Alle nieuws

Van vastzitten naar vooruitgang: architectuur met veerkracht

3 months ago

Deze blog is geschreven door de deelnemers van het PQC Benchmarking-project, afkomstig van TNO, Achmea, Belastingdienst, ABN Amro en ING. De testcase die in deze blog wordt beschreven, is uitgevoerd door de Belastingdienst.

Introductie

Nu de ontwikkelingen in quantumtechnologie voortschrijden, worden we geconfronteerd met een realiteit waarin de kracht van cryptografie afneemt. Decennialang konden we vertrouwen op stabiele standaarden, gebaseerd op Diffie Hellman, RSA en elliptische krommen (ECC) algoritmes. Nu moeten we opnieuw digitale beveiligingsdoelen zoals vertrouwelijkheid, integriteit en beschikbaarheid bereiken, door over te stappen op post-quantum cryptografie (PQC).

De internationale gemeenschap is bezig met het ontwikkelen en standaardiseren van nieuwe post-quantum algoritmes. IT-afdelingen moeten zich voorbereiden en deze algoritmes gaan toepassen in hun infrastructuur en systemen. Naast gereedheid bij leveranciers vereist dit eigen kennis en inzicht: de nieuwe algoritmes zijn minder bekend bij IT-professionals en er is weinig ervaring met de effecten in productieomgevingen.

Bij PQC-migratie spelen IT- en securityarchitecten een leidende rol. Zij moeten

beschikbare alternatieven en hun haalbaarheid per use case beoordelen. Deze complexe taak vereist expertise die sterk gebaat is bij praktijkervaring, zoals benchmarks in echte implementatiescenario's.

In ons project zijn we gestart met een benchmarktest om precies dat te doen. In deze blog delen we onze meest waardevolle inzichten. Je leest over de verschillende valkuilen die we tegenkwamen die typisch zijn voor migratie. Het goede nieuws: we vonden manieren om eruit te komen!

De zoektocht naar inzicht

Om de benchmarkreis te starten, moesten we een use case kiezen die praktische kennis en inzichten zou opleveren over het gebruik van PQC-algoritmes en die uitvoerbaar was binnen de tijdsgrenzen van het project. We kozen voor het migreren van de connectie-opzet van een message queuing (MQ) applicatie, waarbij gebruikers PQC-certificaten gebruiken in plaats van klassieke certificaten. Wat zou de impact op de prestaties zijn?

Een belangrijke reden voor deze keuze was dat MQ-applicaties vaak onderdeel zijn van IT-architecturen die hoge prestaties vereisen qua doorlooptijd en volume (throughput). Daarnaast was er een praktische kant: MQ maakt deel uit van het dagelijkse werk van veel teams, wat betekende dat essentiële middelen voor een succesvolle benchmark – zoals domeinkennis, tooling en een testomgeving – waarschijnlijk al beschikbaar waren.

Na het kiezen van een use case moesten we bepalen welke PQC-algoritmes en prestatiekenmerken we zouden testen. Het was interessant om te zien wat het effect zou zijn op de throughput bij het gebruik van PQC-certificaten voor de opzet van veilige verbindingen, aangezien dat meestal de bottleneck is. We besloten certificaten te maken met verschillende PQC-algoritmes voor sleuteluitwisseling en digitale handtekeningen, inclusief hybride combinaties van klassieke en post-quantum algoritmes, en de resultaten te vergelijken met de huidige quantum-kwetsbare versie van de MQ-applicatie. Andere prestatie indicatoren die we meenamen waren CPU-gebruik, netwerkbandbreedte en geheugenverbruik, omdat die inzichten kunnen bieden voor andere organisaties en toepassingen.

Testscenario's

Omdat er verschillende algoritmeopties zijn binnen post-quantum cryptografie, moesten we selecteren welke we zouden meenemen in de migratie en benchmark.

