Dekapsulacija vzame kot vhod vrednosti zasebnega ključa in kriptograma. Vrne vrednost deljene skrivnosti/ključa ali pa ne uspe (to se lahko zgodi, če so uporabljeni napačni ključi in včasih, čeprav zelo redko, se lahko zgodi tudi pri uporabi ustreznih ključev - v tem primeru je treba postopek ponoviti). Postopek delovanja algoritmov KEM je predstavljen spodaj (Slika 1).



Slika 1: **Prikaz delovanja algoritmov KEM po FIPS 2023, povzeto po [14].**

2.1 ML-KEM

ML-KEM (Module-Lattice-Based Key-Encapsulation Mechanism) je trenutno edini standardiziran post-kvantni KEM. Standardiziran je bil v FIPS 203 (Federal Information Processing Standards) [14]. ML-KEM temelji na CRYSTALIS-Kyber iz tretjega kroga tekmovanja NIST, vendar vključuje nekaj manjših sprememb (npr. fiksna velikost deljenega skrivnega ključa in manj uporabljenih zgoščevalnih operacij).

ML-KEM ima hitro generiranje ključev, inkapsulacijo in dekapsulacijo (v primerjavi z drugimi primerljivimi algoritmi), velikost javnega ključa in kriptogram pa predstavljata dobro ravnovesje med varnostjo in učinkovitostjo [19].

Varnost ML-KEM temelji na problemu modularnega učenja z napakami (angl. Module Learning with Errors - MLWE) v modularnih mrežah (angl. Modular lattice) [20]. Trenutno ni znakov, da je ta metoda vzpostavitve skupne skrivnosti nevarna v tradicionalnih ali kvantnih računalnikih. ML-KEM je zgrajen v dveh korakih. Problem MLWE se uporablja za ustvarjanje algoritma šifriranja javnega ključa (angl. Public-Key Encryption - PKE). PKE se nato pretvori z uporabo Fujisaki-Okamoto transformacije [21] v mehanizem za inkapsulacijo ključev (KEM). Nastali KEM zagotavlja varnost v skladu s splošnejšim modelom napada kot osnovni algoritem PKE, kar ima za posledico varnost IND-CCA2 [22].

ML-KEM je definiran s tremi nabori parametrov: ML-KEM-512, ML-KEM-768 in ML-KEM-1024. NIST priporoča uporabo ML-KEM-768 kot privzetega nabora parametrov, saj zagotavlja veliko varnostno maržo ob sprejemljivi hitrosti izvajanja [14]. ML-KEM je med vsemi kandidati za standardizacijo bil prepoznaven po svoji hitrosti delovanja in velikosti kriptogramov ter ključev.

Neuspeh dekapsulacije (angl. Decapsulation failure) je, ko postopek dekapsulacije ne vrne istega skrivnega ključa, kot je bil ustvarjen v inkapsulaciji, čeprav so bili vhodi pravilni in je bilo generiranje naključnosti uspešno. Verjetnost, da se to zgodi, pa je izjemno majhna. Verjetnost neuspeha dekapsulacije (angl. decapsulation failure rate - DFR) ML-KEM je [14]:

- $2^{-138,8}$ za ML-KEM-512,
- $2^{-164,8}$ za ML-KEM-768 in
- $2^{-174,8}$ za ML-KEM-1024.

ML-KEM in vsi algoritmi KEM opisani v tem prispevku uporabljajo naključne vrednosti, ustvarjene pri njihovem generiranju ključev (ali fazi pred njim). Ker funkcija generiranja ključev ne sprejema nobenih parametrov, so te naključne vrednosti tiste, ki v celoti določajo, kakšen par ključev se ustvari. Standard ML-KEM omogoča, da se dve (32 zlogov veliki) naključni vrednosti shranita kot seme namesto zasebnega ključa. Ko je zasebni ključ potreben, se lahko uporabi notranja funkcija, ki generira ključ iz semena. Ta možnost močno zmanjša količino podatkov, ki jih je treba shraniti kot zasebni/dekapsulacijski ključ (64 namesto 1632, 2400 ali 3168 zlogov, odvisno od nabora parametrov) in zagotavlja dodatno varnostno lastnost (kot bo omejeno v poglavju 3.1.2). Slaba stran shranjevanja semena namesto zasebnega ključa je dodatno procesiranje; vendar je razlika v obdelavi za razširitev zasebnega ključa ML-KEM minimalna, medtem ko je uporaba semen veliko enostavnejša. Obstajajo že pobude, da bi uporaba semen postala de-facto način implementacije ML-KEM [23]. Nekaj podobnega bi se lahko implementiralo tudi z drugimi algoritmi KEM, vendar uradno tega trenutno ne podpirajo. Drugi algoritmi imajo večje zasebne ključe in počasneje ustvarjajo pare ključev, zato bi bil kompromis med uporabo pomnilnika in hitrostjo večji.

2.2 Classic McEliece

Classic McEliece [16], [24] je konzervativna rešitev na problem, ki ga predstavlja kvantni računalnik, ker temelji na kriptosistemu McEliece, ki je bil prvič predlagan leta 1978 [25]. To pomeni, da je sistem in problem na katerem sloni algoritem imel veliko časa za dozorevanje in strokovnjaki, so imeli veliko časa, da bi v njem našli ranljivosti. Iz tega razloga, je ta algoritem tudi veljal kot favorit za standardizacijo, kot alternativna izbira algoritmu ML-KEM.

Classic McEliece KEM je zgrajen iz PKE, ki je narejen iz Niederreiterjeve dvojne različice McEliecejevega PKE z uporabo binarnih kod Goppa in preko posebej modificirane Fujisaki-Okamoto transformacije pretvorjen v KEM, da doseže dodatno učinkovitost in varnost IND-CCA2 [26], [27]. Classic McEliece je tudi v procesu ISO standardizacije [28], [29].

Obstaja več različic algoritma Classic McEliece. Tu bomo omenili tri najpomembnejše. Prva je označena kot različica pc (Plaintext Confirmation). To je bila različica Classic McEliece, ki je bila predložena v NIST tekmovanje v krogih 1 do 3 [16], [30]. V 4. krogu je bil nadomeščen z različicama f in ne-f (tj. za naborom parametrov ni oznake). Različica pc je imela daljše kriptograme in je bila varnejša, a tudi počasnejša. Različice f imajo hitrejše generiranje ključev, medtem ko imajo različice brez oznake, enostavnejše generiranje ključev [31]. Različne različice algoritma uporabljajo različne parametre in niso interoperabilne. Za različici f in ne-f Classic McEliece obstaja pet izbranih naborov parametrov (ti so predstavljeni kasneje v članku).

Glavna prednost algoritma Classic McEliece je njegova učinkovita inkapsulacija in dekapsulacija. Po drugi strani pa je generacija ključev zelo počasna, kar bi lahko predstavljalo težavo pri komunikaciji v realnem času. Druga pomanjkljivost je velikost ključev, zlasti javnega (tj. inkapsulacijskega) ključa. Glede na nabor parametrov je lahko ta do 1,3 MB velik. To bi lahko bila velika težava v omejenih okoljih z omejenimi viri (npr. vgrajene naprave). Glede na slabosti in prednosti algoritma Classic McEliece in dejstva, da so njegovi pari ključev varni za ponovno uporabo, se zdi, da je najučinkovitejši način uporabe tega algoritma ponovna uporaba para ključev in izogibanje režijskim stroškom povezanih z generiranjem ključev in pošiljanja javnega ključa.

2.3 FrodoKEM

FrodoKEM [17] je družina algoritmov, katerih varnost izhaja iz previdne parametrizacije dobro raziskanega problema učenja z napakami (LWE), ki ima tesno povezavo z domnevno težkimi problemi na generičnih, algebrsko nestrukturiranih mrežah (angl. unstructured lattices) [17]. To je nekoliko povezano z ML-KEM, vendar ML-KEM uporablja strukturirane mreže (obstaja struktura/simetrija LWE). Strukturirane mreže ponujajo boljšo učinkovitost delovanja, manjše kriptograme in manjše ključe. Čeprav ni bila dokazana nobena razlika v stopnji varnosti, se nekateri strokovnjaki bojijo, da bi lahko bile strukturirane mreže šibka točka kriptografskega sistema in zato podpirajo bolj konzervativno rešitev, ki je FrodoKEM, kot alternativo [12]. Od rešitev predstavljenih v tem prispevku sta Classic McEliece in FrodoKEM dve najbolj konzervativni (tj. najbolj zreli in najmanj verjetno, da bi vsebovali ranljivosti) rešitvi.

Kot vsi drugi KEM v tem dokumentu je FrodoKEM zgrajen iz PKE (FrodoPKE) z dokazljivo varnostjo IND-CPA, ki se pretvori, podobno kot ML-KEM, z uporabo različice transformacije Fujisaki-Okamoto (natančneje HHK), v KEM z varnostjo IND-CCA2 [32]. Za svoje delovanje FrodoKEM uporablja psevdonaključni generator za generiranje javne matrike A, ki jo je mogoče izvesti z uporabo algoritmov AES-128 [33] ali SHAKE128 [34] (to razlikovanje bo postalo pomembno pri merjenju zmogljivosti).

Od tretjega kroga tekmovanja NIST je bil FrodoKEM posodobljen in je v postopku standardizacije v reviziji ISO/IEC 18033-2 [28], [35]. Kot del najnovejše posodobitve je bil FrodoKEM razdeljen na dve različici [17]. Med obema so majhne razlike (npr. FrodoKEM uporablja sol (angl. Salt) in ima večja semena), vendar je glavna razlika med njimi, da je pri FrodoKEM par ključev mogoče prosto ponovno uporabiti, medtem ko je pri eFrodoKEM treba par ključev spremeniti po vsaj vsakih 256 uporabah, kar preprečuje določene oblike napadov [17]. Ta razdelitev na dve različici lahko povzroči zmedo, ker je bila do vključno tretjega kroga tekmovanja NIST predložena samo ena različica algoritma. Ta različica je bila poimenovana FrodoKEM, vendar se je sedaj preoblikovala v eFrodoKEM. Pri pregledovanju starejših zapisov o algoritmu FrodoKEM je zato pomembno upoštevati, da se starejše informacije (pred razdelitvijo) imenujejo FrodoKEM, vendar se sklicujejo na trenutno specifikacijo eFrodoKEM, in to, kar je zdaj FrodoKEM, pred razdelitvijo ni obstajalo.

FrodoKEM ima tri nabore parametrov in dva možna algoritmom za generiranje matrike A. Skupaj ti predstavljajo šest naborov: FrodoKEM-640-AES, FrodoKEM-976-AES, FrodoKEM-1344-AES, FrodoKEM-640-SHAKE, FrodoKEM-976-SHAKE in FrodoKEM-1344-SHAKE. Enake kombinacije obstajajo za eFrodoKEM.

Neuspeh pri dekapsulaciji je, tako kot pri vseh tukaj predstavljenih post-kvantnih algoritmih KEM, možen tudi v FrodoKEM, čeprav je zelo malo verjetno [17], tudi v primerjavi z ML-KEM (enake vrednosti veljajo za eFrodoKEM):

- $2^{-138,7}$ za FrodoKEM-640 (AES ali SHAKE),
- $2^{-199,6}$ za FrodoKEM-976 (AES ali SHAKE) in
- $2^{-252,5}$ za FrodoKEM-1344 (AES ali SHAKE).

Glavna prednost algoritma FrodoKEM je njegova navidezna robustnost, ki izhaja iz uporabe dobro uveljavljenega in razumljenega problema LWE. Po drugi strani pa ima nadpovprečno velik javni ključ in največji kriptogram in zasebni ključ od vseh predstavljenih algoritmov KEM. Hitrost delovanja je posledično slabša.

2.4 BIKE

BIKE (Bit-flipping Key Encapsulation) [18] je algoritem KEM, ki temelji na vrsti kod za popravljanje napak QC-MDPC (Quasi Cyclic Moderate Density Parity Check). Tako kot Classic McEliece temelji na sistemu McEliece [25], vendar uporablja druge kode. Kode QC-MDPC so cenjene zaradi svoje učinkovitosti pri popravljanju napak in se v veliki meri upora-

bljajo v kriptografiji (med drugim jih uporablja tudi algoritem HQC).

Kot vsi ostali opisani algoritmi KEM, tudi BIKE uporablja različico transformacije Fujisaki-Okamoto, da naredi IND-CCA2 varen KEM iz PKE [18], [36]. Vendar pa lahko BIKE doseže IND-CCA2 le, če je mogoče doseči dovolj nizko verjetnost neuspeha dekapsulacije algoritma (DFR). Čeprav se zdi, da je DFR v praksi dovolj majhen, avtorji uradno ne trdijo, da je dosežena varnost IND-CCA2. Nekatere nedavne raziskave podpirajo idejo, da bi lahko bil BIKE INDCCA2 varen (z manjšimi spremembami specifičnih parametrov algoritma) [37]. BIKE torej uradno dosega samo varnost IND-CPA. BIKE se zato lahko uporablja le z efemernimi (tj. kratkotrajnimi) ključi, kar pomeni, da se za vsako sejo ustvari nov par ključev in dekapsulacija več kot enega kriptograma (z istim parom ključev) ni dovoljena. BIKE je bil zasnovan za uporabo s sinhronimi komunikacijskimi protokoli (npr. TLS) s kratkotrajnimi ključi, kjer je potreben (in zadosten) samo IND-CPA [18]. BIKE se lahko uporablja tudi z dolgotrajnimi ali statičnimi pari ključev (čeprav tega prvotno ni podpiral), vendar morajo biti v tem primeru izpolnjene predpostavke o dekoderju, da se zagotovi IND-CCA2.

BIKE je definiran s tremi nabori parametrov, ki ustrezajo trem različnim NIST varnostnim ravnem. Običajno so označeni kot BIKE-L1 (L je za raven (angl. Level)), BIKE-L3 in BIKE-L5.

