Post-quantum Cryptographic Analysis of SSH

Varun Maram

Cybersecurity Group

SandboxAQ

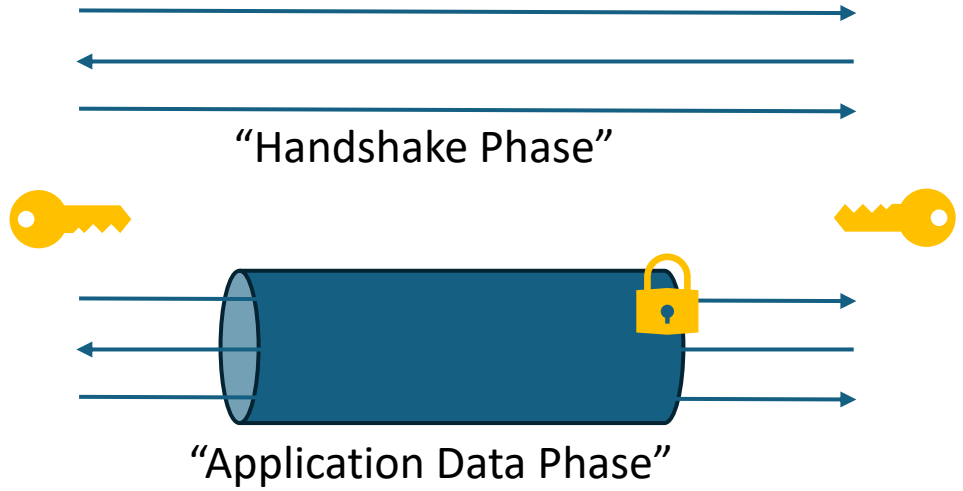

: https://varun-maram.github.io/


: varun-maram-pqc

Joint work with Benjamin Benčina, Benjamin Dowling, and Keita Xagawa

SSH Protocol

Client



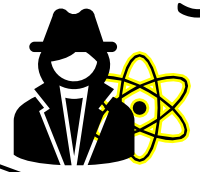
Server

2. Secure Shell (SSH)

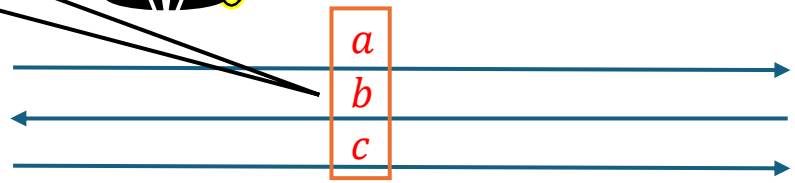
- **Use Case**: Secure remote access to servers and encrypted file transfers (SCP, SFTP).
- **Description**: Uses public-key cryptography and symmetric encryption to secure login sessions over an unsecured network.

SSH Protocol

Diffie-Hellman key-exchange, not secure against quantum attackers.



Client



"Handshake Phase"

"Application Data Phase"

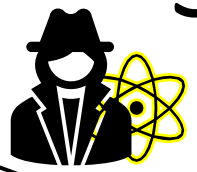
Server

2. Secure Shell (SSH)

- **Use Case**: Secure remote access to servers and encrypted file transfers (SCP, SFTP).
- **Description**: Uses public-key cryptography and symmetric encryption to secure login sessions over an unsecured network.

SSH Protocol

Diffie-Hellman key-exchange, not secure against quantum attackers.



"Hybridize" with (plausibly) quantum-secure KEM.

Client

aA
bB
cC

"Handshake Phase"

"Application Data Phase"

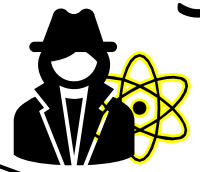
Server

2. Secure Shell (SSH)

- **Use Case**: Secure remote access to servers and encrypted file transfers (SCP, SFTP).
- **Description**: Uses public-key cryptography and symmetric encryption to secure login sessions over an unsecured network.

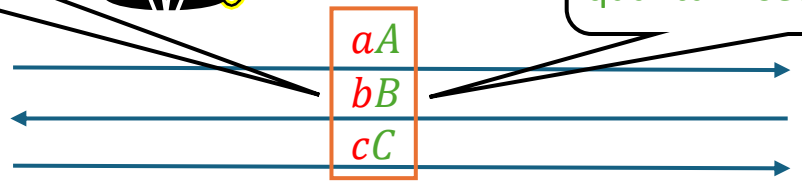
SSH Protocol

Diffie-Hellman key-exchange, not secure against quantum attackers.



"Hybridize" with (plausibly) quantum-secure KEM.

Client



"Handshake Phase"

Server

History of PQC in SSH

- **2018/03** OpenSSH adds experimental support for XMSS signatures. Disabled by default.
- **2018/12** TinySSH added support for hybrid Streamlined NTRU Prime / X25519 KEM `sntrup4591761x25519-sha512`
- **2019/01** OpenSSH added interoperable implementation labeled as experimental
- **2020/12** OpenSSH replaces implementation with `sntrup761x25519-sha512`
- **2021/11** OpenSSH includes `sntrup761x25519-sha512` in the default client/server algorithms proposal
- **2022/02** OpenSSH promotes this algorithm the highest-preference position