We kozen de vier NIST-standaarden (ML-KEM, ML-DSA, SLH-DSA en Falcon) en FrodoKEM en McEliece om te testen. Hoewel FrodoKEM en McEliece nog niet gestandaardiseerd zijn, worden ze vaak aanbevolen door EU-lidstaten. Deze selectie biedt een goed overzicht van populaire algoritmes, met uiteenlopende prestaties en verschillende veiligheidsaannames.

Meerdere organisaties bevelen het gebruik van hybride schema’s aan, waarbij klassieke en post-quantum algoritmes worden gecombineerd. Als één van de twee algoritmes faalt, blijft de veiligheid behouden, wat een vangnet biedt als een nieuw algoritme minder veilig blijkt dan gedacht. We namen meerdere hybride schema’s op in onze benchmark.

Om de impact van het migreren van sleuteluitwisseling (KEMs) en digitale handtekeningen te begrijpen, besloten we de huidige klassieke oplossingen te benchmarken als referentie, post-quantum KEMs afzonderlijk te testen (met klassieke handtekeningen), en post-quantum handtekeningen afzonderlijk te testen (met klassieke KEMs), en ten slotte de combinatie van post-quantum KEMs en handtekeningen te testen. Een bijkomend voordeel van deze opzet is dat

organisaties die eerst KEMs willen migreren om het “harvest now, decrypt later”-risico te beperken, een indruk krijgen van de initiële prestatievermindering. Naast inzicht in de impact op prestaties, wilden we ook leren hoe moeilijk het is om al deze algoritmes (en hun hybride combinaties) in echte producten te integreren.

PQC Algorithms:

Key exchange (KEM):
- A. ML-KEM,
- B. FrodoKEM,
- C. McEliece

Digital signature (DSA):
- D. ML-DSA
- E. Falcon
- F. SLH-DSA

Test cases:

	KEX	DSA	Algorithms to test
0.	Current (FFDH)	Current (RSA)	First baseline
1.	ECC	ECC	Second baseline
2.	PQC	ECC	A,B,C
3.	ECC	PQC	D,E,F
4.	PQC	PQC	All combinations of A-C with D-F
5.	(ECC+PQC)	ECC	A,B,C
6.	ECC	(ECC+PQC)	D,E,F
7.	(ECC+PQC)	(ECC+PQC)	All combinations of A-C with D-F

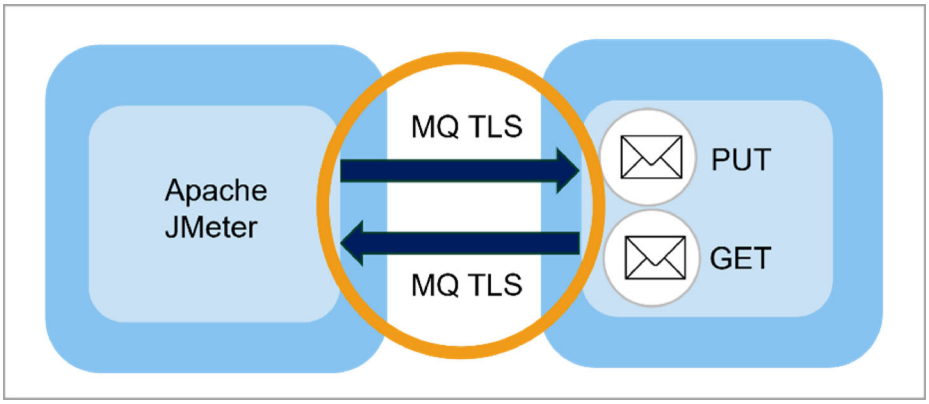
Van de 32 combinaties werden er in deze test slechts 22 ondersteund. Voor verdere technische details is er een uitgebreide bijlage bij deze blog. Je kunt die hier lezen: How to get un-stuck by Architecting for Resilience 2 | News