Prednost BIKE je relativno majhna velikost komunikacijskih podatkov v primerjavi z drugimi algoritmi KEM. Pri uporabi s kratkotrajnimi ključi samo ML-KEM pošlje manj podatkov po ‚žici' (ko se ključi uporabljajo dolgoročno, postane Classic McEliece učinkovitejši zaradi majhnih kriptogramov). Ko pride do hitrosti delovanja pa je primerljiv s HQC, čeprav je tipično malo počasnejši.

2.5 HQC

Hamming Quasi-Cyclic (HQC) [15] je KEM, ki temelji na strukturiranih kodah (angl. structured codes). Zaradi tega je podoben Classic McEliece in še posebej algoritmu BIKE, ker se tudi zanaša na kode QC-MDPC (Quasi Cyclic Moderate Density Parity Check). Shema šifriranja javnega ključa HQC PKE se pretvori v HQC KEM s transformacijo HHK [38] (enako kot FrodoKEM) [15]. To, tako kot pri drugih algoritmih, daje HQC IND-CCA2 varnost, ker je za razliko od BIKE pokazan dovolj majhen DFR.

HQC ponuja tri nabore parametrov, HQC-128, HQC-192 in HQC-256, poimenovane po nivojih varnosti, ki jih zagotavljajo. Po zmogljivost je HQC primerljiv z BIKE, z nekoliko večjimi velikostmi parametrov, vendar nekoliko boljšo računalniško hitrostjo.

HQC, je drugi algoritem, ki je bil pred kratkim izbran za standardizacijo s strani NIST, vendar trenutno še nima standardizirane oblike. Čeprav bo tudi HQC standardiziran, NIST primarno priporoča uporabo algoritma ML-KEM in implementacijo HQC, samo kot rezervo v primeru, da se v ML-KEM odkrijejo ranljivosti.

3 PRIMERJAVA POST-KVANTNIH ALGORITMOV KEM

Za izbiro najboljšega algoritma KEM je treba upoštevati vidika varnosti in zmogljivosti algoritmov. Pri obravnavi zmogljivosti algoritmov mislimo predvsem na računsko kompleksnost (več o tem v nadaljevnaju), vendar je velikost javnega (tj. inkapsulacijskega) ključa, zasebnega (tj. dekapsulacijskega) ključa, kriptograma in deljene skrivnosti/ključa prav tako del učinkovitosti. Ti podatki določajo, koliko podatkov bo treba shraniti na napravah in koliko podatkov bo treba prenesti. To je lahko zelo pomemben dejavnik pri izvajanju protokolov (lahko imajo omejitve velikosti paketov ali parametrov) ali pri izvajanju na napravah, ki imajo omejen pomnilnik (npr. naprave IoT) in/ali so omejene pri količini prenosa (npr. vgrajene naprave, kjer je potrebno porabiti čim manj energije za prenos podatkov). Velikost deljene skrivnosti (ta vrednost je ob prenosu vsebovana v kriptogramu in zato ne prispeva k velikosti prenesenih podatkov) lahko prav tako vpliva na odločitev o uporabi KEM. Če obstajajo zahteve glede velikosti deljene skrivnosti, ker naj bi se skrivnost uporabila, na primer, kot ključ v AES-256, bi se zahtevala 32 zlogov velika skrivnost. Čeprav se deljena skrivnost praviloma ne uporabi neposredno kot ključ (z varnostnega vidika je lahko to popolnoma sprejemljivo, če uporabljate KEM [39] in ob upoštevanju varnostnih lastnosti vezave), ampak se za pretvorbo skrivnosti v ključe in/ali druge skrivne vrednosti poljubne velikosti uporabi funkcija izpeljave ključa (angl. Key Derivation Function - KDF).

Poleg velikosti elementov algoritmov, Tabela 2 vključuje tudi informacije o varnostnem modelu, po katerem so algoritmi varni, in doseženi NIST varnostni stopnji (več o obeh v nadaljevanju).

Tabela 2: **Osnovne informacije o algoritmih KEM [40].**

KEM z naborom parametrov	Varnostni model	NIST varnostna stopnja	Velikost javnega ključa (zlogi)	Velikost zasebnega ključa (zlogi)	Velikost kriptograma (zlogi)	Velikost deljene skrivnosti (zlogi)
ML-KEM-512	IND-CCA2	1	800	1632 (ali 64 seme)	768	32
ML-KEM-768	IND-CCA2	3	1184	2400 (ali 64 seme)	1088	32
ML-KEM-1024	IND-CCA2	5	1568	3168 (ali 64 seme)	1568	32
Classic-McEliece-348864	IND-CCA2	1	261120	6492	96	32
Classic-McEliece-348864f	IND-CCA2	1	261120	6492	96	32
Classic-McEliece-460896	IND-CCA2	3	524160	13608	156	32
Classic-McEliece-460896f	IND-CCA2	3	524160	13608	156	32
Classic-McEliece-6688128	IND-CCA2	5	1044992	13932	208	32
Classic-McEliece-6688128f	IND-CCA2	5	1044992	13932	208	32
Classic-McEliece-6960119	IND-CCA2	5	1047319	13948	194	32
Classic-McEliece-6960119f	IND-CCA2	5	1047319	13948	194	32
Classic-McEliece-8192128	IND-CCA2	5	1357824	14120	208	32
Classic-McEliece-8192128f	IND-CCA2	5	1357824	14120	208	32
eFrodoKEM-640-AES ali SHAKE	IND-CCA2	1	9616	19888	9720	16
FrodoKEM-640-AES ali SHAKE	IND-CCA2	1	9616	19888	9752	16
eFrodoKEM-976-AES ali SHAKE	IND-CCA2	3	15632	31296	15744	24
FrodoKEM-976-AES ali SHAKE	IND-CCA2	3	15632	31296	15792	24
eFrodoKEM-1344-AES ali SHAKE	IND-CCA2	5	21520	43088	21632	32
FrodoKEM-1344-AES ali SHAKE	IND-CCA2	5	21520	43088	21696	32
BIKE-L1	IND-CPA	1	1541	5223	1573	32
BIKE-L3	IND-CPA	3	3083	10105	3115	32
BIKE-L5	IND-CPA	5	5122	16494	5154	32
HQC-128	IND-CCA2	1	2249	2305	4433	64
HQC-192	IND-CCA2	3	4522	4586	8978	64
HQC-256	IND-CCA2	5	7245	7317	14421	64

3.1 Varnost

V tem razdelku bodo obravnavane različne varnostne lastnosti in varnostni modeli, ki jih je mogoče uporabiti za algoritme KEM (npr. nerazločljivost kriptograma (IND), ki je že omenjena v Tabeli 2. Podrobneje bodo predstavljeni tudi NIST nivoji varnosti, ki so tudi že bili omenjeni v Tabeli 2.

3.1.1 Nerazločljivost kriptograma

Varnostni model nerazločljivosti kriptograma se uporablja za dokazovanje varnosti algoritmov šifriranja. Delujejo kot igra med izzivalcem, ki ima skrivni ključ (tj. zasebni ključ v primeru algoritmov javnega ključa), in nasprotnikom (tj. napadalcem), ki želi razbiti algoritem šifriranja. Del imena »IND« (v IND-CPA ali IND-CCA2) pomeni nerazločljivost kriptograma (angl. Ciphertext Indistinguishability) in se nanaša na nasprotnikovo nezmožnost razlikovanja med šifriranimi sporočili (ali inkapsuliranih skrivnih ključev, v primeru KEM) na podlagi kriptogramov. V igri nasprotnik pošlje dve sporočili v čistopisu (angl. Plaintext) izzivalcu, ki šifrira naključno in vrne kriptogram. Da bi bil algoritem varen (v skladu s tremi modeli, ki jih bomo kmalu predstavili), nasprotnik ne sme imeti bistveno večje možnosti kot 50 % (ker izbira med dvema sporočiloma), da pravilno ugane izvorno sporočilo. Edini način za zmago s kakršno koli doslednostjo je, da izzivalec razbije osnovni matematični problem, na katerem temelji algoritem (npr. problem mreže). Nerazločljivost kriptograma je na voljo v treh nivojih, ki v bistvu določajo, koliko ‚moči' lahko ima napadalec in še vedno ne more

razlikovati med šifriranimi sporočili. Nerazločljivost pri napadu z izbranim čistopisom (angl. Indistinguishability under chosen plaintext attack - IND-CPA) je najšibkejša med njimi.

Z IND-CPA lahko nasprotnik poleg izvajanja poljubnih operacij, šifrira (ker ima dostop do javnega ključa) katerikoli čistopis, ki ga želi in dobi veljaven kriptogram. Tudi potem, ko nasprotnik pošlje dva čistopisa in izzivalec vrne nasprotniku v ugibanje kriptogram enega izmed čistopisov, lahko nasprotnik še vedno šifrira poljubna sporočila. Ideja te igre je dokazati, da napadalec, ki vidi šifrirana sporočila (tj. kriptograme), kar je normalno, iz njih ne more pridobiti nobenih pomembnih informacij. IND-CPA vključuje tudi naključnost šifriranja, sicer bi nasprotnik šifriral čistopisa, poslana izzivalcu, in bi lahko primerjal rezultate z vrnjenim kriptogramom.

Nerazločljivost pri (neprilagodljivem) napadu z izbranim kriptogramom (angl. Indistinguishability under (non-adaptive) chosen ciphertext attack - IND-CCA1) daje nasprotniku vse možnosti iz IND-CPA in doda možnost, da izzivalec dešifrira kriptograme, ki jih je ustvaril. Ko nasprotnik pošlje dva čistopisa izzivalcu, nima več dostopa do možnosti dešifriranja. Nerazločljivost pri napadu s prilagodljivim izbranim kriptogramom (angl. Indistinguishability under adaptive chosen ciphertext attack - IND-CCA2) je enaka kot IND-CCA1 z dodatkom, da lahko nasprotnik dešifrira poljubne kriptograme tudi po prejemu odgovora na izziv (edina izjema je kriptogram vrnjen v izzivu). Ta raven varnosti dokazuje, da tudi če lahko nasprotnik nekoga prelisiči, da dešifrira poljubno število šifriranih sporočil, nasprotniku ne bo dala prednosti in pokaže, da so šifrirana besedila zaščitena pred posegi (sicer bi napadalec lahko predvidljivo spremenil kriptogram izziva in ga dešifriral). V kriptosistemih z IND-CCA2 nivojem varnosti se lahko par ključev obravnava kot dolgoročen in ponovno uporabi [41]. Sicer to na splošno ni najboljša praksa, če se ji je mogoče izogniti. Da bi dosegli popolno prihodnjo varnost (angl. Perfect Forward Secrecy), je treba zasebni ključ izbrisati po dekapsulaciji deljene skrivnosti [42]. Ključe, ki so namenjeni enkratni uporabi imenujemo efemeralni ključi (angl. Ephemeral key). IND-CCA2 vključuje varnostne značilnosti IND-CPA in IND-CCA1, IND-CCA1 pa vključuje lastnosti IND-CPA. [43]

Skoraj vsi post-kvantni algoritmi KEM, pregledani v tem prispevku (glej Tabelo 2) dosežejo varnost IND-CCA2 (pogosto imenovano samo IND-CCA), ki je bila določena tudi v merilih NIST za ocenjevanje varnosti predlaganih rešitev. Rahlo odstopanje je shema BIKE, ki ima dokazano samo varnost IND--CPA; vendar pa bi lahko algoritem dosegel tudi IND-CCA2 (kot je bilo obravnavano v sekciji o algoritmu BIKE).

3.1.2 Varnostne lastnosti vezave

KEM so razmeroma nova oblika algoritmov, vsaj v praksi. Tradicionalno se za izmenjavo ključa uporabljajo konstrukti, ki temeljijo na Diffie-Hellman algoritmu, vendar ti niso primerni za uporabo v post--kvantnem svetu. Zato jih nadomeščamo z algoritmi KEM. Pri prehodu iz tradicionalnih na nove algoritme so bile prenesene tudi varnostne lastnosti (tj. nerazločljivost kriptograma). Novi algoritmi KEM so zato bili zasnovani tako, da izpolnjujejo in dokazujejo le to varnostno lastnost [44]. Na žalost se je izkazalo, da samo ta lastnost morda ne bo dovolj, kot so pokazali napadi s ponovno inkapsulacijo (angl. re--encapsulation attacks).

Napad s ponovno inkapsulacijo [45] izkorišča dejstvo, da je v nekaterih KEM mogoče dekapsulirati kriptogram v deljeno skrivnost/ključ in nato ustvariti drug kriptogram za drug javni ključ, ki se dekapsulira v isto deljeno skrivnost. Posledica tega je, da dve entiteti izračunata isto skupno skrivnost, čeprav ena od njih misli, da komunicira z nasprotnikom. To ni v nasprotju s stopnjo varnosti IND-CCA2.

Raziskovalci [45] so formalizirali nove varnostne lastnosti »vezave« za algoritme KEM. Da bi en kriptografski element (npr. deljena skrivnost/ključ) vezal drugega (npr. javni ključ), mora prvi kriptografski element enolično določiti drugi element (tj. med vrednostmi drugega elementa ni trkov). V primeru napad s ponovno inkapsulacijo mora skrivnost v skupni rabi vezati javni ključ (tj. deljena skrivnost/ključ lahko nastane samo iz določenega javnega ključa) in deljena skrivnost mora vezati kriptogram (tj. dva različna kriptograma se ne moreta dekapsulirati v isto deljeno skrivnost/ključ), da napad ni mogoč. Skupaj obstaja šest veljavnih veznih parov med deljeno skrivnostjo/ključem K, kriptogramom CT in javnim ključem PK. Javni ključ PK ne more vezati kriptograma CT ali deljen skrivnosti K, ker bi to pomenilo, da bi bili kriptogrami in deljene skrivnosti vedno enake. Vezi so lahko tudi med več kriptografskimi elementi. Šest veljavnih vezav vključuje:

- X-BIND-K-CT: Deljena skrivnost/ključ K veže kriptogram CT.
- X-BIND-K-PK: Deljena skrivnost/ključ K veže javni ključ PK.
- X-BIND-CT-K: Kriptogram CT veže deljeno skrivnost/ključ K.
- X-BIND-CT-PK: Kriptogram CT veže javni ključ PK.
- X-BIND-K,CT-PK: Deljena skrivnost/ključ K in kriptogram CT skupaj vežeta javni ključ PK.
- X-BIND-K,PK-CT: Deljena skrivnost/ključ K in javni ključ PK skupaj vežeta kriptogram CT.