2. Secure Sl

- Use Case:
- Descriptio

an unsecu

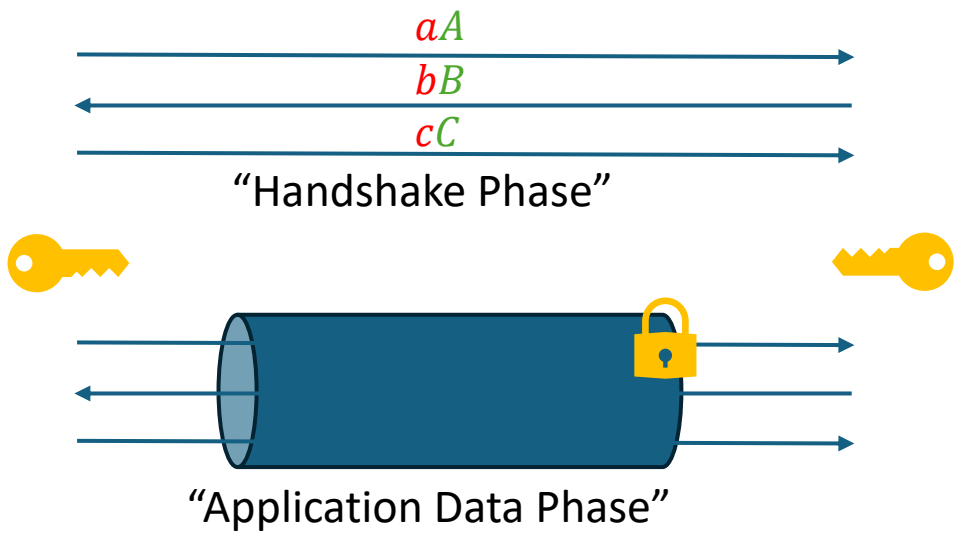
).

gin sessions over

D. Miller, *"OpenSSH PQC: Past, Present, Future"*, RWPQC 2024

Prior Analyses

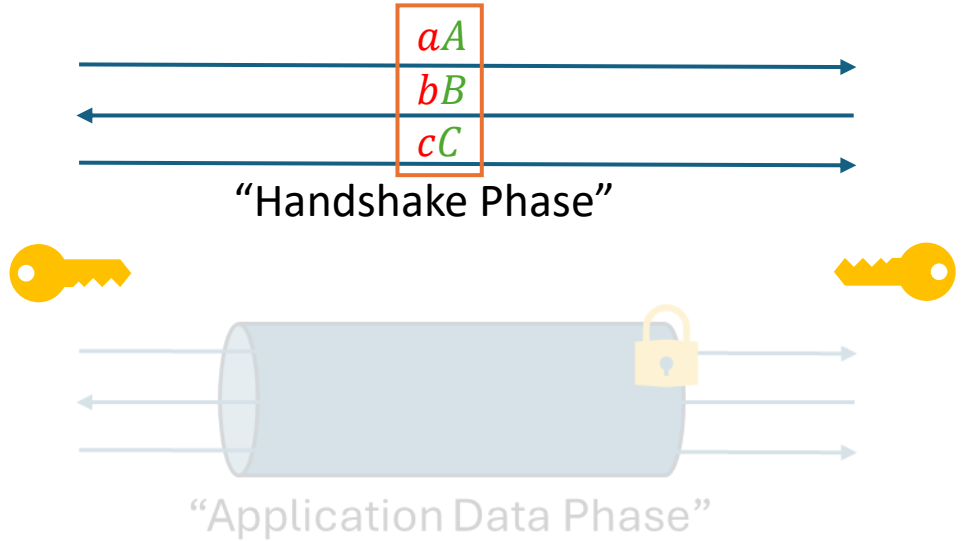
Client



Server

Prior Analyses

Client



Server

Security of Hybrid Key Establishment using Concatenation

Adam Petcher and Matthew Campagna

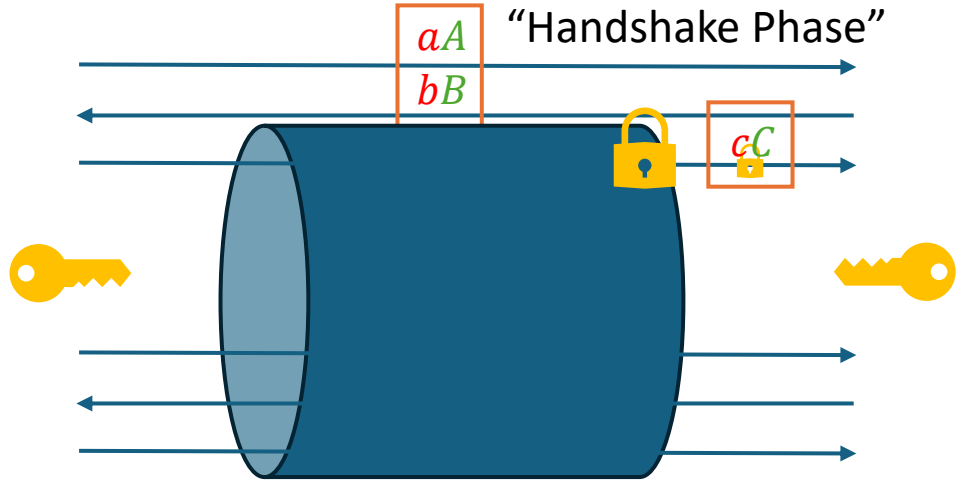
Amazon Web Services

[ePrint '23]

- Prior works either analyze the hybrid key exchange in isolation, or…

Prior Analyses

Client



Server

Security of Hybrid Key Establishment using

Post-quantum sound CryptoVerif and verification of hybrid TLS and SSH key-exchanges

Bruno Blanchet
Inria, F-75012 Paris, France
bruno.blanchet@inria.fr

Charlie Jacomme
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
charlie.jacomme@inria.fr

[CSF '24]

- Prior works either analyze the hybrid key exchange in isolation, or…
- … analyze post-quantum SSH in unsuitable protocol models, such as "authenticated key exchange (AKE)".

Our Contributions

Our Contributions

#1: “Post-quantum SSH is cryptographically secure in practice.”

Post-quantum Cryptographic Analysis of SSH

[S&P ‘25]

Benjamin Benčina
Royal Holloway, University of London, UK
Email: benjamin.bencina.2022@live.rhul.ac.uk

Benjamin Dowling
King’s College London, UK
Email: benjamin.dowling@kcl.ac.uk

Varun Maram
SandboxAQ, UK
Email: varun.maram@sandboxaq.com

Keita Xagawa
Technology Innovation Institute, UAE
Email: keita.xagawa@tii.ae

Our Contributions

#1: "Post-quantum SSH is cryptographically secure in practice."*

(*Analysis needs to assume a fix to the Terrapin Attack in the classical setting.)