---

Veerkrachtige architectuur

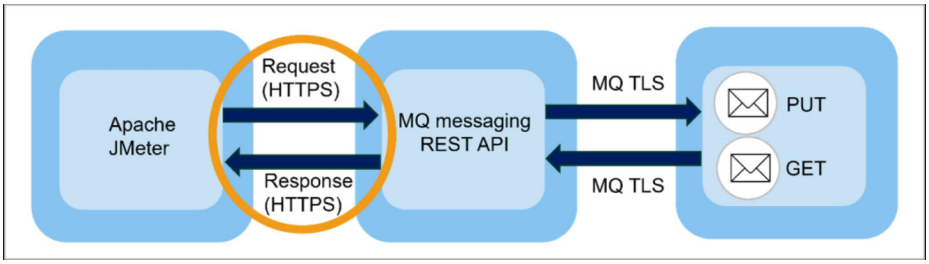
Na het bepalen van de testscenario’s konden we aan de slag met het migreren van de MQ-setup. Ons MQ team gebruikt Apache JMeter voor prestatietests als onderdeel van hun dagelijkse werk. Ze gebruiken certificaten voor het MQ-product en ook voor TLS-verbindingen tussen clients en servers. Dat betekende dat er al tooling beschikbaar was én dat ze praktische certificaatkennis hadden. Een goed begin.

De onderstaande figuur toont ons initiële scenario. Het oranje deel is waar we PQC wilden configureren:



Helaas liepen we tegen beschikbaarheidsproblemen aan: er konden geen leverancierscertificaten worden gebruikt. Een ander groot probleem was dat het MQ-product nog geen PQC ondersteunde, waardoor we onze testscenario's niet konden uitvoeren met dat product.

Met snel schakelen pasten we ons plan aan en besloten we een OpenSSL certificaat te gebruiken, voor de verbinding naar de REST API naar MQ:



Maar ook dat werkte niet. Hoewel PQC-ondersteuning beschikbaar was in OpenSSL, kon de REST API niet aangepast worden om het PQC-certificaat te accepteren. Weer een tegenslag. Hoewel dit slecht nieuws was, bevestigde het onze vermoedens dat migratie geen triviale taak is en dat ervaring in dit stadium inzicht geeft in praktische uitdagingen.

Tijd om afscheid te nemen en van perspectief te veranderen.

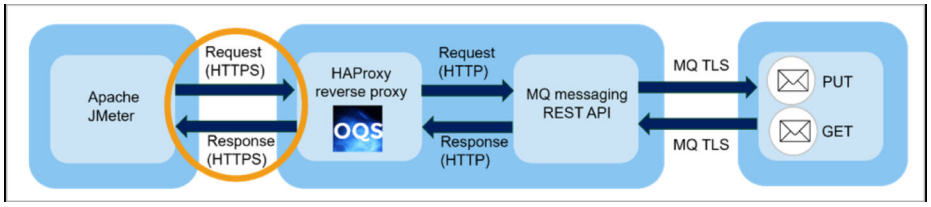
Cryptografie is een maatregel om risico's te beperken. Als het algoritme kwetsbaar is, is die maatregel ineffectief en keert het risico terug. Je moet dan opnieuw kijken hoe je dat risico beperkt – dat kan met een ander cryptografisch middel, maar ook door de gevoeligheid van data te verminderen, toegang tot een applicatie te beperken, fysieke beveiliging toe te

passen of een andere beveiligingsarchitectuur te gebruiken.

We moesten out-of-the-box denken. Zoals Anita Wehmann van het Nederlandse QvC-programma zegt: Dutch quantum safe cryptography support program (QvC)

“Je zit niet vast als je niet kunt migreren”

That made us pivot the architectural design to the following final setup:



We koppelden de architectuur los om de flexibiliteit te krijgen die we nodig hadden. We plaatsten een reverse proxy tussen de componenten die PQC ondersteunden en die dat nog niet deden. Specifiek gebruikten we een HAProxy met extenties van het OpenQuantumSafe framework and OpenSSL.

Voorlopig accepteren we SSL-offloading totdat de leverancier PQC ondersteunt. Hierdoor zijn er minder quantumveilige verbindingen dan we zouden willen (alleen het oranje deel).