Veliko algoritmov KEM implicitno zavrača (angl. implicitly rejecting), kar pomeni, da bo algoritem vedno vrnil vrednost (namesto napake) tudi, ko dekapsulacija ne uspe. To velja za vse algoritme KEM, ki uporabljajo Fujisaki-Okamotovo transformacijo, ki je uporabljena v vseh algoritmih KEM obravnavanih v tem prispevku. Implicitna zavrnitev pomeni, da bo vsako šifrirano besedilo sprejeto, zato vezanje kriptograma na deljeno skrivnost ali javni ključ ni mogoče. To pomeni, da algoritmi z implicitnim zavračanjem ne morejo doseči X-BIND-CT-K ali X-BIND-CT-PK.

X v zapisu je označba mesta za različne varnostne modele nasprotnikov: pošten (angl. Honest - HON), uhajanje (angl. Leak - LEAK) in zlonamerni (angl. Malicious - MAL). V poštenem scenariju pare ključev ustvari funkcija za generiranje ključev KEM. V scenariju uhajanja so ključi tudi pošteno ustvarjeni, nato pa uhajajo nasprotniku (tj. nasprotnik pridobi zasebni ključ in ima možnost dekapsulacije). V zlonamernem scenariju lahko nasprotnik sam ustvari veljavne pare ključev. Višji varnostni model vključuje nižje (npr. varnost MAL-BIND-K-CT vključuje varnost LEAK in HON-BIND-K-CT).

Tabela 3 predstavlja varnostne lastnosti vezav za vseh pet algoritmov KEM obravnavanih v tem članku. Ker vsi implicitno zavračajo algoritme, nobeden od njih ne doseže HON-BIND-CT-K ali HON-BIND--CT-PK (zato te lastnosti niso vključene v tabelo).

Kot je razvidno iz Tabele 3 nobeden od algoritmov ne doseže vseh varnostnih lastnosti vezave. Vendar pa obstajajo nekatere olajševalne okoliščine. Prvič, ni popolnoma jasno, kako pomembne so te varnostne lastnosti za protokole v realni uporabi. Predpostavka za te varnostne lastnosti je, da lahko napadalec manipulira z javnim ključem (ki se vrne skupaj s kriptogramom za dekapsulacijo). To bi povzročilo različne rezultate dekapsulacije in neuspešno izmenjavo. Vendar pa se javni ključi, ki jih posreduje napadalec, običajno ne uporabljajo; namesto tega, ko je za dekapsulacijo potreben javni ključ, se uporabi shranjena vrednost ključa. Predpostavka postane ponovno relevantna, če lahko napadalec nekako manipulira s shranjeno vrednostjo, kar pa bi pomenilo, da ima napadalec že zelo napreden dostop, ki bi ga lahko izkoristil za številne napade. Drugič, obstaja zelo enostavna rešitev za vezanje kriptografskih elementov, ki pa niso dela algoritma KEM. Funkcije izpeljave ključev (KDF) se uporabljajo za razširitev deljene skrivnosti na ustrezno dolžino (npr. HKDF v TLS) in za združevanje več deljenih skrivnosti (npr. v hibridnih KEM). Funkcija KDF se lahko uporabi

Tabela 3: **Varnostne lastnosti vezav za ML-KEM, Classic McEliece, FrodoKEM, BIKE in HQC [46], [47].**

Zavezujoč pojem	ML-KEM	Classic McEliece	FrodoKEM	BIKE	HQC
Hon-Bind-K-PK	✓	×	✓	✓	✓
LEAK-BIND-K-PK	✓	×	✓	×	×
MAL-BIND-K-PK	×	×	×	×	×
Hon-Bind-K, Ct-PK	✓	×	✓	✓	✓
LEAK-BIND-K, CT-PK	✓	×	✓	×	×
MAL-BIND-K, CT-PK	×	×	×	×	×
HON-BIND-K-CT	✓	✓	✓	✓	✓
LEAK-BIND-K-CT	✓	✓	✓	✓	×
MAL-BIND-K-CT	×*	✓	×	✓	×
HON-BIND-K, PK-CT	✓	✓	✓	✓	✓
LEAK-BIND-K, PK-CT	✓	✓	✓	✓	×
MAL-BIND-K, PK-CT	✓	✓	×	✓	×

za povezovanje elementov. Na primer, z dodajanjem javnega ključa kot vhoda v funkcijo KDF, shema postane LEAK-BIND-K-PK, ker je končna deljena skrivnost zdaj vezana na javni ključ. Varnostne lastnosti vezave se lahko dodajo tudi preko protokola, ki obdaja KEM. Alternativno je X-BIND-K-PK mogoče doseči tudi z uporabo robustnega PKE in vključitvijo kriptograma v KDF, ki bo deljeno skrivnost vezal z javnim ključem [47]. To je uporabno tudi zato, ker so javni ključi v post-kvantno varnih algoritmih KEM lahko zelo veliki in posledično njihova vključitev v KDF ni učinkovita.

Če nobeden od teh blažilnih dejavnikov ne zadostuje, je pri uporabi teh algoritmov potrebno biti pozoren, da se ne slepo zanašamo na vsebovanost teh lastnosti. Prav tako se zdi, da ne obstaja noben protokol (v uporabi ali hipotetični protokol), kjer bi izkoriščanje pomanjkljivosti varnostnih lastnosti vezave zlonamernemu napadalcu (MAL) prineslo kritično prednost, ker je napadalec v takšni situaciji (kjer lahko manipulira s pari ključev) preprosto že premočan.

Določene pomanjkljivosti algoritma ML-KEM izhajajo iz uporabljenega načina serializacije (angl. serialization). Čeprav so varnostne lastnosti vezave precej novi, so bili znani v času, ko je NIST standardiziral ML-KEM in rešitev je bila na voljo. Rešitev je bila preprosto shraniti zasebni ključ v obliki semena in ga razširiti vsakič, ko je ključ potreben [48]. NIST je delno prisluhnil in podprl shranjevanje ključev ML-KEM v obliki dveh semen, zaradi česar je ML-KEM MAL-BIND-K-CT-varen (če se uporabljajo semena – zato * v Tabeli 3), vendar ne doseže MAL-BIND-K-PK. To je eden od razlogov, zakaj mnogi raje shranjujejo zasebni ključ kot seme (64 zlogov, ki združuje dve semeni vsako velikosti 32 zlogov). To se odraža tudi v predlaganih implementacijah protokolov z algoritmom ML-KEM, kjer se pogosto uporabljajo semena namesto razširjenih ključev. MAL-BIND-K-PK bi tudi lahko bil dosežen, če bi uporabili samo eno seme, kjer bi izračun napačno oblikovanega zasebnega ključa postal nemogoč, vendar bi to zahtevalo nekaj manjših sprememb algoritma. Zanimivo je, da je bil Kyber, predhodnik ML-KEM, varen za obe varnostni lastnosti, vendar ju je izgubil v preobrazbi v ML-KEM zaradi osredotočanja na učinkovitost algoritma.

3.1.3 NIST varnostne ravni

Tabela 4 vsebuje pet nivojev varnosti, ki jih je definiral NIST [49] za namene natečaja za izbiro post-kvantno varnih algoritmov in ki se danes lahko uporabljajo kot način primerjave varnostne moči posameznih algoritmov oz. njihovih naborov parametrov. Ravni so povezane z varnostjo obstoječih (in dobro preizkušenih) kriptografskih algoritmov. Vseh pet stopenj moči se danes šteje za varnih, vendar se to lahko v prihodnosti spremeni. S povezovanjem ravni varnosti z obstoječimi algoritmi z dobro uveljavljenimi stopnjami varnosti je enostavno prepoznati ravni, če oz. ko postanejo prešibke. Na primer, ko bodo simetrične šifre s 128-bitnimi ključi (npr. AES-128) prešibke za uporabo, to pomeni, da se rešitve PQC z NIST varnostno stopnjo moči 1 ne smejo več uporabljati.

V Tabeli 2 lahko opazite, da algoritmi podpirajo samo NIST varnostne ravni 1, 3 in 5, ki ustrezajo varnostnim nivojem treh velikosti ključev AES, ki so že ustaljeno merilo za raven varnosti. Raven 1 je najboljša možnost za tiste, ki želijo najhitrejšo in najučinkovitejšo rešitev, ki naj bi ostala varna še vsaj nekaj časa. To je optimalna izbira, če je hitrost najpomembnejši izbirni parameter, če imajo naprave omejeno moč, če obstajajo omejitve v količini podatkov, ki jih je mogoče prenašati, ali če je ta raven varnosti dovolj dobra. Raven 5 je popolno nasprotje in daje prednost varnosti pred vsem drugim, predvsem na račun učinkovitosti (časa, moči, pasovne širine in pomnilnika). Raven 3 je zlata sredina in je namenjena za situacije, ko želimo dobra varnostna zagotovila, vendar ni-

Tabela 4: **NIST varnostne ravni [49]**

NIST varnostne ravni	Varnostni opis	Post-kvantni nivo varnosti
1	Vsaj tako težko za razbiti kot AES128 (izčrpno iskanje ključev)	64-bitov
2	Vsaj tako težko za razbiti kot SHA256 (iskanje trkov)	85-bitov
3	Vsaj tako težko za razbiti kot AES192 (izčrpno iskanje ključev)	96-bitov
4	Vsaj tako težko za razbiti kot SHA384 (iskanje trkov)	128-bitov
5	Vsaj tako težko za razbiti kot AES256 (izčrpno iskanje ključev)	128-bitov

smo za to pripravljeni žrtvovati učinkovitosti. Višji nivo varnosti ne zagotavlja le večje odpornosti proti kvantnim računalnikom, temveč daje tudi višja zagotovila pred morebitnimi prihodnjimi ranljivostmi v algoritmih in/ali problemih, na katerih temeljijo (npr. učenje z napakami na mreži). Upoštevajte, da večja varnostna moč (tj. večji parametri) ne nujno izničijo posledic novih ranljivosti.

Post-kvantni nivo varnosti v Tabeli 4 je izračunan na podlagi Groverjevega algoritma [50] (za algoritem AES) in algoritma Brassard et al. [51] (za algoritem SHA). Čeprav 64-bitna varnost (tj. 2^{64} poizkusov za razbitje) danes ne zveni kot dovolj (glede, na to da so minimalni uporabni ključi velikosti 128 bitov), je pomembno vedeti, da Groverjev algoritem ne omogoča vzporednega izvajanja (t.i. paralelizacije) kot jo omogočajo tradicionalni računalniki [52], [53]. Da bi dosegli zmanjšanje računskega časa za S, je potrebnih S^2 kvantnih procesorjev [54]. Na primer, milijarda (10^9) kvantnih računalnikov bi še vedno morala izvesti 2^{49} ponovitev (zmanjšano iz 2^{64}), da bi razbili AES-128. Zato bi morali biti tudi algoritmi na varnostni ravni 1 še kar nekaj časa varni.

Za konec naj omenimo še, da je, ko gledamo na varnost v celotni komunikacijski seji, na splošno dobra ideja ohraniti enako raven varnosti za vse sodelujoče komponente. Utemeljitev je, da združevanje zelo močnih in zelo šibkih komponent ne daje dobrih rezultatov, ker je celoten sistem le tako močan kot njegov najšibkejši člen, in neproporcionalno varnejše komponente le upočasnijo celoten proces. Varnostne ravni vključenih kriptografskih gradnikov bi v idealnem primeru morale biti primerljive. Primer tega je mogoče videti pri TLS, kjer se algoritmi s primerljivimi stopnjami varnosti uporabljajo skupaj (npr. X25519, AES-128 in SHA256, ki imajo vsi 128-bitno varnost za tradicionalne računalnike). Zato nam NIST varnostne ravni služijo tudi kot vodilo za združevanje algoritmov KEM in njihovih naborov parametrov z drugimi kriptografskimi gradniki (npr. simetričnimi šiframi in podpisnimi algoritmi).