Post-quantum Cryptographic Analysis of SSH

[S&P '25]

Benjamin Benčina
Royal Holloway, University of London, UK
Email: benjamin.bencina.2022@live.rhul.ac.uk

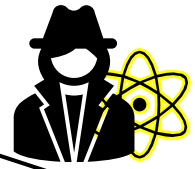
Benjamin Dowling
King's College London, UK
Email: benjamin.dowling@kcl.ac.uk

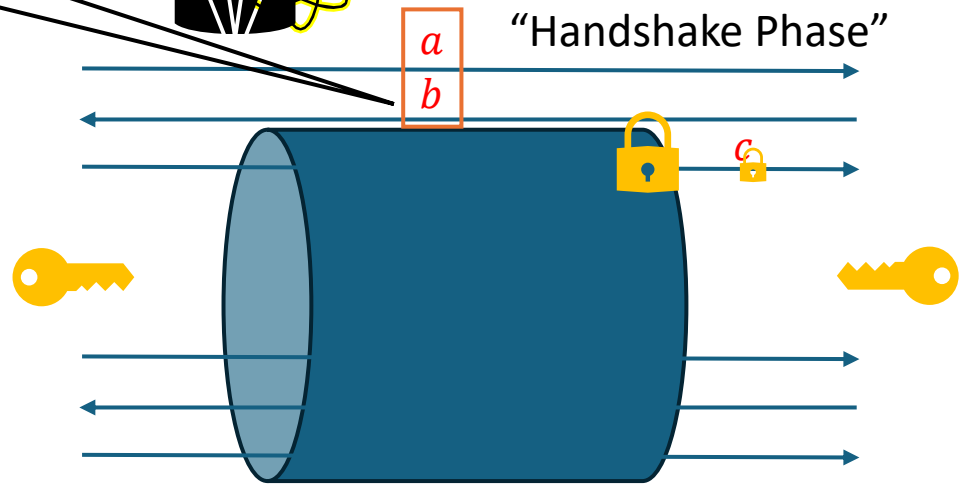
Varun Maram
SandboxAQ, UK
Email: varun.maram@sandboxaq.com

Keita Xagawa
Technology Innovation Institute, UAE
Email: keita.xagawa@tii.ae

Our Contributions

Diffie-Hellman key-exchange, not secure against quantum attackers.





Client

Server

Multi-ciphersuite security of the Secure Shell (SSH) protocol

Florian Bergsma[1] Benjamin Dowling[2a] Florian Kohlar[1] Jörg Schwenk[1]
Douglas Stebila[2a,2b]

[CCS ‘14]

[1] *Horst Görtz Institute, Ruhr-Universität Bochum, Bochum, Germany*
{florian.bergsma,florian.kohlar,joerg.schwenk}@rub.de
[2a] *School of Electrical Engineering and Computer Science*
[2b] *School of Mathematical Sciences*
[2a,2b] *Queensland University of Technology, Brisbane, Australia*
{b1.dowling,stebila}@qut.edu.au

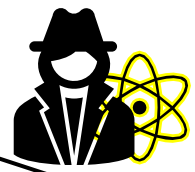
- [BDKSS14] proves security of SSH in the “authenticated and confidential channel establishment (ACCE)” model.
- However, analysis is in the classical Diffie-Hellman setting.

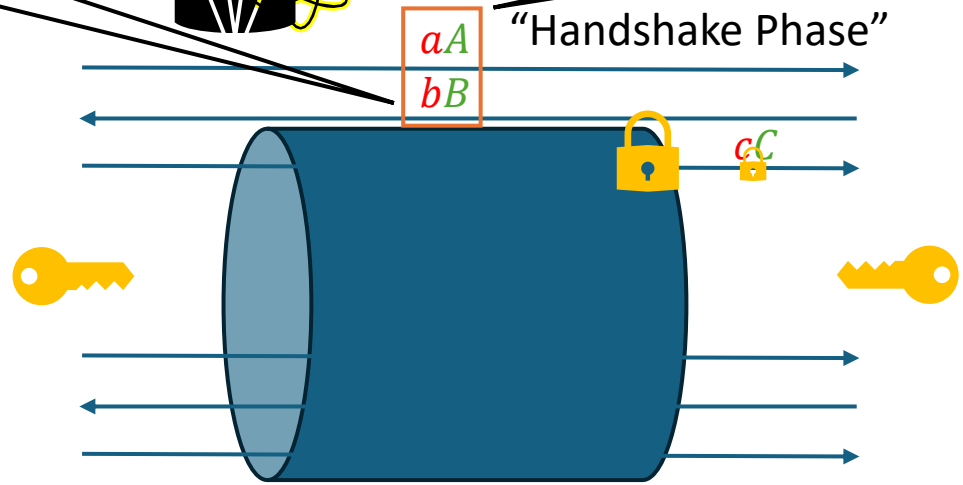
[BDKSS14]: F. Bergsma, B. Dowling, F. Kohlar, J. Schwenk, D. Stebila, *“Multi-ciphersuite Security of the Secure Shell (SSH) Protocol”*, CCS 2014

Our Contributions

Diffie-Hellman key-exchange, not secure against quantum attackers.

"Hybridize" with (plausibly) quantum-secure KEM.





Client

Server

Multi-ciphersuite security

Post-quantum Cryptographic Analysis of SSH

[S&P '25]

Benjamin Benčina
Royal Holloway, University of London, UK
Email: benjamin.bencina.2022@live.rhul.ac.uk

Benjamin Dowling
King's College London, UK
Email: benjamin.dowling@kcl.ac.uk

Varun Maram
SandboxAQ, UK
Email: varun.maram@sandboxaq.com

Keita Xagawa
Technology Innovation Institute, UAE
Email: keita.xagawa@tii.ae

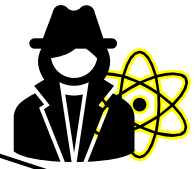
- We establish security of SSH in a PQ-extension of ACCE model that accounts for "harvest now, decrypt later" attacks.
- Analysis also captures forward secrecy (unlike the classical ACCE analysis in [BDKSS14]).