Maar deze aanpak heeft voordelen. We kunnen nu wel een benchmark uitvoeren, de meeste algoritmische combinaties testen, tooling opzetten en ervaring opdoen. Op een later moment, wanneer andere componenten PQC ondersteunen, kunnen we terugschakelen naar de eerdere architectuur en de tests opnieuw uitvoeren.

Resultaten

Uiteindelijk konden we hierdoor alle gedefinieerde testscenario’s uitvoeren. De tooling leverde informatie op over onze belangrijkste metriek: berichtverwerking per seconde. We kunnen deze data gebruiken om de configuraties te analyseren. Daarnaast bekeken we CPU-tijd, netwerkbandbreedte en controleerden we op geheugengebruik en pakketverlies. In deze sectie delen we

onze belangrijkste conclusies. Een gedetailleerde analyse vind je in de addendum bij deze blog.

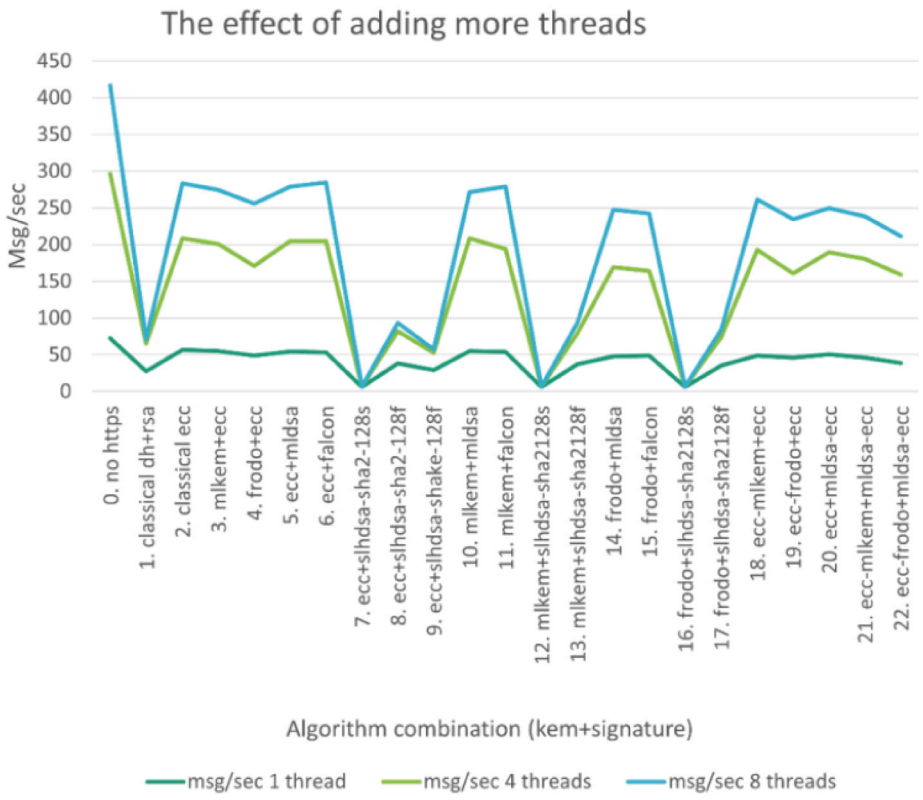
Baseline

Wat betreft de baseline-data blijkt dat klassieke ECC aanzienlijk sneller is dan klassieke RSA en Diffie Hellman. ECC is dus een goede referentie voor wat klassieke cryptografie momenteel te bieden heeft.

Multithreading

Ons testscenario werd meerdere keren uitgevoerd met verschillende aantallen threads. Het opschalen van threads lijkt voor bijna alle KEM- en DSA-configuraties een lineaire prestatieverbetering te geven. CPU-gebruik is dus meestal niet de bottleneck. Tijdens de tests zagen we geen pakketverlies, wat deze conclusie ondersteunt. Een waarschijnlijke verklaring is dat netwerkvertragingen groot genoeg zijn zodat andere threads gestart kunnen worden voordat de benodigde berichten binnenkomen.