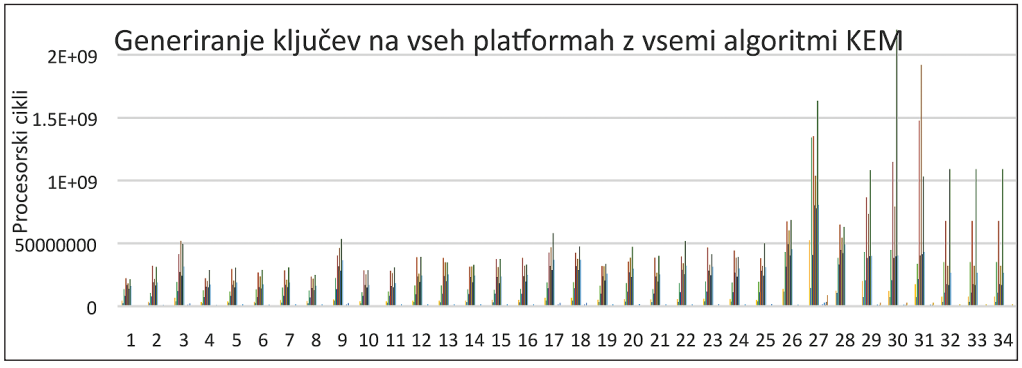
3.2 Hitrost delovanja post-kvantnih algoritmov KEM

To poglavje je namenjeno učinkovitosti delovanja post-kvantnih algoritmov KEM. Zanimala nas bo računalniška zmogljivost generiranja ključev, inkapsulacije in dekapsulacije. Podatki so bili zbrani iz dveh virov: eBATS (ECRYPT Benchmarking of Asymmetric Systems) in podatkov projekta OQS (Open Quantum Safe). V analizo so bili vključeni samo rezultati, ki so bili ustvarjeni za vse opredeljene algoritme na istih platformah.

eBATS [55] je projekt namenjen merjenju učinkovitosti kriptografskih sistemov z javnim ključem. Projekt izgleda, kot da še vedno zbira meritve, vendar algoritmi že nekaj časa niso bili posodobljeni. Za nekatere algoritme KEM je težko oceniti kako stare so že implementacije. Zdi se, da je najstarejša izvedba za BIKE, ki je iz maja 2020 (tik preden je NIST objavil tretji krog tekmovanja) in vključuje samo nabore parametrov L1 in L3 (takrat BIKE ni imel L5) [56]. Glede na opise je uporabljena različica algoritma FrodoKEM-a tudi iz drugega kroga tekmovanja NIST. Čeprav so to nekoliko starejše različice oz. implementacije teh algoritmov, se stanje v novejših različicah ne spreminja drastično, tako da bi se vsaj razmerja med algoritmi moralo ohraniti, čeprav uporabljamo starejše meritve. Ostali algoritmi imajo novejše implementacije (Classic McEliece ima različice pc, f in ne-f, kar nakazuje da gre za trenutne različice algoritmov, HQC je različica 4. kroga, Kyber pa iz 3. kroga, po čemer je bil standardiziran). ML-KEM ni podprt in namesto njega bomo uporabili Kyber. FrodoKEM nima dveh različic, zato bodo rezultati predstavljeni po novem poimenovanju kot eFrodoKEM. eBATS zbira podatke z velikega števila platform[1], vendar smo se omejili samo na tiste, ki so merili najnovejšo različico vseh algoritmov, vključenih v to primerjavo. Čeprav so v eBATS vključene različne platforme, so po odstranitvi procesorjev, ki nimajo meritev za najnovejše izvedbe vseh algoritmov KEM[2], ostale le platforme amd64. Iz eBATS je bilo zbranih podatkov za 25 platform. Rezultati so podani v procesorskih ciklih za generiranje ključev, inkapsulacijo in dekapsulacijo za 25, 50 in 75 percentilov. Uporabili bomo 50 %, kar predstavlja mediano potrebnih ciklov. To je tudi najbolj primerljiva vrednost z OQS rezultati.

Projekt Open Quantum Safe (OQS) [57] je odprtokodni projekt, katerega cilj je podpreti prehod na post-kvantno kriptografijo. Projekt je ustvaril liboqs, ki je odprtokodna knjižnica v programskem jeziku C za kvantno varne kriptografske algoritme. Na žalost njihov primerjalni projekt, ki meri uspešnost njihove knjižnice, ni več aktiven. Njihove zadnje meritve so iz aprila 2024[3]. ML-KEM takrat ni obstajal,

[1] https://bench.cr.yp.to/results-kem.html
[2] https://bench.cr.yp.to/primitives-kem.html
[3] https://openquantumsafe.org/benchmarking/visualization/speed_kem.html



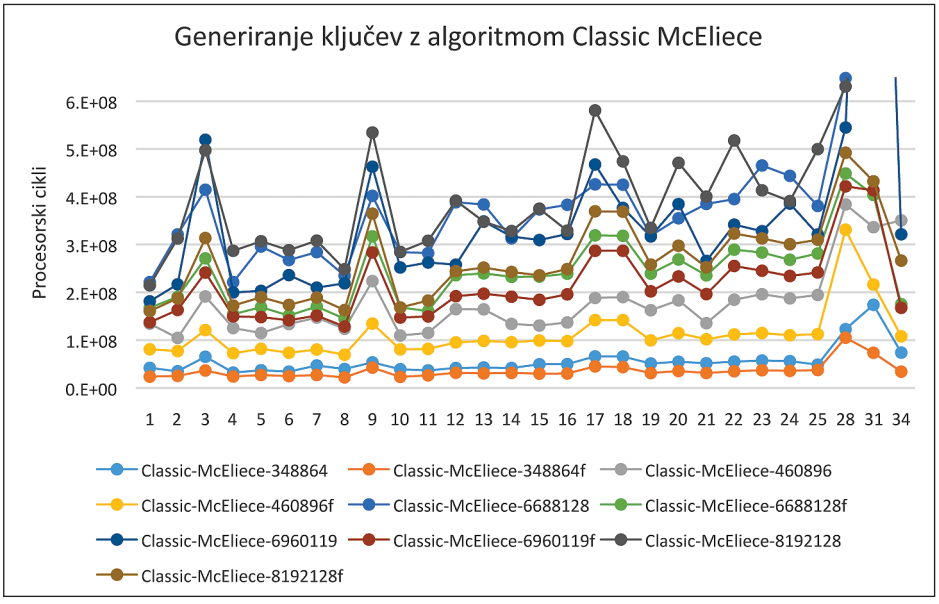
Slika 2: **Hitrost generiranja ključev za vse algoritme KEM na vseh platformah**

zato bomo namesto njega vključili Kyber (čeprav je ML-KEM podprt v novejših različicah knjižnice). Pri prehodu iz Kyberja v ML-KEM je NIST naredil spremembe na algoritmu, zaradi katerih je ML-KEM še hitrejši, kot je bil Kyber. Meritve vključujejo vse obravnavane algoritme KEM, čeprav knjižnica ne podpira obeh različic algoritma FrodoKEM (samo različico iz tretjega kroga NIST tekmovanja). Zato bomo rezultate, poročane za FrodoKEM, spremenili v eFrodoKEM, da bodo odražali spremembo imena. Merila vključujejo tri različne platforme (x86_64, ki je enaka amd64, aarch64 in M1) in tri različne izvedbe algoritmov. Referenčne implementacije so implementirane brez uporabe pospešenih funkcij CPE-ja (Centralno Procesna Enota) in so zato bistveno počasnejše od drugih dveh. Ostali dve implementaciji se razlikujeta glede na to, ali sta prenosni ali ne. To daje devet kombinacij platform in implementacij, vendar nekaterim zaradi njihove neučinkovitosti ne bomo posvečali veliko pozornosti.

Na Sliki 2 je prikazana hitrost generiranja ključev za vse algoritme KEM na vseh 34 platformah (os x), izraženo v procesorskih ciklih (os y). Ta slika ni namenjena predstavi hitrosti delovanja, ampak samo prikazu razlik v delovanju med posameznimi platformami oz. implementacijami. Na sliki lahko hitro vidimo nekaj osamelcev na desni strani grafa. Rezultat označen s številom 27 je referenčna implementacija v OQS, rezultati #29-31 so aarch64 (tj. mobilni procesor), zadnje tri (platforme #32-34) pa so procesor Apple M1. V zadnjih treh kombinacijah M1 platforme in različnih implementacij algoritmov ni razlik, zato v nadaljevanju ne bomo več vključevali vseh treh. Rezultati so podobni, če ne celo slabši, za dekapsulacijo, vendar je presenetljivo, da sta aarch64 in M1 veliko boljša v inkapsulaciji, kjer je M1 najhitrejša platforma, aarch64 pa je prav tako dober, če ne boljši od x86-64. V nadaljevanju bomo izpustili nekatere najslabše rezultate. Specifično so to referenčne implementacije (#27, #30 in #33), ker to niso implementacije, ki bi ji jim bila pomembna hitrost delovanja in prenosne različice implementacij (#26, #29 in #32), ker v povprečju dajejo slabše rezultate kot neprenosne različice.

Hitrost generiranja ključev je na dveh ločenih grafih za Classic McEliece (Slika 3) in ostale algoritme (Slika 4), ker so razlike v zmogljivosti tako velike. Classic McEliece je najpočasnejši algoritem pri generiranju ključev na vseh varnostnih ravneh. Slika 3 prikazuje procesorske cikle, ki so potrebni za generiranje ključev v vseh desetih naborih parametrov Classic McEliece (pet naborov za dve različici z in brez »f« oznake). Hitrost delovanja je zelo konsistentna med nabori parametrov, zlasti v primeru različice f. Rezultati različice, ki ni f imajo visoko varianco na določenih platformah, ki pojasnjuje prepletanje podatkovnih točk na vrhu grafa (trije najpočasnejši nabori parametrov so vse različice, ki niso f in imajo izrazito neoptimalno delovanje na arhitekturi aarch64 - platforma #31). Različice f so tudi konsistentno boljše od različic, ki niso f (pri vseh naborih parametrov). Meritve iz OQS imajo bistveno slabše rezultate za platformo x86_64 (#28) in še posebej za aarch64 (#31).

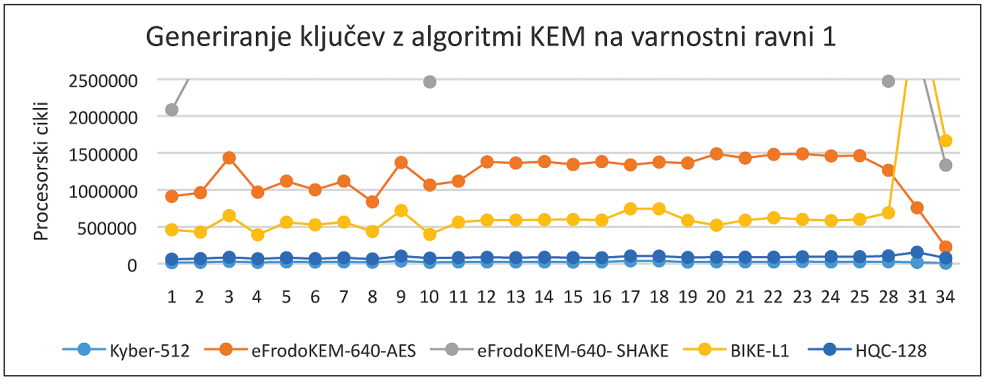
Hitrost generiranja ključev za preostale algoritme KEM z nabori parametrov na varnostni ravni 1 je predstavljena na Sliki 4. Najpočasnejši algoritem (razen Classic McEliece, ki je bil predstavljen ločeno) je FrodoKEM. Implementacija s SHAKE je tako počasna, da je le delno vidna na grafu. Implementacija AES je veliko hitrejša in še posebej učinkovita na platformah aarch64 in M1, kjer algoritem postane skoraj tako hiter kot najuspešnejši algoritmi HQC in Kyber.



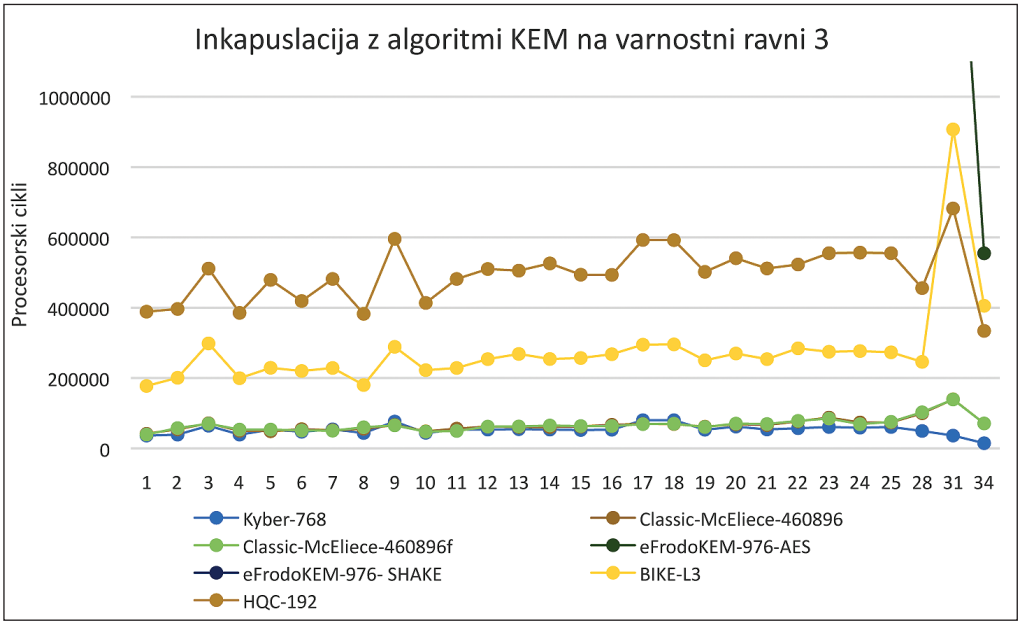
Slika 3: **Hitrost generiranja ključev z algoritmom Classic McEliece.**

BIKE je tretji najhitrejši na x86_64 platformah (#1-28), vendar očitno ni dobro optimiziran za aarch64 ali M1. Kyber je najhitrejši pri generiranju ključev, HQC pa je na drugem mestu. Čeprav Slika 4 predstavlja samo nabore parametrov algoritmov KEM z varnostno ravnjo 1, so razmerja med algoritmi enaka tudi za ravni 3 in 5, kot bo prikazano v tabeli na koncu tega poglavja.