[BDKSS14]: F. Bergsma, B. Dowling, F. Kohlar, J. Schwenk, D. Stebila, *"Multi-ciphersuite Security of the Secure Shell (SSH) Protocol"*, CCS 2014

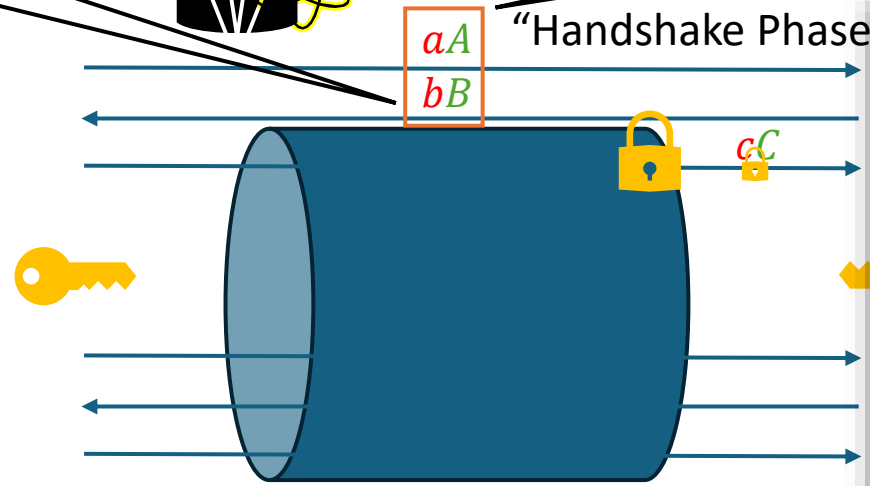
Our Contributions

Diffie-Hellman key-exchange, not secure against quantum attackers.

"Hybridize" with (plausibly) quantum-secure KEM.



Client



Multi-ciphersuite security

Post-quantum Cryptographic Analysis of SSH

[S&P '25]

Benjamin Benčina
Royal Holloway, University of London, UK
Email: benjamin.bencina.2022@live.rhul.ac.uk

Benjamin Dowling
King's College London, UK
Email: benjamin.dowling@kcl.ac.uk

Varun Maram
SandboxAQ, UK
Email: varun.maram@sandboxaq.com

Keita Xagawa
Technology Innovation Institute, UAE
Email: keita.xagawa@tii.ae

- We establish security of SSH in a PQ-extension of ACCE model that accounts for "harvest now, decrypt later" attacks.
- Analysis also captures forward secrecy (unlike the classical ACCE analysis in [BDKSS14]).

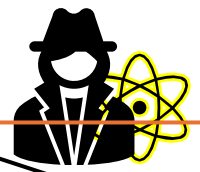
Can't manipulate "sequence numbers" in already finished handshakes.

[BDKSS14]: F. Bergsma, B. Dowling, F. Kohlar, J. Schwenk, D. Stebila, *"Multi-ciphersuite Security of the Secure Shell (SSH) Protocol"*, CCS 2014

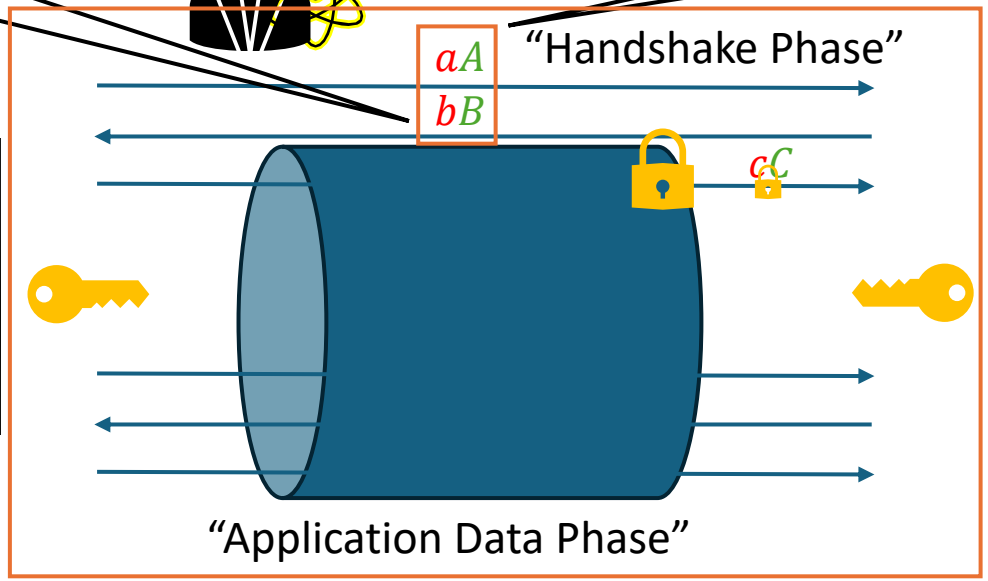
Our Contributions

Diffie-Hellman key-exchange, not secure against quantum attackers.

"Hybridize" with (plausibly) quantum-secure KEM.



Client



"PQ ACCE Security"

Server

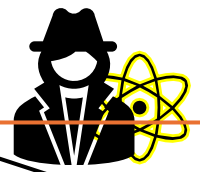
- We establish security of SSH in a PQ-extension of ACCE model that accounts for "harvest now, decrypt later" attacks.
- Analysis also captures forward secrecy (unlike the classical ACCE analysis in [BDKSS14]).