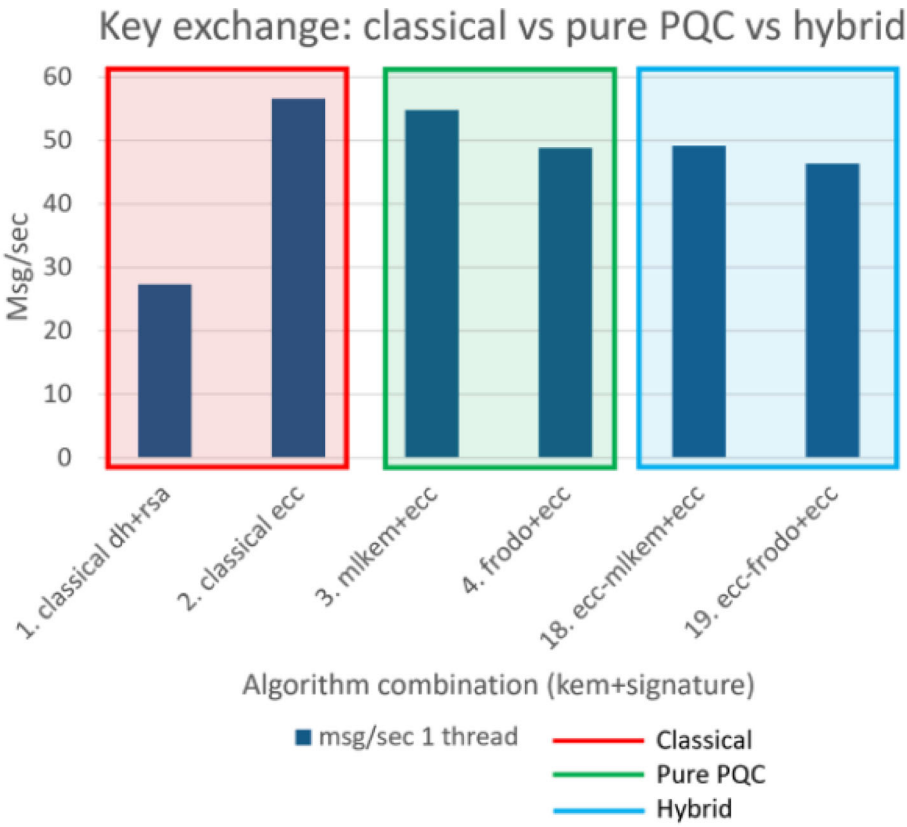
Een opvallende bevinding is dat de variant van het digitale handtekeningalgoritme SLH-DSA die geoptimaliseerd is voor berichtgrootte veel van de resources vraagt. Het verhogen van het aantal threads bij SLH-DSA-testcases leidde tot maximale CPU-belasting en geen lineaire prestatieverbetering. Dit lijken CPU-intensieve operaties te zijn (testcases 7, 12 en 16 in de grafiek).



Sleutelwisseling

Bij vergelijking tussen het klassieke ECC met de pure vervanging door het PQC algoritme ML-KEM zijn de prestaties vergelijkbaar. Verrassend genoeg liggen de resultaten voor FrodoKEM daar niet ver onder. FrodoKEM heeft veiligheidsvoordelen vanwege conservatievere aannames dan ML-KEM. Gezien het kleine prestatieverschil kan het een relevante optie zijn voor organisaties die behoudende met risico's omgaan.

Het prestatieverlies bij het gebruik van hybride ML-KEM in plaats van pure ML-KEM lijkt veel kleiner dan verwacht, ongeveer 10 procent.