Hitrost inkapsulacije je prikazana na Sliki 5. Na njej lahko vidite število procesorskih ciklov, ki jih vsak algoritem KEM z naborom parametrov na varnostni ravni 3 zahteva za ustvarjanje šifriranega besedila. Najslabši algoritem je eFrodoKEM. eFrodoKEM-SHAKE je tako počasen, da ni viden v grafu. eFrodoKEM-AES je skoraj trikrat hitrejši, vendar še vedno zelo počasen. Na Sliki 5 je eFrodoKEM-AES viden le na skrajni desni strani grafa (platforma #34), kjer optimizacija na platformi M1 omogočajo boljše delovanje. Sledita mu HQC in BIKE, ki imata očitno težave z optimizacijo na mobilnih platformah, sicer pa delujeta šest do dvanajstkrat hitreje kot eFrodoKEM-AES. Dva najuspešnejša algoritma sta Classic McEliece in Kyber. Kyber je le malo hitrejši (in z ML-KEM optimizacijo, bi se morala ta razlika le še okrepiti). Rezultati Classic McEliece različice f in ne-f se bistveno ne razlikujejo. Rezultati oz. razmerja med učinkovitostjo algoritmov se ponovno signifikantno ne razlikujejo, med različnimi nivoji varnosti. Classic McEliece ima več naborov parametrov z varnostno ravnjo 5 in vsi imajo primerljivo hitro delovanje.

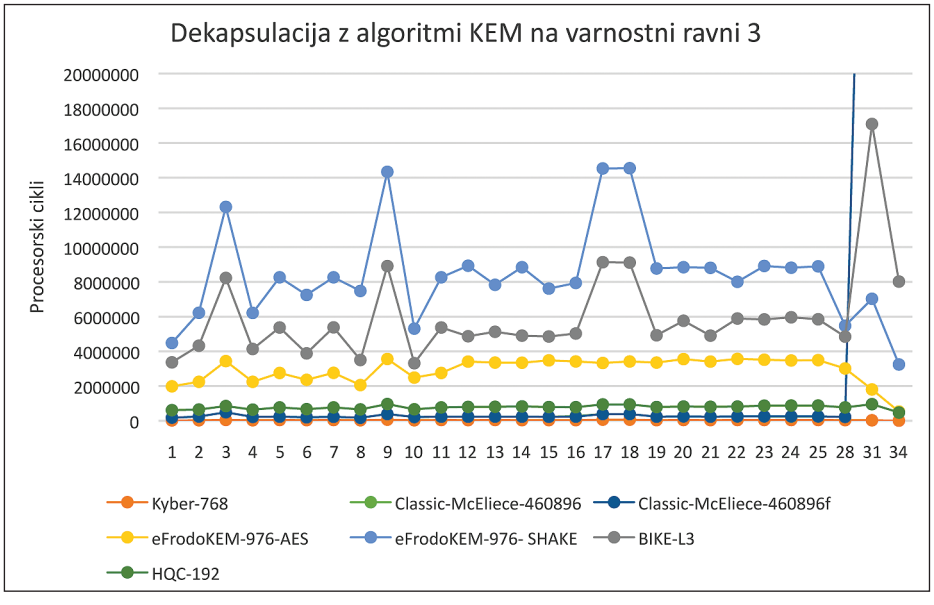


Slika 4: **Hitrost generiranja ključev z algoritmi KEM.**



Slika 5: **Hitrost inkapsulacije z algoritmi KEM.**

Na zadnje imamo na Sliki 6 prikazano še hitrost operacije dekapsulacije. Graf prikazuje povprečno število potrebnih CPE ciklov za dekapsulacijo šifriranega besedila z algoritmi KEM in nabori parametrov za varnostno raven 3. Kot pri ostalih operacijah ostaja razmerje učinkovitosti med algoritmi konsistentno med vsemi tremi nivoji varnosti. eFrodoKEM-SHAKE je spet najslabši. Zanimivo je, da je drugi najpočasnejši BIKE, in šele zatem je eFrodoKEM-AES. Čeprav se zdi, kot da so trije najhitrejši algoritmi za izvajanje dekapsulacije primerljivo hitro so med njimi precej velike razlike. Najučinkovitejši je ponovno Kyber. Classic McEliece je drugi najboljši (različici f in ne-f sta praktično identični), vendar skoraj šestkrat počasnejši od algoritma Kyber (na x86_64 platformah), HQC pa je tretji najhitrejši, a še dodatno trikrat počasnejši od Classic McEliece (na x86_64 platformah). Classic McEliece in BIKE sta bili posebej počasni na



Slika 6: **Hitrost dekapsulacije z algoritmi KEM.**

platformah aarch64 in M1. Možno je, da uporabljeni implementaciji nista bili optimizirani za takšne platforme. Po drugi strani pa je AES očitno zelo dobro optimiziran na teh platformah (zlasti za M1). Classic McEliece nabori parametrov z varnostno ravnjo 5 ponovno dosegajo skoraj identične rezultate.

Na zadnje so tu predstavljene še razlike v zmogljivosti med vsemi algoritmi KEM. Za to primerjavo so algoritmi razdeljeni v kategorije glede na njihovo raven varnosti, tako da se primerjajo samo algoritmi z nabori parametrov na enakem varnostnem nivoju. Rezultati so pokazali, da se implementacije na platformah aarch64 in M1 obnašajo bolj nepredvidljivo. Zato so v končno primerjavo bili vključeni le rezultati platforme x86_64 (tj. platforme #1-26 in #28). Za rezultate algoritma Classic McEliece z varnostno ravnjo 5, so bile uporabljene vrednosti povprečnih procesorskih ciklov vseh treh naborov skupaj, ker delujejo primerljivo hitro. Classic McEliece različici f in non-f sta vztrajno delovali zelo podobno, vendar je bila različica f vedno primerljiva ali boljša. Zato je v končno tabelo rezultatov vključena samo različica f (ker ni razloga, da ne bi uporabili različice f). Enako ne velja za eFrodoKEM, kjer so razlike med različicama AES in SHAKE precejšnje (potencialno bi bil SHAKE veliko bolj primerljiv, če platforme ne bi uporabljale strojno podprtega AES). Zaradi velikih razlik sta v končnih rezultatih različici predstavljeni kot ločena algoritma.

Končna primerjava je predstavljena v Tabeli 5. Tabela je razdeljena na generiranje ključev, inkapsulacijo in dekapsulacijo. Vsaka od treh operacij je nadalje razdeljena na tri varnostne ravni. Za vsakega od algoritmov KEM je poročano povprečno število procesorskih ciklov, potrebnih za izvedbo operacije. Poleg ciklov je delta, ki predstavlja množitelja med številom ciklov algoritma in naslednjim boljšim algoritmom. To pomaga določiti ne le vrstnega reda algoritmov po učinkovitosti, ampak tudi dobro pokazati, kako velike so razlike med njimi. Čeprav se razlike

Tabela 5: **Hitrost algoritmov KEM, ločenih glede na raven varnosti in z delto, ki prikazuje multiplikator do hitrosti prejšnjega algoritma.**

	Generiranje ključev					
	Varnostna raven 1		**Varnostna raven 3**		**Varnostna raven 5**	
KEM	**povp. cikl.**	**Δ**	**Avg. cy.**	**Δ**	**Avg. cy.**	**Δ**
Kyber / ML-KEM	24891	/	42592	/	60343	/
HQC	85176	3,42	200393	4,70	407027	6,75
BIKE	577157	6,78	1633767	8,15	4502751	11,06
eFrodoKEM (AES)	1268476	2,20	2534721	1,55	4286058	0,95
eFrodoKEM (SHAKE)	3842852	3,03	8221838	3,24	14380603	3,36
Classic McEliece (f različice)	37366143	9,72	116149661	14,13	243574748	16,94
	Inkapsulacija					
Kyber / ML-KEM	36033	/	55116	/	78205	/
Classic McEliece (f različice)	29702	0,82	65087	1,18	116681	1,49
BIKE	106749	3,59	249784	3,84	500944	4,29
HQC	216921	2,03	492785	1,97	950774	1,90
eFrodoKEM (AES)	1687015	7,78	3211743	6,52	5333815	5,61
eFrodoKEM (SHAKE)	4149954	2,46	8711655	2,71	15171487	2,84
	Dekapsulacija					
Kyber / ML-KEM	29157	/	45198	/	65767	/
Classic McEliece (f različice)	125742	4,31	261470	5,78	300710	4,57
HQC	378164	3,01	791014	3,03	1560939	5,19
eFrodoKEM (AES)	1615383	4,27	3074454	3,89	5171348	3,31
BIKE	1724035	1,07	5450595	1,77	11126235	2,15
eFrodoKEM (SHAKE)	4080854	2,37	8615275	1,58	14996451	1,35

ne zdijo tako velike, je potrebno upoštevati multiplikativno razmerje med različnimi mesti. Na primer, Kyber-512 (tj. Kyber z varnostno ravnjo 1) je najhitrejši pri generiranju ključev, medtem ko je Classic-McEliece-348864f (na zadnjem mestu) v povprečju 1500-krat počasnejši.

Rezultati za vsako operacijo so med algoritmi KEM zelo konsistentni, z dvema izjemama. BIKE-L5 prehiti eFrodoKEM-AES, pri generiranju ključev na varnostni ravni 5 (razlog je najverjetneje manjše število meritev). Drugi presenetljiv rezultat je hitrejša inkapsulacija Classic-McEliece-348864f kot Kyber-512 na varnostni ravni 1. To je edina varnostna raven, na kateri se to zgodi, in glede na izboljšave ML-KEM v primerjavi s Kyber ni jasno, ali razlika še vedno ostaja.

Končna razvrstitev algoritmov KEM, ki temelji hitrosti njihovega delovanja pri generiranju, inkapsulaciji in dekapsulaciji ključev, je predstavljena v Tabeli 6.

Tabela 6: **Razvrstitev KEM na podlagi učinkovitosti generiranja, inkapsulacije in dekapsulacije ključev.**

#	Generiranje ključev	Inkapsulacija	Dekapsulacija
1	ML-KEM / Kyber	ML-KEM / Kyber	ML-KEM / Kyber
2	HQC	Classic McEliece	Classic McEliece
3	BIKE	BIKE	HQC
4	FrodoKEM (AES)	HQC	FrodoKEM (AES)
5	Classic McEliece	FrodoKEM (AES)	BIKE

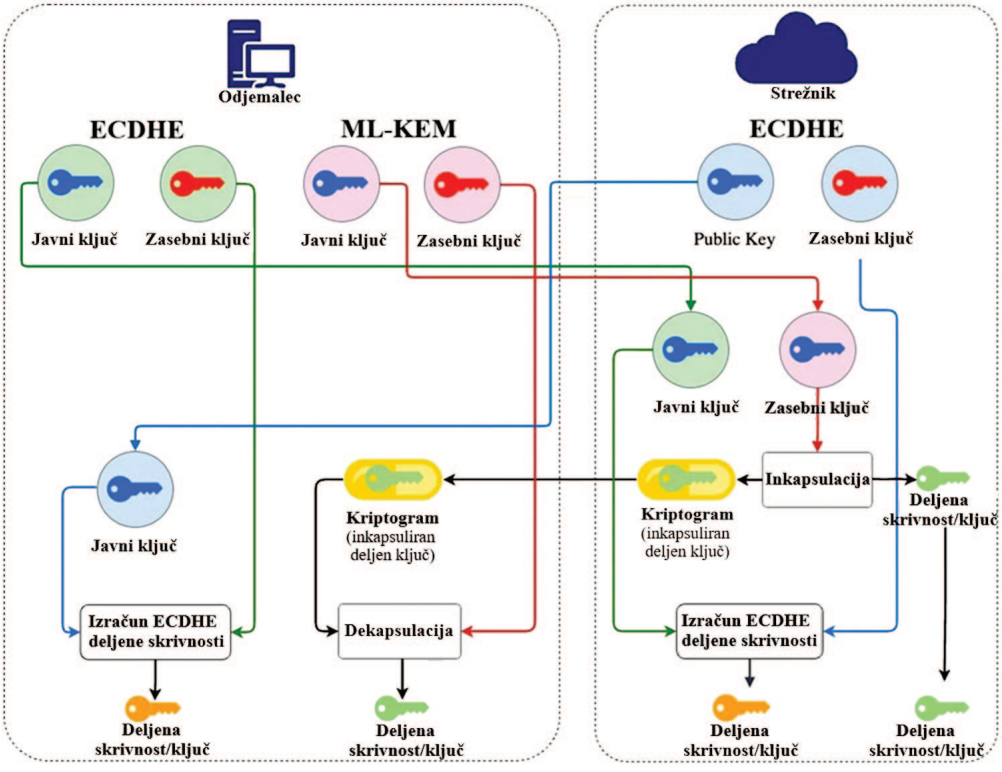
4 UPORABA POST-KVANTNIH ALGORITMOV KEM

Obstaja že kar nekaj (predlaganih) rešitev, ki vključujejo post-kvantne algoritme KEM. Izvajajo se lahko kot samostojni algoritmi KEM ali kot del hibridnega KEM, ki se nato integrira v protokol. Večina rešitev uporablja hibridne KEM rešitve.

Hibridni algoritmi KEM (angl. Hybrid KEM ali PQ/T Hybrid Combiners) so mehanizmi za inkapsulacijo ključa, ki združujejo post-kvanten algoritem KEM s tradicionalnim algoritmom KEM (bolj običajno imenovanim algoritem za izmenjavo ključa ali dogovor o ključu), z namenom doseganja robustnosti (ki jo zagotavljajo dobro uveljavljeni tradicionalni KEM) in odpornost pred kvantnimi računalniki (ki jo zagotavljajo post-kvantni KEM). Poleg varovanja pred morebitnimi neznanimi varnostnimi ranljivostmi in napakami v implementaciji novih post-kvantnih algoritmov KEM, so tradicionalni algoritmi včasih tudi zahtevani zaradi predpisov ali drugih pravil. Ko bodo post-kvantni algoritmi KEM bolje varnostno raziskani in preverjeni čez daljše časovno obdobje, se bo svet lahko vrnil k optimalnejši rešitvi uporabe enega samega algoritma KEM naenkrat. Hibridni algoritem KEM ostane varen, dokler je vsaj en od prispevajočih algoritmov varen. Kot bo razvidno iz primerov v tem poglavju, so hibridni algoritmi KEM le dva (možno je tudi več) neodvisnih KEM-ov, vsak od katerih neodvisno izmenja deljeno skrivnost, ki se nato pretvori v eno skrivnost, ki je odvisna od vseh uporabljenih algoritmov KEM. Razlike med hibridnimi algoritmi KEM izhajajo predvsem iz majhnih razlik v implementaciji (kakšne vrste ključev se uporabljajo ali kako se shranjujejo), uporabljenem naboru parametrov, načinu izračuna končne skupne skrivnosti/ključa in komunikacijskem procesu (tj. način izmenjave sporočil), ki ga opredeljujejo protokoli, v katere so algoritmi KEM integrirani.