[BDKSS14]: F. Bergsma, B. Dowling, F. Kohlar, J. Schwenk, D. Stebila, *"Multi-ciphersuite Security of the Secure Shell (SSH) Protocol"*, CCS 2014

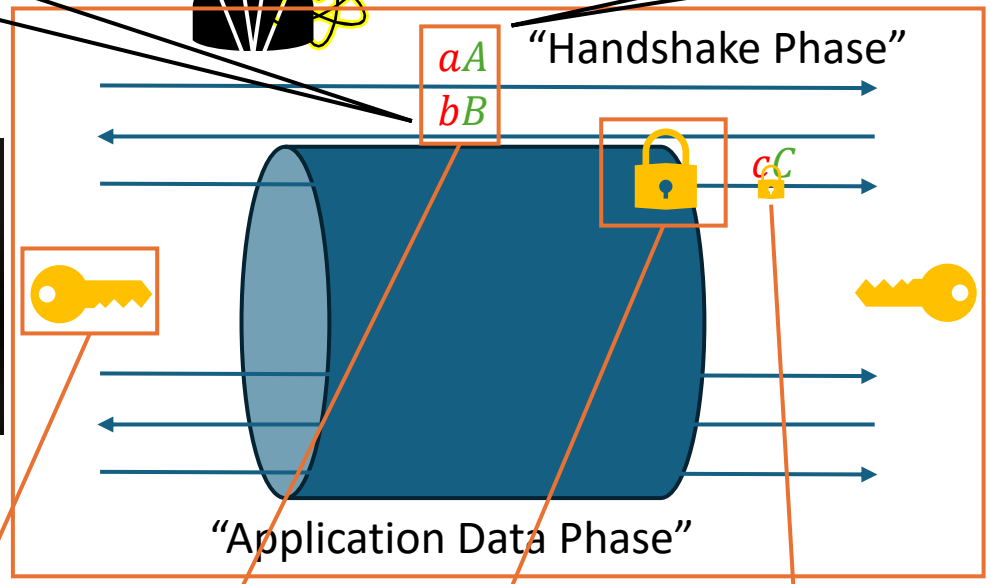
Our Contributions

Diffie-Hellman key-exchange, not secure against quantum attackers.

"Hybridize" with (plausibly) quantum-secure KEM.



Client



Server

"PQ ACCE Security"

PRF	KEM	Symmetric Encryption	Digital Signature
"Pseudo-random"	"IND-CPA" (passive security)	"BSAE" (Buffered Stateful Authenticated Encryption)	"EUF-CMA" (existentially unforgeable)

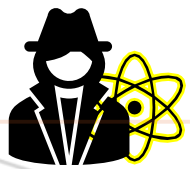
- We establish security of SSH in a PQ-extension of ACCE model that accounts for "harvest now, decrypt later" attacks.
- Analysis also captures forward secrecy (unlike the classical ACCE analysis in [BDKSS14]).

[BDKSS14]: F. Bergsma, B. Dowling, F. Kohlar, J. Schwenk, D. Stebila, *"Multi-ciphersuite Security of the Secure Shell (SSH) Protocol"*, CCS 2014

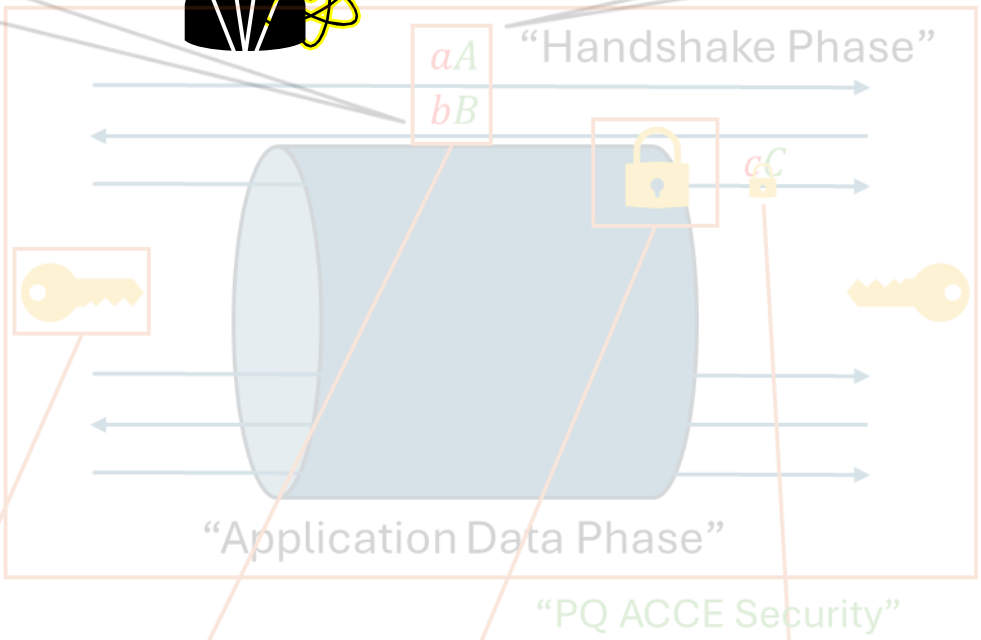
Our Contributions

Diffie-Hellman key-exchange, not secure against quantum attackers.

"Hybridize" with (plausibly) quantum-secure KEM.



Client



Server

PRF	KEM	Symmetric Encryption	Digital Signature
"Pseudo-random"	"IND-CPA" (passive security)	"BSAE" (Buffered Stateful Authenticated Encryption)	"EUF-CMA" (existentially unforgeable)

- We establish security of SSH in a PQ-extension of ACCE model that accounts for "harvest now, decrypt later" attacks.
- Analysis also captures forward secrecy (unlike the classical ACCE analysis in [BDKSS14]).
- We then prove corresponding PQ security properties of SSH primitives.

[BDKSS14]: F. Bergsma, B. Dowling, F. Kohlar, J. Schwenk, D. Stebila, *"Multi-ciphersuite Security of the Secure Shell (SSH) Protocol"*, CCS 2014

Our Contributions

#2: “Post-quantum SSH can be more efficient, and eco(log|nom)ical.”