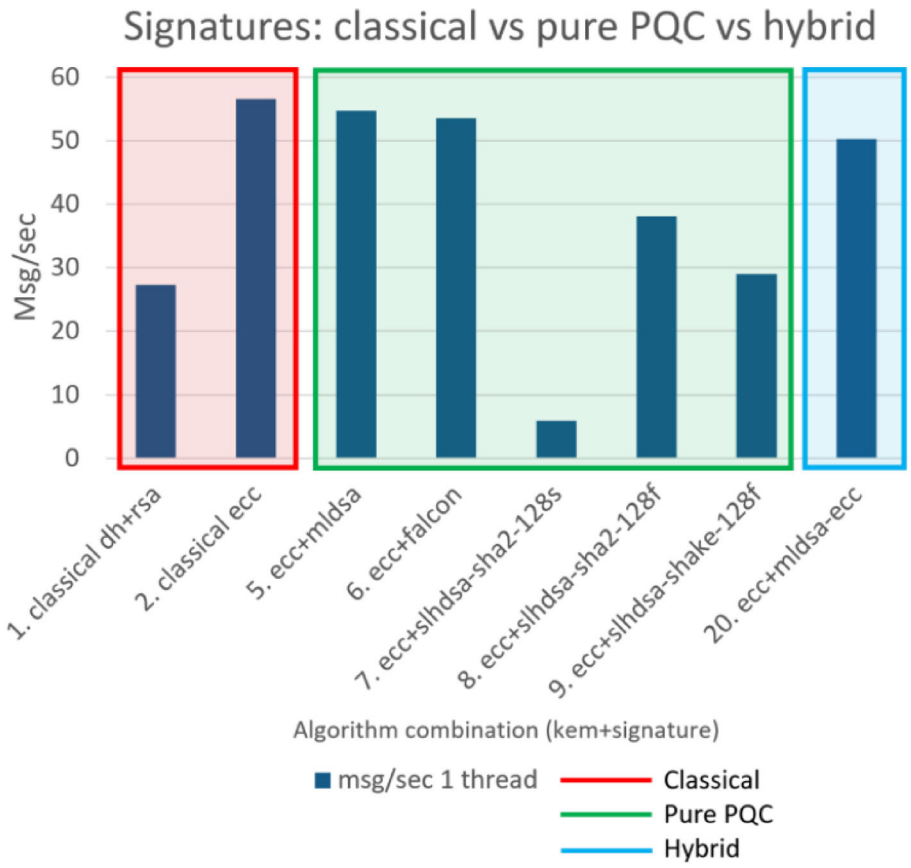


Digitale handtekeningen

Bij het aanpassen van alleen het handtekeningalgoritme naar PQC zien we meer variatie in de prestatie.

ML-DSA en Falcon-handtekeningen in een pure PQC-configuratie vergeleken met klassieke ECC-handtekeningen zijn qua prestaties vergelijkbaar. Maar voor SLH-DSA-handtekeningen is het een ander verhaal. Die presteren aanzienlijk slechter, vooral de grootte-geoptimaliseerde variant van SLH-DSA (slhdsa-sha2-128s).

Voor hybride configuraties verwachtten we enig prestatieverlies door extra verificatie, maar onze data ondersteunt dat niet. De resultaten suggereren dat hybride handtekeningen slechts een beperkte extra prestatie impact hebben. Dat is goed nieuws, aangezien hybride configuraties vaak worden aanbevolen in deze fase van de PQC-migratie.