Upoštevajte, da v času pisanja tega članka skoraj nobeden od obravnavanih hibridnih algoritmov KEM in njihova vključitev v protokole ni standardizirana. Predstavljeno so večinoma osnutki (angl. Draft) in se bodo v prihodnosti mogoče spremenili in/ali nikoli ne bodo sprejeti.

Najbolj znan in najbolj razširjen primer uporabe hibridnega algoritma KEM je zagotovo v TLS protokolu, ki je bil dobro podprt s strani proizvajalcev brskalnikov in tudi adaptacije s strani večjih spletnih ponudnikov. Post-kvantni hibridni dogovor o ključu ECDHE-MLKEM, razvit za uporabo v TLSv1.3 opredeljuje tri hibridne algoritme KEM [58]. To so X25519MLKEM768, SecP256r1MLKEM768 in SecP384r1MLKEM1024. Prva kombinacija uporablja X25519 ECDH (Elliptic Curve Diffie-Hellman) z ML-KEM in je zamenjava za X25519Kyber768 [59], [60], ki je bil prvi množično uporabljen hibridni KEM, ko je bil med drugim podprt v brskalnikih Chromium, Firefox in Safari ter na strežnikih Googla in Cloudflara [61] za zaščito povezav s TLS 1.3. Kot nakazuje že poimenovanje, ostala druga dva hibridna algoritma KEM združujeta ML-KEM (z naborom parametrov 768 ali 1024) z ECDH secp256r1 (NIST P-256) in secp384r1 (NIST P-384). Vsi omenjeni algoritmi ECDH uporabljajo efemerne ključe (tj. za enkratno uporabo), kar jih naredi ECDHE in predstavljajo glavne tradicionalne algoritme za dogovor o ključu, ki se trenutno uporabljajo. Ločeno je bi predlagan tudi



Slika 7: **Delovanje ECDHE-MLKEM [63]**

curveSM2MLKEM768 [62], ki je zelo enak konstruktom ECDHE-MLKEM, vendar uporablja eliptično krivuljo curveSM2.

V hibridnih algoritmih KEM se tipično kombinira tradicionalni KEM z nižjim nivojem varnosti in post-kvantni KEM z višjim nivojem varnosti (na primer ECDHE X25519 ali secp256r1 s 128-bitno varnostjo, v kombinaciji z ML-KEM-768, ki ima 192-bitno varnost oz. NIST varnostno ravnjo 3). Takšna kombinacija je narejena z namenom zaščite novih in še relativno malo preizkušenih post-kvantnih algoritmov KEM pred napredkom v njihovi kriptoanalizi.

X25519MLKEM768 deluje tako, da odjemalec najprej generira pare ključev za pogajanja (ključe X25519 je treba ustvariti vsakič, ker so efemerni, medtem ko je par ML-KEM-768 mogoče ponovno uporabiti). Oba javna ključa sta poslana strežniku. Strežnik ustvari svoj par ključev X25519 in za inkapsulacijo uporabi odjemalčev javni ključ ML-KEM, iz katerega dobi deljeno skrivnost ML-KEM in kriptogram. Kriptogram in ustvarjeni javni ključ X25519 sta poslana nazaj odjemalcu. Strežnik lahko nato združi odjemalčev javni ključ X25519 in svoj zasebni ključ, da ustvari deljeno skrivnost X25519. Medtem je odjemalec prejel strežnikov javni ključ X25519 in kriptogram iz ML-KEM. Odjemalec dekapsulira prejeti kriptogram s svojim zasebnim ključem ML-KEM, da dobi prvo (ML-KEM) deljeno skrivnosti, nato pa uporabi javni X25519 ključ strežnika in svoj zasebni X25519 ključ, da dobi drugo deljeno skrivnost. Tako odjemalec kot strežnik združita skrivnosti ML-KEM (na prvem mestu) in X25519, ki sta jih pridobila v izmenjavi, da dobita končno deljeno skrivnost. SecP256r1MLKEM768 in SecP384r1MLKEM1024 delujeta na popolnoma enak način, vendar je njihov vrstni red združevanja deljenih skrivnosti obrnjen (X25519MLKEM768 je verjetno edini hibridni KEM, ki ne sledi vrstnemu redu, ki ga nakazuje ime). Konstrukcija končne deljene skrivnosti brez uporabe KDF ali kakršnih koli dodatnih podatkov (npr. obeh javnih ključev ECDHE) razen deljenih skrivnosti je varno le zato, ker se ti hibridni algoritmi KEM uporabljajo v protokolu TLS 1.3, ki že sam razširi deljeno skrivnost/ključ in vključi potrebne dodatne podatke (tj. javne ključe), da se zagotovi IND-CCA2, in predvsem ne-zmožnost preoblikovanja (angl. non-malleability) [59]. Uporaba takšnih skrivnosti ni sicer ni priporočljiva, razen če se ustrezno upravlja, tako kot v TLS 1.3. Obstajajo samozadostni hibridni algoritmi KEM, ki vsebujejo združevanje deljenih skrivnosti,

Tabela 7: **Velikosti ključev, kriptogramov in deljenih skrivnosti/ključev post-kvantnih, tradicionalnih in hibridnih algoritmov KEM (v zlogih).**

Algoritem	Velikost javnega ključa	Velikost zasebnega ključa	Velikost kriptog-rama	Velikost podatkov, ki jih pošlje odjemalec	Velikost podatkov, ki jih pošlje strežnik	Velikost deljene skrivnosti
ML-KEM-768	1184	2400 (64 če je seme)	1088	1184	1088	32
ML-KEM-1024	1568	3168 (64 če je seme)	1568	1568	1568	32
X25519	32	32	32	32	32	32
X448	56	56	56	56	56	56
SecP256r1	65	32	65	65	65	32
SecP384r1	97	48	97	97	97	48
CurveSM2	65	32	65	65	65	32
X25519 MLKEM768	/	/	/	1216	1120	64
SecP256r1 MLKEM768	/	/	/	1249	1153	64
SecP384r1 MLKEM1024	/	/	/	1665	1665	80
CurveSM2MLKEM768	/	/	/	1249	1153	64
MLKEM1024-X448	/	/	/	1624	1624	88

tako da varnost ni odvisna od protokola, ki obdaja KEM. Na Sliki 7, je predstavljena izmenjava podatkov v ECDHE-MLKEM dogovoru za ključ.

Tabela 7 navaja velikosti podatkov post-kvantnih, tradicionalnih in hibridnih algoritmov KEM. V nasprotju s tradicionalnimi izmenjavami ključev vključevanje post-kvantnih algoritmov KEM povzroči signifikantno povečanje velikosti izmenjanih podatkov.

Kako se lahko predlagane hibridne sheme ECDHE-MLKEM ali kateri koli drug KEM vključi v TLS 1.3, je opisano v ločenem dokumentu [64]. Osnutek opredeljuje, kako poteka izmenjava ključev med odjemalci in strežniki, ki podpirajo ali ne podpirajo hibridnih rešitev. Pogajanja in izmenjava potrebnih podatkov (npr. deljenih skrivnosti) se izvaja v okviru TLS 1.3 usklajevanja (angl. Handshake). Velik del tega je opredelitev novih »groups« (TLS poimenovanje za algoritme, uporabljene za izmenjavo skrivnosti/ključa), ki sedaj vključuje možnosti s post-kvantnimi algoritmi KEM. Pristop hibridizacije v TLS je zelo preprost. Informacije o algoritmih, ki sestavljajo hibridno rešitev so poslani v sporočilih usklajevanja in preprosto združeni v eno vrednost, tako da obstoječe podatkovne strukture TLS protokola ostanejo nespremenjene. Končna deljena skrivnost se pridobi tudi z združevanjem rezultatov iz dveh (ali več) uporabljenih KEM-ov (kot je bilo predstavljeno v primeru ECDHE-MLKEM), ki se nato razširi na pričakovano velikost skrivnosti v funkciji izpeljave ključev (tj. KDF). Priporočilo je, da se post-kvantni algoritmi (in tradicionalni) KEM izvajajo z efemernimi ključi. Za post-kvantne algoritme KEM je dovoljena tudi ponovna uporaba parov ključev, vendar le, če to izbrani KEM podpira. Ker ima TLS omejene velikosti javnih ključev in kriptogramov (do 65535 zlogov), Classic McEliece ni primeren za uporabo v protokolu. Google je pripravil tudi osnutek o tem, kako predvidevajo, da bo TLS 1.3 deloval s hibridnimi algoritmi KEM [65], s poudarkom na načinih ublažitve povečane porabe računske zahtevnosti in pasovne širine, ki jo prinaša post-kvantna kriptografija.

X-Wing [66], [67] je hibridni KEM, ki združuje ML-KEM-768 in X25519. X-Wing predvideva uporabo semen namesto razširjenih ML-KEM ključev. To omogoča X-Wingu, da hrani majhno vrednost namesto vseh ključev (kot je bilo obravnavano v razdelku za ML-KEM). Zasebni ključi se lahko začasno shranijo v predpomnilniku (namesto da se shranijo samo kot seme), če se s tem izboljša učinkovitost (npr. v primerih večkratne dekapsulacije z istim ključem ali če se generacija in dekapsulacija ključev zgodita

v kratkem časovnem okviru). Velikost podatkov, ki se prenašajo med dvema odjemalcem in strežnikom je enaka velikosti X25519MLKEM768 (glej Tabelo 7). X-Wing uporablja SHA3-256 kot KDF za izračun končne deljene skrivnosti z zgoščevanjem deljene skrivnosti ML-KEM-768, deljene skrivnosti X25519, X25519 kriptograma, obeh javnih ključev X25519 in konstante (imenovane XWingLabel). X-Wing je bil narejen kot samostojni hibridni algoritem KEM, kar pomeni, da ima dokazano celovito varnost in ga je mogoče vstaviti kot zamenjavo za dogovor o ključu v katerem koli protokolu (npr. TLS, HPKE, SSH itd.) brez sprememb ali dodatnega upravljanja.

X-Wing določa kombinacijo algoritma ECDHE (X25519) s post-kvantnim ML-KEM, vendar bi lahko hibridni algoritem KEM zgradili na enak način iz poljubne kombinacije tradicionalnega in post-kvantnega algoritma KEM (ob upoštevanju nastalih varnostnih lastnosti). V ta namen je bil pripravljen osnutek za generično ustvarjanje hibridnih algoritmov KEM [68]. Opredeljuje generične tehnike za doseganje hibridnih KEM iz poljubnih post-kvantnih in tradicionalnih algoritmov KEM, ki ustrezajo specifičnim varnostnim lastnostim. Osnutek predlaga dve različni konstrukciji: KitchenSink in QSF. KitchenSink je bolj splošna možnost, zasnovana tako, da lahko združi največje število tradicionalnih in post-kvantnih algoritmov KEM, medtem ko je QSF bolj prilagojen in zahteva KEM s specifičnimi lastnostmi. X-Wing je v bistvu specifična različica QSF oblike izgradnje hibridnega algoritma KEM. Obstajajo tudi generični kombinatorji, kjer je pomembno, samo da imajo kombinirane sheme neodvisno generiranje ključev, način združevanja deljenih skrivnosti pa je pomemben za varnost celotnega sistema. Odobreni načini za združevanje deljenih skrivnosti dveh (ali več) KEM-ov so opredeljeni v [69] in [70].

Nazadnje je tu omenjenih še nekaj drugih protokolov, ki že načrtujejo oz. mogoče tudi že delujejo z uporabo hibridnih algoritmov KEM. Vsi hibridni algoritmi KEM so ustvarjeni na zelo podobne ali identične načine opisanemu. Tipično edine večje razlike so v tem kako se ti algoritmi integrirajo v protokole ali uporaba KDF. Za Internet Key Exchange Protocol Version 2 (IKEv2) so bilo pripravljenih precej osnutkov v pripravi na vključitev post-kvantnih in hibridnih algoritmov KEM [71], [72] vključno s predlogi o vključevanju specifičnih algoritmov [73], [74]. Nekateri ponudniki VPN so že vključili post-kvantne algoritme KEM v svoje protokole. ExpressVPN je integriral ML-KEM (in pred tem Kyber) v svoj protokol Lightway, NordVPN je integriral ML-KEM v svoj protokol NordLynx [75], in obstaja že predlog za post-kvantni WireGuard [76]. Cryptographic Message Syntax (CMS) v RFC 5652 [77] že predvideva uporabo post-kvantnih algoritmov KEM. Hibridne rešitve za CMS in X.509 PKI so bile predlagane v ločenem osnutku [78]. Predlog za uporabo post-kvantne kriptografije v OpenPGP vključuje hibridne KEM in hibridne sheme podpisov javnih ključev [79]. Signal je predstavil PQXDH [80], ki je post-kvantno varna zamenjava za njihov 3XDH protokol. Za razliko od ostalih protokolov, omenjenih v tem poglavju je PQXDH namenjen asinhronemu delovanju. Še ena post-kvantno varna alternativa protokolu 3XDH je K-Waay [81].