Post-quantum Cryptographic Analysis of SSH

[S&P ‘25]

Benjamin Benčina
Royal Holloway, University of London, UK
Email: benjamin.bencina.2022@live.rhul.ac.uk

Benjamin Dowling
King’s College London, UK
Email: benjamin.dowling@kcl.ac.uk

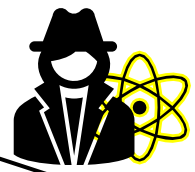
Varun Maram
SandboxAQ, UK
Email: varun.maram@sandboxaq.com

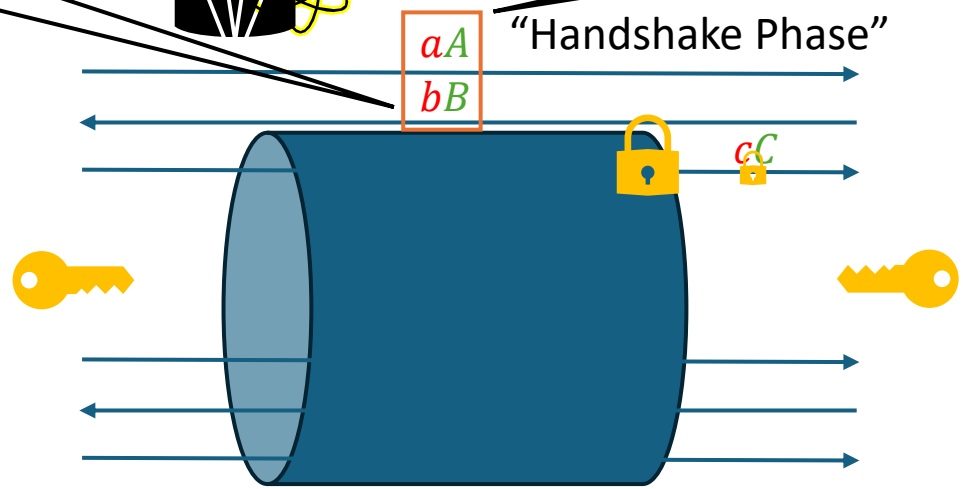
Keita Xagawa
Technology Innovation Institute, UAE
Email: keita.xagawa@tii.ae

Our Contributions

Diffie-Hellman key-exchange, not secure against quantum attackers.

"Hybridize" with (plausibly) quantum-secure KEM.





Client

Server

Post-quantum sound CryptoVerif and verification of hybrid TLS and SSH key-exchanges

Theorem 5 (Post-quantum SSH security, simplified) *Under the post-quantum IND-CCA2 assumption for the KEM, the post-quantum PRF assumption for* dPRF *and* PRFc*, the CR assumption for h, the classical EUF-CMA assumption for signatures, the hybrid SSH key exchange ensures forward secrecy even against quantum attackers, provided quantum attackers do not exist yet when the handshake is performed.*

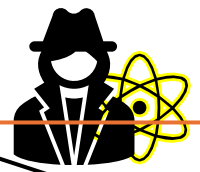
- PQ-SSH analysis of [BlaJac24] relies on IND-CCA secure (ephemeral) KEMs.

[BlaJac24]: B. Blanchet, C. Jacomme, *"Post-quantum sound CryptoVerif and verification of hybrid TLS and SSH key-exchanges"*, CSF 2024

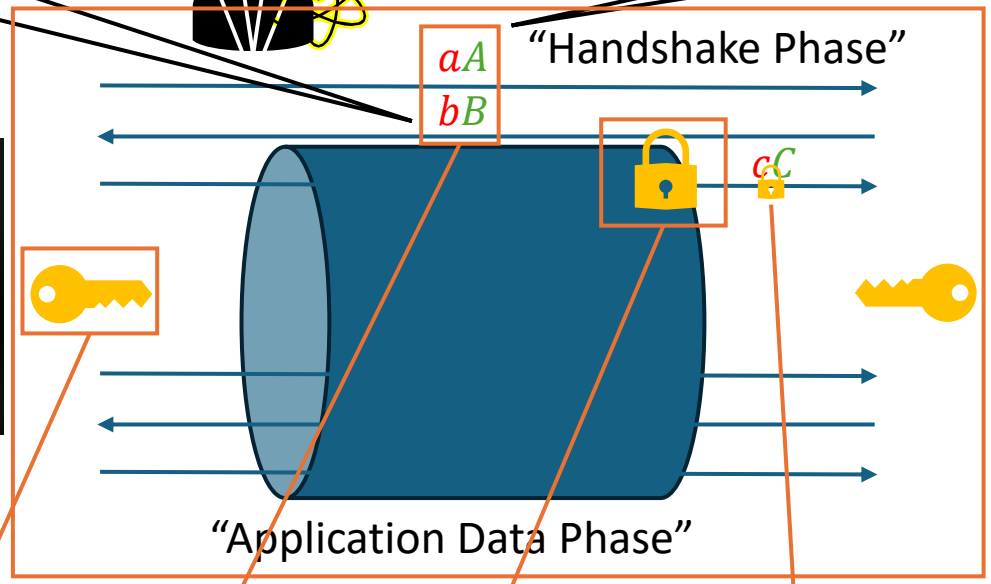
Our Contributions

Diffie-Hellman key-exchange, not secure against quantum attackers.

"Hybridize" with (plausibly) quantum-secure KEM.



Client



Server

"PQ ACCE Security"

PRF	KEM	Symmetric Encryption	Digital Signature
"Pseudo-random"	"IND-CPA" (passive security)	"BSAE" (Buffered Stateful Authenticated Encryption)	"EUF-CMA" (existentially unforgeable)

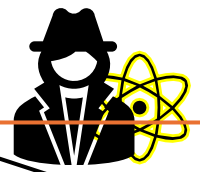
- PQ-SSH analysis of [BlaJac24] relies on IND-CCA secure (ephemeral) KEMs.

[BlaJac24]: B. Blanchet, C. Jacomme, *"Post-quantum sound CryptoVerif and verification of hybrid TLS and SSH key-exchanges"*, CSF 2024