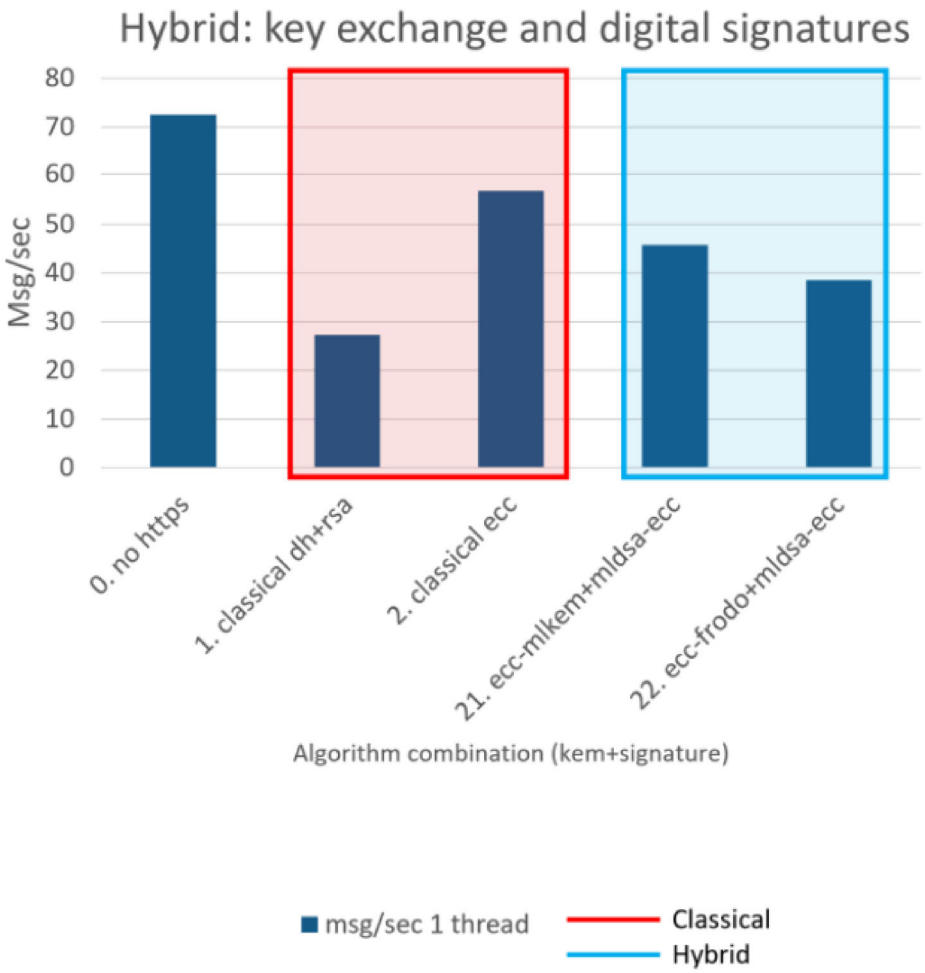
Pure PQC Sleuteluitwisseling en digitale handtekening

Op basis van eerdere resultaten verwachtten we dat de combinatie van pure PQC voor zowel KEMs als DSAs geen grote prestatieverlies zou geven – mits SLH-DSA buiten beschouwing wordt gelaten. Dat zien we bevestigd in onze resultaten voor de ML-KEM en ML-DSA-combinatie. Iets meer prestatieverlies is zichtbaar bij combinaties met FrodoKEM. Let wel op dat FrodoKEM problematisch kan zijn als berichtgrootte belangrijk is, wat in onze testopzet blijkbaar niet het geval was. Zie de bijlage voor details over berichtgrootte.

KEM	DSA	Msg/sec (1 thread)
First Baseline (DH)	First Baseline (RSA)	27,3
Second Baseline (ECC)	Second Baseline (ECC)	56,6
ML-KEM	ML-DSA	55,3
ML-KEM	Falcon	53,8
ML-KEM	SLH-DSA-SHA2-128s	6
ML-KEM	SLH-DSA-SHA2-128f	37,3
FrodoKEM	ML-DSA	47,8
FrodoKEM	Falcon	48,7
FrodoKEM	SLH-DSA-SHA2-128s	5,9
FrodoKEM	SLH-DSA-SHA2-128f	35,5

Hybrid sleuteluitwisseling en digitale handtekening

In de vorige paragraaf bekeken we pure PQC-testresultaten. Welke conclusies kunnen we trekken uit het combineren van de snelste klassieke algoritmes met de snelste post-quantum algoritmes? In hybride configuratie voor zowel sleuteluitwisseling als digitale handtekeningen zagen we ongeveer 20% prestatieverlies in berichtenverwerking. Door meer threads toe te voegen, daalde dat tot ongeveer 15%.