5 SKLEP

Razvoj kvantnih računalnikov se neomajno nadaljuje in njihov napredek je začel ogrožati kriptografske elemente, na katere se danes zanašamo za overjanje in varovanje komunikacije ter podatkov. Kvantni računalniki še niso dovolj zmogljivi, da bi ogrozili varnost trenutne kriptografije, vendar je njihov napredek dovolj hiter, da predstavljajo realno grožnjo za podatke, ki so danes zavarovani, vendar bodo še vedno imeli vrednost tudi v času, ko bodo kvantni računalniki postali dovolj zmogljivi, da bodo lahko razkrili te podatke. V preprečevanje tega, je potrebno čimprej nadomestiti uporabo ranljive kriptografije s kvantno-varnimi alternativami. Prispevek predstavlja širok pregled post-kvantnih mehanizmov za inkapsulacijo ključev (KEM), ki so pomemben del post-kvantne kriptografije. Algoritmi KEM skrbijo za varno izmenjavo skrivnosti na podlagi katerih se varuje nadaljnja komunikacija.

V prispevku smo se omejili na post-kvantno kriptografijo namenjeno izmenjavi ključev, zato algoritmi namenjeni overjanju oz. podpisovanju niso bili vključeni. Pri pregledu specifičnih post-kvantnih KEM algoritmov smo se omejili na algoritme, ki so se uvrstili v zadnji krog NIST-ovega tekmovanja za standardizacijo post-kvantnih algoritmov, algoritme, ki so v tekmovanju že bili izbrani za standardizacijo in algoritem FrodoKEM, ker ga nekatere pomembne organizacije izpostavljajo kot dober rezerven algoritem. Skupaj je bilo vključenih pet algoritmom KEM, za katere je bila predstavljena njihova osnovna konstrukcija, nabori

parametrov, varnostne lastnosti in učinkovitost njihovega delovanja. Pri analizi učinkovitosti oz. hitrosti njihovega delovanja smo se omejili na že zbrane in javno dostopne podatke.

Če povzamemo varnostne lastnosti kvantno varnih algoritmov KEM, razlike med njimi niso velike. Večinoma vsi dosegajo varnost IND-CCA2, čeprav je ob uporabi kratkotrajnih ključev zadostna IND-CPA. Lastnosti vezave so med algoritmi bolj različne, čeprav je glede na novost teh lastnosti možno, da bo še prišlo do sprememb (predvsem nestandardiziranih) algoritmov, ki bi spremenile varnostne lastnosti vezave. Ne glede na to, pomanjkanje nekaterih od teh lastnosti ni nujno velika ovira za KEM, vendar je pomembno te pomanjkljivosti upoštevati v KDF ali samem protokolu.

Ob prehodu na uporabo algoritmov KEM obstaja velika verjetnost, da bo shranjevanje semen prednostna oblika pred shranjevanjem razširjenih zasebnih ključev (vsaj v primeru ML-KEM). Izvajanje algoritmov z uporabo semen ponuja nekatere varnostne prednosti (npr. MAL-BIND-K-CT), porabi manj prostora za shranjevanje in naj bi bilo manj nagnjeno k napakam pri implementaciji (semena za razliko od dekapsulacijskega ključ ni potrebno vedno preverjati).

NIST varnostna raven 1 trenutno smatra kot popolnoma varna, čeprav je pomembno podpirati tudi višje ravni (če bo nekoč potrebno preiti na te). Dejansko izbiro ravni je potrebno opraviti glede na okoliščine uporabe in v skladu z drugimi kriptografskimi gradniki. Okoliščine, med drugim, vključujejo napredek kvantnih računalnikov, namen in okolje izvajanja. Ker s časom kvantni računalniki napredujejo, in če pride do pomembnih prebojev, ki znatno povečajo njihov napredek/moč, ali če se varnostne zahteve ali pomen zaščitenih podatkov sčasoma spreminja, je najboljša možnost spremembe varnostne ravni brez spreminjanja implementacije. To je povezano s kripto agilnostjo, ki je lastnost sistema, da lahko spreminja kriptografske mehanizme kot odziv na spreminjajoče se okoliščine. To vključuje nabore parametrov in/ali velikosti ključev ter tudi možnost spreminjanja algoritmov ali knjižnic. Kripto agilnost je strategija, ki zagotavlja trdno in odporno kibernetsko varnost.

Razpored algoritmov KEM glede na njihovo hitrost oz. učinkovitost delovanja ni takoj očiten. Najuspešnejši algoritem je zagotovo ML-KEM (prej imenovan Kyber). Drugi najhitrejši algoritem je težje določiti. Classic McEliece zavzema drugo mesto, če je namen ponovna uporaba ključev in posledično le redko generiranje novega para ključev. Če je pričakovana/zahtevana uporaba kratkotrajnih ključev, potem Classic McEliece ni učinkovita rešitev in je druga najučinkovitejša rešitev HQC. Glede na to da je potrebna uporaba efemernih ključev za zagotavljanje popolne prihodnje varnosti, je ponovna uporaba para ključev redka. To je predvidoma tudi razlog zakaj je NIST izbral HQC kot alternativni algoritem za standardizacijo in uporabo poleg ML-KEM. Naslednji najuspešnejši algoritem je BIKE, ki je v določenih operacijah celo boljši kot HQC. FrodoKEM je konsistentno počasnejši od ostalih algoritmov. Medtem ko različice, ki uporabljajo algoritem AES še lahko proizvedejo dobre rezultate (zlasti na strojni opremi, ki lahko v celoti izkoristi strojno podporo algoritma AES), različice SHAKE sploh niso konkurenčne. Morda bi bili rezultati te različice boljši na specializirani strojni opremi (čeprav to lahko velja za vse algoritme KEM). Tu je potrebno še upoštevati, da je ta razporeditev pripravljena zgolj na podlagi učinkovitosti delovanja. Varnost in velikost ključev ter kriptogramov KEM bi prav tako morali igrati vlogo pri izbiri najboljšega algoritma KEM za določen namen, čeprav glede na vse rezultate predstavljene v tem prispevku je ML-KEM praktično vedno najboljša izbira.

V zadnjem delu prispevka so bile predstavljene obstoječe aplikacije in/ali predloge za vključitev post-kvantnih algoritmov KEM v protokole ali hibridne algoritme KEM. Slednji je kombinacija tradicionalnega in post-kvantnega algoritma KEM, ki bo postal prehodni korak med tradicionalnimi dogovori o ključu in bodočo izključno uporabo post-kvantne kriptografije. Podrobneje predstavimo predvsem primer uporabe hibridnega algoritma KEM v protokolu TLS 1.3, ker ta predstavlja najbolj razširjen primer uporabe takšnih algoritmov. Zasnova predlaganih post-kvantnih protokolov odraža in poudarja specifične lastnosti algoritmov, obravnavane v prejšnjih delih prispevka.

Afiliacije

Ta raziskava je nastala kot rezultat projekta PQC Key Manager, ki ga je financirala Evropska vesoljska agencija (pog. št. 4000146856/24/NL/MH/mp) ter ob podpori raziskovalnega programa št. P2-0057, ki ga je sofinancirala Javna agencija za znanstvenoraziskovalno in inovacijsko dejavnost Republike Slovenije iz državnega proračuna.

LITERATURA

[1] P. W. Shor, »Algorithms for quantum computation: Discrete logarithms and factoring«, *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, str. 124–134, 1994, doi: 10.1109/SFCS.1994.365700.

[2] D. Moody, R. Perlner, A. Regenscheid, A. Robinson, in D. Cooper, »Transition to Post-Quantum Cryptography Standards«, nov. 2024. doi: 10.6028/NIST.IR.8547.IPD.

[3] »Securing Tomorrow, Today: Transitioning to Post-Quantum Cryptography«, 27. november 2024, *BSI - Bundesamt für Sicherheit in der Informationstechnik*. Pridobljeno: 4. april 2025. [Na spletu]. Dostopno na: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/PQC-joint-statement.pdf?__blob=publicationFile&v=3

[4] NIST, »Post-Quantum Cryptography Standardization«. Pridobljeno: 3. februar 2025. [Na spletu]. Dostopno na: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization

[5] C. Gidney in M. Ekerå, »How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits«, *Quantum*, let. 5, str. 433, apr. 2021, doi: 10.22331/q-2021-04-15-433.

[6] »Discover the World's Largest Quantum Computer in 2025«, spinquanta.com. Pridobljeno: 3. februar 2025. [Na spletu]. Dostopno na: https://www.spinquanta.com/newsDetail/ab2b9158-0907-4028-95b7-f8c5efdd9a2b

[7] S. Turner, P. Kampanakis, J. Massimo, in B. Westerbaan, »Internet X.509 Public Key Infrastructure - Algorithm Identifiers for the Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) - draft-ietf-lamps-kyber-certificates-08«, feb. 2025. Pridobljeno: 3. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-ietf-lamps-kyber-certificates/

[8] J. Massimo, P. Kampanakis, S. Turner, in B. Westerbaan, »Internet X.509 Public Key Infrastructure: Algorithm Identifiers for ML-DSA - draft-ietf-lamps-dilithium-certificates-07«, feb. 2025. Pridobljeno: 3. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-ietf-lamps-dilithium-certificates/

[9] M. Ounsworth, J. Gray, M. Pala, J. Klaußner, in S. Fluhrer, »Composite ML-DSA For use in X.509 Public Key Infrastructure and CMS - draft-ietf-lamps-pq-composite-sigs-03«, okt. 2024. Pridobljeno: 3. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-ietf-lamps-pq-composite-sigs/

[10] NIST, »Announcing PQC Candidates to be Standardized, Plus Fourth Round Candidates«. Pridobljeno: 3. februar 2025. [Na spletu]. Dostopno na: https://csrc.nist.gov/news/2022/pqc-candidates-to-be-standardized-and-round-4

[11] »NIST Selects HQC as Fifth Algorithm for Post-Quantum Encryption«, NIST. Pridobljeno: 8. april 2025. [Na spletu]. Dostopno na: https://www.nist.gov/news-events/news/2025/03/nist-selects-hqc-fifth-algorithm-post-quantum-encryption

[12] BSI, »Migration to Post Quantum Cryptography - Recommendations for action by the BSI«, 2021. Pridobljeno: 31. januar 2025. [Na spletu]. Dostopno na: https://www.bsi.bund.de

[13] »Follow up position paper on Post-Quantum Cryptography«, okt. 2023. Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://cyber.gouv.fr/en/publications/follow-position-paper-post-quantum-cryptography

[14] NIST, »FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard«, avg. 2024, doi: 10.6028/NIST.FIPS.203.

[15] C. Aguilar Melchor *idr.*, »HammingQuasi-Cyclic (HQC) - Fourth round version«, okt. 2024, Pridobljeno: 4. februar 2025. [Na spletu]. Dostopno na: https://pqc-hqc.org/documentation.html

[16] »Classic McEliece: Specification«. Pridobljeno: 4. februar 2025. [Na spletu]. Dostopno na: https://classic.mceliece.org/spec.html

[17] »FrodoKEM: Learning With Errors Key Encapsulation«, dec. 2024, Pridobljeno: 4. februar 2025. [Na spletu]. Dostopno na: https://frodokem.org/#spec

[18] N. Aragon *idr.*, »BIKE: Bit Flipping Key Encapsulation - Round 4 Submission«, okt. 2024, Pridobljeno: 4. februar 2025. [Na spletu]. Dostopno na: https://bikesuite.org/#spec

[19] M. Harmalkar, K. Jain, in P. Krishnan, »A Survey of Post Quantum Key Encapsulation Mechanism«, *Proceedings - 2024 5th International Conference on Mobile Computing and Sustainable Informatics, ICMCSI 2024*, str. 141–149, 2024, doi: 10.1109/ICMCSI61536.2024.00028.

[20] A. Langlois in D. Stehlé, »Worst-case to average-case reductions for module lattices«, *Des Codes Cryptogr*, let. 75, št. 3, str. 565–599, jun. 2015, doi: 10.1007/S10623-014-9938-4/METRICS.

[21] E. Fujisaki in T. Okamoto, »Secure integration of asymmetric and symmetric encryption schemes«, *Journal of Cryptology*, let. 26, št. 1, str. 80–101, jan. 2013, doi: 10.1007/S00145-011-9114-1/METRICS.

[22] J. B. Almeida *idr.*, »Formally verifying Kyber Episode V: Machine-checked IND-CCA security and correctness of ML-KEM in EasyCrypt«, *Cryptology ePrint Archive*, 2024, Pridobljeno: 3. februar 2025. [Na spletu]. Dostopno na: https://eprint.iacr.org/2024/843

[23] F. Valsorda, »Let's All Agree to Use Seeds as ML-KEM Keys«. Pridobljeno: 10. februar 2025. [Na spletu]. Dostopno na: https://words.filippo.io/dispatches/ml-kem-seeds/

[24] H. Singh, »Code based Cryptography: Classic McEliece«, jul. 2019, Pridobljeno: 4. februar 2025. [Na spletu]. Dostopno na: https://arxiv.org/abs/1907.12754v2

[25] R. J. McEliece, »A public-key cryptosystem based on algebraic coding theory«, *Prog. Rep., Jet Prop. Lab., California Inst. Technol., Pasadena, CA, 114–116*, 1978.

[26] M. Á. González de la Torre in L. Hernández Encinas, »About the Fujisaki-Okamoto Transformation in the Code-Based Algorithms of the NIST Post-quantum Call«, *Lecture Notes in Networks and Systems*, let. 532 LNNS, str. 75–85, 2023, doi: 10.1007/978-3-031-18409-3_8.