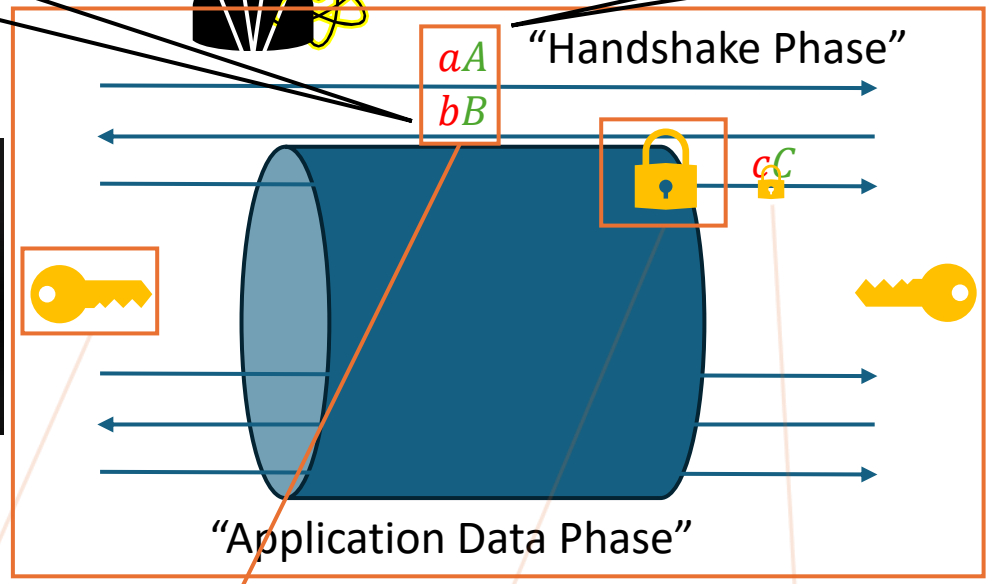
Our Contributions

Diffie-Hellman key-exchange, not secure against quantum attackers.

"Hybridize" with (plausibly) quantum-secure KEM.



Client



Server

"PQ ACCE Security"

PRF — "Pseudo-random"

KEM — "IND-CPA" (passive security)

Symmetric Encryption — "BSAE" (Buffered Stateful Authenticated Encryption)

Digital Signature — "EUF-CMA" (existentially unforgeable)

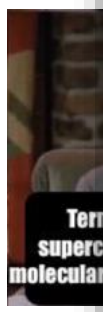
- PQ-SSH analysis of [BlaJac24] relies on IND-CCA secure (ephemeral) KEMs.
- Whereas our analysis relies on the weaker property of IND-CPA security.

[BlaJac24]: B. Blanchet, C. Jacomme, *"Post-quantum sound CryptoVerif and verification of hybrid TLS and SSH key-exchanges"*, CSF 2024

Our Contributions

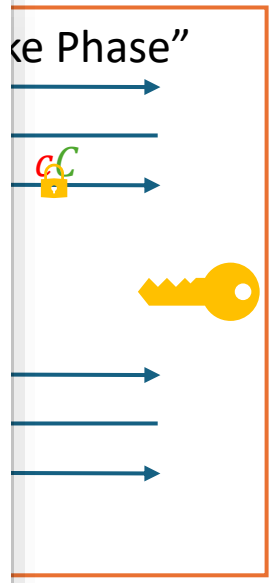
Diffie-Hellman key-exchange, not secure against quantum attackers.

"Hybridize" with (plausibly) quantum-secure KEM.



History of PQC in SSH

- **2018/03** OpenSSH adds experimental support for XMSS signatures. Disabled by default.
- **2018/12** TinySSH added support for hybrid Streamlined NTRU Prime / X25519 KEM `sntrup4591761x25519-sha512`
- **2019/01** OpenSSH added interoperable implementation labeled as experimental
- **2020/12** OpenSSH replaces implementation with `sntrup761x25519-sha512`
- **2021/11** OpenSSH includes `sntrup761x25519-sha512` in the default client/server algorithms proposal
- **2022/02** OpenSSH promotes this algorithm the highest-preference position



Server

KEM → "Fujisaki-Okamoto" transform → KEM

"IND-CPA" (passive security)

"IND-CCA" (active security)

- PQ-SSH analysis of [BlaJac24] relies on IND-CCA secure (ephemeral) KEMs.
- Whereas our analysis relies on the weaker property of IND-CPA security.
- Our analysis suggests FO transform is not needed, which in turn can lead to performance improvements in PQ-SSH.

D. Miller, *"OpenSSH PQC: Past, Present, Future"*, RWPQC 2024
[BlaJac24]: B. Blanchet, C. Jacomme, *"Post-quantum sound CryptoVerif and verification of hybrid TLS and SSH key-exchanges"*, CSF 2024

Our Contributions

Table 1. IND-CCA vs IND-CPA benchmarks w.r.t. ephemeral KEMs in post-quantum SSH.

Scope	KEM	IND-CCA[s]	IND-CPA[s]	Speedup[%]
Primitive-level (CPU timings of all KEM operations)	sntrup761	$2.8164 \cdot 10^{-2}$	$2.7056 \cdot 10^{-2}$	3.93
	mlkem768	$2.7853 \cdot 10^{-5}$	$1.3242 \cdot 10^{-5}$	52.46
	sntrup761x25519-sha512	$3.0290 \cdot 10^{-2}$	$2.9105 \cdot 10^{-2}$	3.91
	mlkem768x25519-sha256	$3.1412 \cdot 10^{-3}$	$3.1015 \cdot 10^{-3}$	1.26
Protocol-level (Networks timings of an SSH connection)	sntrup761x25519-sha512	0.1565	0.1534	1.98
	mlkem768x25519-sha256	0.1325	0.1316	0.68

Our Contributions

Table 1. IND-CCA vs IND-CPA benchmarks w.r.t. ephemeral KEMs in post-quantum SSH.

Scope	KEM	IND-CCA[s]	IND-CPA[s]	Speedup[%]
Primitive-level (CPU timings of all KEM operations)	sntrup761	$2.8164 \cdot 10^{-2}$	$2.7056 \cdot 10^{-2}$	3.93
	mlkem768	$2.7853 \cdot 10^{-5}$	$1.3242 \cdot 10^{-5}$	52.46
	sntrup761x25519-sha512	$3.0290 \cdot 10^{-2}$	$2.9105 \cdot 10^{-2}$	3.91
	mlkem768x25519-sha256	$3.1412 \cdot 10^{-3}$	$3.1015 \cdot 10^{-3}$	1.26
Protocol-level (Networks timings of an SSH connection)	sntrup761x25519-sha512	0.1565	0.1534	1.98
	mlkem768x25519-sha256	0.1325	0.1316	0.68

Measurement w.r.t. a single SSH connection.

Small performance gains should accumulate in large scale SSH deployments.

Our Contributions

Table 1. IND-CCA vs IND-CPA benchmarks w.r.t. ephemeral KEMs in post-quantum SSH.