Conclusie

De zoektocht naar inzicht over het gebruik van PQC in praktijksituaties was een interessante reis. Migreren van quantum-kwetsbare cryptografie naar quantum-veilige alternatieven brengt uitdagingen met zich mee. Eén daarvan is interoperabiliteit tussen systemen die migratieklaar zijn en componenten die dat nog niet zijn.

Blijf niet afwachten

Het is makkelijk om vast te lopen in een situatie waarin je moet wachten tot alles klaar is voor migratie, wat soms

‘crypto procrastination’ wordt genoemd: cryptografisch uitstelgedrag. Maar er is urgentie om door te gaan met de PQC-migratie, aangezien quantumcomputers die huidige cryptografie kunnen kraken dichterbij komen. We ontdekten dat alternatieve architectuuropties doorslaggevend kunnen zijn om weer momentum te krijgen in de transitie.

Ontkoppeling is een alternatief om systemen te beschermen die nog geen PQC ondersteunen

In dit PQC Benchmarking-project hebben we een deel van de cryptografie losgekoppeld van de applicatie. Hoewel onze applicatie nog niet klaar was, konden we met een PQC reverse proxy op dezelfde server de gevoelige data op het netwerk beschermen en het risico beperken. Daardoor konden we ook een benchmarkscript ontwikkelen, testen en uitvoeren voor bijna alle cryptografische testcases die we hadden gedefinieerd. Naast het oplossen van het probleem van wachten op een leverancier, geeff dit de mogelijkheid om snel de gebruikte algoritmes te updaten. Deze flexibiliteit, vaak aangeduid als crypto-agility of cryptografische wendbaarheid, is zeer wenselijk voor cryptografische systemen.

De impact op de prestaties varieerde, maar was over het algemeen minder dan verwacht

We richtten ons op berichtverwerking als prestatiemetriek in een MQ-scenario. Hoe verhouden de PQC-opties voor sleuteluitwisseling en digitale handtekeningen zich tot de huidige DH/RSA en ECC-algoritmes? Uit onze benchmark blijkt dat ML-KEM en FrodoKEM vergelijkbaar presteren met klassieke ECC. Zelfs de hybride combinatie van deze KEMs met ECC lijkt geen grote prestatieverlies te geven.

Experimenten helpen om ervaring op te doen en geven duidelijkheid over je opties

SLH-DSA heeft als voordeel dat het zeer conservatieve veiligheidsaannames heeft. Cryptografen begrijpen de veiligheid ervan zo goed dat ze er net zoveel vertrouwen in hebben als in ECC en DH/RSA. Daardoor hoeft het in de meeste aanbevelingen niet gecombineerd te worden met klassieke oplossingen in een hybride configuratie, wat gunstig kan zijn voor prestaties. Onze resultaten tonen echter aan dat SLH-DSA slechter presteert, zelfs vergeleken met hybride configuraties van klassieke en post-quantum DSAs, wat het minder praktisch maakt. Of dit verlies aan prestatie acceptabel is ligt aan de toepassing.

Al met al voelde de reis soms als een achtbaan, maar we zijn blij dat we meer hebben geleerd over de afwegingen bij post-quantum algoritmes en architectuuroplossingen bij blokkerende leveranciersafhankelijkheid. We hopen dat we je hebben geïnspireerd om je eigen benchmarks uit te voeren binnen echte IT-infrastructuur, zodat we samen soepel cryptografische weerbaarheid kunnen bereiken.

Wil je de technische details weten? Zie hier de bijlage van de blog.

Houd de andere blogseries in de gaten! I

In deze reeks van vier blogs delen we zowel organisatorische als technische resultaten, vanuit vier perspectieven: *Management*, *Developers*, *IT Architecten* en *Vendoren*.

Disclaimer: de eerste afbeelding is gemaakt door middel van AI, en AI heeft ondersteund in de vertaling van Engels naar Nederlands.

Alleen door samenwerking kunnen we de beste resultaten behalen in de strijd tegen cybercriminaliteit