[27] »Classic McEliece: conservative code-based cryptography: design rationale«, okt. 2022. Pridobljeno: 4. februar 2025. [Na spletu]. Dostopno na: https://classic.mceliece.org/mceliece-rationale-20221023.pdf

[28] »BSI TR-02102-1: Cryptographic Mechanisms: Recommendations and Key Lengths«, jan. 2024. Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr02102/tr02102_node.html

[29] »Classic McEliece: ISO«. Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://classic.mceliece.org/iso.html

[30] »Classic McEliece: conservative code-based cryptography: what plaintext confirmation means«, okt. 2023. Pridobljeno: 4. februar 2025. [Na spletu]. Dostopno na: https://classic.mceliece.org/nist.html

[31] »Classic McEliece: conservative code-based cryptography: guide for implementors«, okt. 2022, Pridobljeno: 4. februar 2025. [Na spletu]. Dostopno na: https://classic.mceliece.org/nist.html

[32] »FrodoKEM Learning With Errors Key Encapsulation - Annex on FrodoKEM updates«. Pridobljeno: 5. februar 2025. [Na

spletu]. Dostopno na: https://frodokem.org/files/FrodoKEM--annex-20230418.pdf
[33] NIST, »Advanced Encryption Standard (AES) - FIPS 197«, maj 2023, doi: 10.6028/NIST.FIPS.197-UPD1.
[34] NIST, »SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions - FIPS 202«, avg. 2015, doi: 10.6028/NIST.FIPS.202.
[35] »FrodoKEM«. Pridobljeno: 5. februar 2025. [Na spletu]. Dostopno na: https://frodokem.org/
[36] M. Á. González de la Torre in L. Hernández Encinas, »About the Fujisaki-Okamoto Transformation in the Code-Based Algorithms of the NIST Post-quantum Call«, *Lecture Notes in Networks and Systems*, let. 532 LNNS, str. 75–85, 2023, doi: 10.1007/978-3-031-18409-3_8.
[37] S. Arpin, J. B. Lau, R. Perlner, A. Robinson, J.-P. Tillich, in V. Vasseur, »Error floor prediction with Markov models for QC--MDPC codes«, *Cryptology ePrint Archive*, 2025, Pridobljeno: 5. februar 2025. [Na spletu]. Dostopno na: https://eprint.iacr.org/2025/153
[38] D. Hofheinz, K. Hövelmanns, in E. Kiltz, »A Modular Analysis of the Fujisaki-Okamoto Transformation«, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, let. 10677 LNCS, str. 341–371, 2017, doi: 10.1007/978-3-319-70500-2_12/FIGURES/19.
[39] G. Alagic *idr.*, »Recommendations for Key-Encapsulation Mechanisms - NIST SP 800-227«, jan. 2025, doi: 10.6028/NIST.SP.800-227.IPD.
[40] »Open Quantum Safe - Algorithms«. Pridobljeno: 22. januar 2025. [Na spletu]. Dostopno na: https://openquantumsafe.org/liboqs/algorithms/
[41] D. Connolly, »ML-KEM Post-Quantum Key Agreement for TLS 1.3 - draft-connolly-tls-mlkem-key-agreement-05«. Pridobljeno: 5. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-connolly-tls-mlkem-key-agreement/
[42] S. Fluhrer, Q. Dang, J. Preuß Mattsson, K. Milner, in D. Shiu, »ML-KEM Security Considerations - draft-sfluhrer-cfrg-ml--kem-security-considerations-02 «, 2024. Pridobljeno: 3. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-sfluhrer-cfrg-ml-kem-security-considerations/
[43] R. Cramer in V. Shoup, »Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack«, *Cryptology ePrint Archive*, 2001, Pridobljeno: 5. februar 2025. [Na spletu]. Dostopno na: https://eprint.iacr.org/2001/108
[44] D. Connolly, »How to Hold KEMs«. Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://durumcrustulum.com/2024/02/24/how-to-hold-kems/
[45] C. Cremers, A. Dax, in N. Medinger, »Keeping Up with the KEMs: Stronger Security Notions for KEMs and Automated Analysis of KEM-based Protocols«, *CCS 2024 - Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, str. 1046–1060, dec. 2024, doi: 10.1145/3658644.3670283.
[46] J. Krämer, P. Struck, in M. Weishäupl, »Binding Security of Implicitly-Rejecting KEMs and Application to BIKE and HQC«, *Cryptology ePrint Archive*, 2024, Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://eprint.iacr.org/2024/1233
[47] C. Cremers, A. Dax, in N. Medinger, »Keeping Up with the KEMs: Stronger Security Notions for KEMs and automated analysis of KEM-based protocols«, *Cryptology ePrint Archive*, nov. 2024, Pridobljeno: 12. februar 2025. [Na spletu]. Dostopno na: https://eprint.iacr.org/2023/1933
[48] S. Schmieg, »Unbindable Kemmy Schmidt: ML-KEM is neither MAL-BIND-K-CT nor MAL-BIND-K-PK«, *Cryptology ePrint Archive*, apr. 2024, Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://eprint.iacr.org/2024/523
[49] NIST, »Post-Quantum Cryptography - Security (Evaluation Criteria)«. Pridobljeno: 5. februar 2025. [Na spletu]. Dostopno na: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria)
[50] L. K. Grover, »A fast quantum mechanical algorithm for database search«, *Proceedings of the Annual ACM Symposium on Theory of Computing*, let. Part F129452, str. 212–219, jul. 1996, doi: 10.1145/237814.237866/ASSET/0A4D7788-347B-41F8-9E19-26B6DE2ACF0A/ASSETS/237814.237866.FP.PNG.
[51] G. Brassard, P. Hoyer, in A. Tapp, »Quantum Algorithm for the Collision Problem«, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, let. 1380, str. 163–169, maj 1997, doi: 10.1007/BFb0054319.
[52] C. Zalka, »Grover's quantum searching algorithm is optimal«, *Phys Rev A (Coll Park)*, let. 60, št. 4, str. 2746, okt. 1999, doi: 10.1103/PhysRevA.60.2746.
[53] S. Fluhrer, »Reassessing Grover's Algorithm«, *Cryptology ePrint Archive*, 2017, Pridobljeno: 5. februar 2025. [Na spletu]. Dostopno na: https://eprint.iacr.org/2017/811
[54] Sarah D., »On the Practical cost of Grover for AES Key Recovery«, v *Fifth PQC Standardization Conference*, apr. 2024. Pridobljeno: 6. februar 2025. [Na spletu]. Dostopno na: https://csrc.nist.gov/Presentations/2024/practical-cost-of--grover-for-aes-key-recovery
[55] »eBATS: ECRYPT Benchmarking of Asymmetric Systems«. Pridobljeno: 13. februar 2025. [Na spletu]. Dostopno na: https://bench.cr.yp.to/ebats.html
[56] »List of key-encapsulation mechanisms measured«. Pridobljeno: 13. februar 2025. [Na spletu]. Dostopno na: https://bench.cr.yp.to/primitives-kem.html
[57] »Open Quantum Safe«. Pridobljeno: 13. februar 2025. [Na spletu]. Dostopno na: https://openquantumsafe.org/
[58] K. Kwiatkowski, P. Kampanakis, B. Westerbaan, in D. Stebila, »Post-quantum hybrid ECDHE-MLKEM Key Agreement for TLSv1.3 - draft-kwiatkowski-tls-ecdhe-mlkem-03«. Pridobljeno: 10. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-kwiatkowski-tls-ecdhe-mlkem/
[59] B. Westerbaan in D. Stebila, »X25519Kyber768Draft00 hybrid post-quantum key agreement - draft-tls-westerbaan-xyber768d00-03«. Pridobljeno: 10. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-tls-westerbaan-xyber768d00/
[60] J. Schaumann, »TLS 1.3 Hybrid Key Exchange using X25519Kyber768 / ML-KEM«. Pridobljeno: 10. februar 2025. [Na spletu]. Dostopno na: https://www.netmeister.org/blog/tls-hybrid-kex.html
[61] »Post-Quantum Key Agreement at Cloudflare«. Pridobljeno: 10. februar 2025. [Na spletu]. Dostopno na: https://pq.cloudflareresearch.com/
[62] P. Yang, C. Peng, J. Hu, in S. Sun, »Hybrid Post-quantum Key Exchange SM2-MLKEM for TLSv1.3 - draft-yang-tls--hybrid-sm2-mlkem-01«. Pridobljeno: 10. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft--yang-tls-hybrid-sm2-mlkem/
[63] U. Pathum, »X25519Kyber768 Post-Quantum Key Exchange for HTTPS Communication«, Medium. Pridobljeno: 10. februar 2025. [Na spletu]. Dostopno na: https://medium.com/@

hwupathum/x25519kyber768-post-quantum-key-exchange-for-https-communication-70eba681931d
[64] D. Stebila, S. Fluhrer, in S. Gueron, »Hybrid key exchange in TLS 1.3 - draft-ietf-tls-hybrid-design-12«, Internet Engineering Task Force (IETF) . Pridobljeno: 10. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/
[65] D. Benjamin, »TLS Key Share Prediction - draft-ietf-tls-key-share-prediction-01«. Pridobljeno: 10. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-ietf-tls-key-share-prediction/
[66] M. Barbosa *idr.*, »X-Wing«, *IACR Communications in Cryptology*, let. 1, št. 1, str. 3006–5496, apr. 2024, doi: 10.62056/A3QJ89N4E.
[67] D. Connolly, P. Schwabe, in B. Westerbaan, »X-Wing: general-purpose hybrid post-quantum KEM - draft-connolly-cfrg-xwing-kem-06«. Pridobljeno: 10. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-connolly-cfrg-xwing-kem/
[68] D. Connolly, »Hybrid PQ/T Key Encapsulation Mechanisms - draft-irtf-cfrg-hybrid-kems-01«. Pridobljeno: 10. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-irtf-cfrg-hybrid-kems/
[69] G. Alagic *idr.*, »Recommendations for Key-Encapsulation Mechanisms«, jan. 2025, doi: 10.6028/NIST.SP.800-227.IPD.
[70] M. Ounsworth, A. Wussler, in S. Kousidis, »Combiner function for hybrid key encapsulation mechanisms (Hybrid KEMs) - draft-ounsworth-cfrg-kem-combiners-05«. Pridobljeno: 10. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-ounsworth-cfrg-kem-combiners/
[71] V. Smyslov, »RFC 9242 - Intermediate Exchange in the Internet Key Exchange Protocol Version 2 (IKEv2)«, dec. 2024. Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/rfc9242/
[72] C. Tjhai *idr.*, »RFC 9370 - Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2)«, dec. 2023. Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/rfc9370/
[73] P. Kampanakis in G. Ravago, »Post-quantum Hybrid Key Exchange with ML-KEM in the Internet Key Exchange Protocol Version 2 (IKEv2) - draft-kampanakis-ml-kem-ikev2-09«. Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-kampanakis-ml-kem-ikev2/
[74] G. Wang, »Post-quantum Hybrid Key Exchange in the IKEv2 with FrodoKEM - draft-wang-hybrid-kem-ikev2-frodo-02«. Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-wang-hybrid-kem-ikev2-frodo/
[75] P. Membrey, »ExpressVPN: Early Adopter of ML-KEM for Quantum Encryption«. Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://www.expressvpn.com/blog/ml-kem-lightway-upgrade/?srsltid=AfmBOoqeaBwEW2X41uPZqYvJa44JB4OctMt_00Cy3uGipXgMUe1YO9xh
[76] A. Hülsing, K.-C. B. Ning KPN, P. Schwabe, F. Johanna Weber, in P. R. Zimmermann, »Post-quantum WireGuard«, *Cryptology ePrint Archive*, sep. 2023, Pridobljeno: 12. februar 2025. [Na spletu]. Dostopno na: https://eprint.iacr.org/2020/379
[77] R. Housley, J. Gray, in T. Okubo, »RFC 9629 - Using Key Encapsulation Mechanism (KEM) Algorithms in the Cryptographic Message Syntax (CMS)«, avg. 2024. Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/rfc9629/
[78] M. Ounsworth, J. Gray, M. Pala, J. Klaußner, in S. Fluhrer, »Composite ML-KEM for use in X.509 Public Key Infrastructure and CMS - draft-ietf-lamps-pq-composite-kem-05«. Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-ietf-lamps-pq-composite-kem/
[79] S. Kousidis, J. Roth, F. Strenzke, in A. Wussler, »Post-Quantum Cryptography in OpenPGP - draft-ietf-openpgp-pqc-07«. Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://datatracker.ietf.org/doc/draft-ietf-openpgp-pqc/
[80] E. Kret in R. Schmidt, »The PQXDH Key Agreement Protocol«. Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://signal.org/docs/specifications/pqxdh/#preliminaries
[81] D. Collins, L. Huguenin-Dumittan, N. K. Nguyen, N. Rolin, in S. Vaudenay, »K-Waay: Fast and Deniable Post-Quantum X3DH without Ring Signatures«, *Cryptology ePrint Archive*, 2024, Pridobljeno: 11. februar 2025. [Na spletu]. Dostopno na: https://eprint.iacr.org/2024/120

▪

Marko Kompara je raziskovalec in asistent na Fakulteti za elektortehniko, računalništvo in informatiko, kjer je tudi pridobil doktorat z disertacijo na temo protokolov za dogovarjanje ključev v omejenih okoljih. Od takrat je bil in je še vedno vključen v številke projekte na področju kibernetske varnosti (npr. CyberSec4Europe, RUKIV, RPUP, AKADIMOS). Njegovi raziskovalni interesi so zasebnost, kriptografija, brezžične komunikacije in varnost informacijskih sistemov.

▪

Marko Hölbl raziskuje kibernetsko varnost in zasebnost na širokem področju, od kriptografije do uporabniških vidikov informacijske varnosti in zasebnosti. Trenutno je prodekan za raziskave na FERI UM, aktiven član in generalni sekretar CEPIS LSI, član ECSO (WG6), član izvršnega odbora Slovenskega društva Informatika in član Sekcije za kibernetsko varnost pri Gospodarski zbornici Slovenije. Sodeluje in/ali vodi številne projekte iz področja kibernetske varnosti (npr. CyberSec4Europe, RUKIV, RPUP, AKADIMOS)