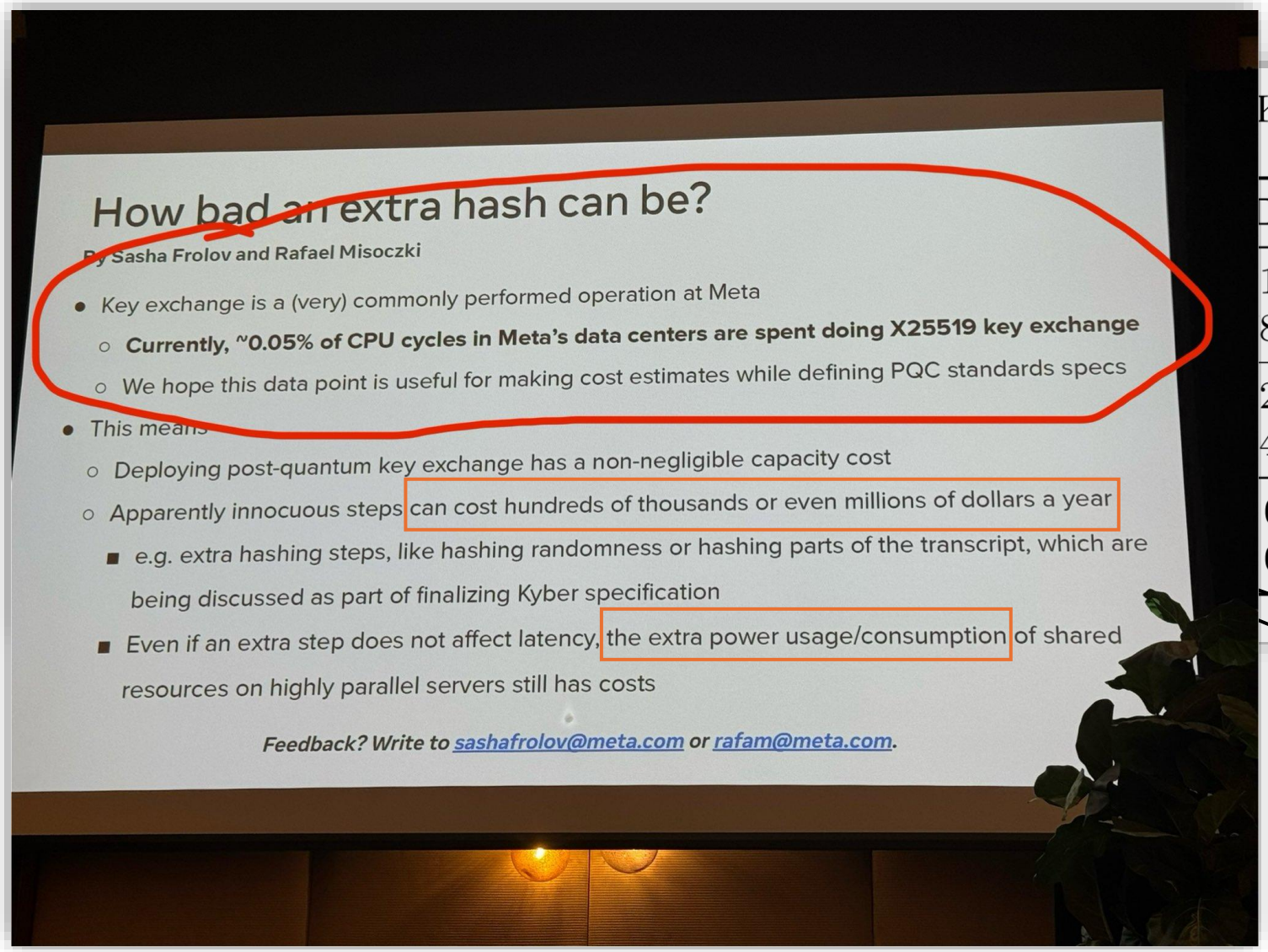
Scope	KEM	IND-CCA[s]	IND-CPA[s]	Speedup[%]
Primitive-level (CPU timings of all KEM operations)	sntrup761	$2.8164 \cdot 10^{-2}$	$2.7056 \cdot 10^{-2}$	3.93
	mlkem768	$2.7853 \cdot 10^{-5}$	$1.3242 \cdot 10^{-5}$	52.46
	sntrup761x25519-sha512	$3.0290 \cdot 10^{-2}$	$2.9105 \cdot 10^{-2}$	3.91
	mlkem768x25519-sha256	$3.1412 \cdot 10^{-3}$	$3.1015 \cdot 10^{-3}$	1.26
Protocol-level (Networks timings of an SSH connection)	sntrup761x25519-sha512	0.1565	0.1534	1.98
	mlkem768x25519-sha256	0.1325	0.1316	0.68

Measurement w.r.t. a single SSH connection.

Small performance gains should accumulate in large scale SSH deployments.

IND-CCA → IND-CPA leads to ≈80% reduction in hash computations.

Our Contributions



KEMs in post-quantum SSH.

D-CCA[s]	IND-CPA[s]	Speedup[%]
$164 \cdot 10^{-2}$	$2.7056 \cdot 10^{-2}$	3.93
$853 \cdot 10^{-5}$	$1.3242 \cdot 10^{-5}$	52.46
$290 \cdot 10^{-2}$	$2.9105 \cdot 10^{-2}$	3.91
$412 \cdot 10^{-3}$	$3.1015 \cdot 10^{-3}$	1.26
0.1565	0.1534	1.98
0.1325	0.1316	0.68

IND-CCA → IND-CPA leads to ≈80% reduction in hash computations.

S. Frolov, R. Misoczki, *"Meta PQC Updates"*, RWPQC 2024 (https://x.com/bwesterb/status/1771958142147973390)

Post-quantum Cryptographic Analysis of SSH

Varun Maram
Cybersecurity Group
SandboxAQ

: https://varun-maram.github.io/

: varun-maram-pqc

Joint work with Benjamin Benčina, Benjamin Dowling, and Keita Xagawa

[Full version of paper: *https://eprint.iacr.org/2025/684.pdf*]