INTERNATIONALE
HOCHSCHULE

SECURWARE 2025

JÖRN-MARC SCHMIDT AND ALEXANDER LAWALL

COMPARISON OF PASSWORD-AUTHENTICATED KEY EXCHANGE SCHEMES ON ANDROID

joern-marc.schmidt@iu.org
alexander.lawall@iu.org
IU International University of Applied Sciences
Erfurt, Germany

Jörn-Marc Schmidt

2024-present	*Professor for Cyber Security* IU International University of Applied Sciences, Germany
2018–2023	*Lead Engineer*, Cryptography Engineering and Solutions, Deutsche Bank AG, Eschborn, Germany
2013–2018	*Senior IT-Security Consultant*, secunet Security Networks AG, Eschborn, Germany
2010–2013	G*roup Coordinator*, Institute for Applied Information Processing and Communication (IAIK), Graz University of Technology, Austria
2006–2009	Ph.D. Studies at Graz University of Technology, Graz, Austria
2002–2006	Study at University of Mannheim, Mannheim, Germany

AGENDA

PASSWORD AUTHENTICATED KEY AGREEMENT (PAKE)

Passwords are human-friendly:

- Rememberable
- Easily entered manually

Do not provide the security level of cryptographic keys

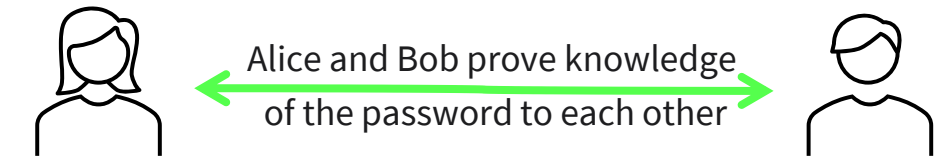
PAKEs combine both worlds

Mobile constraints:
CPU, battery, latency, libraries

Generated with M365 Copilot AI – 08.10.2025 by J-M Schmidt

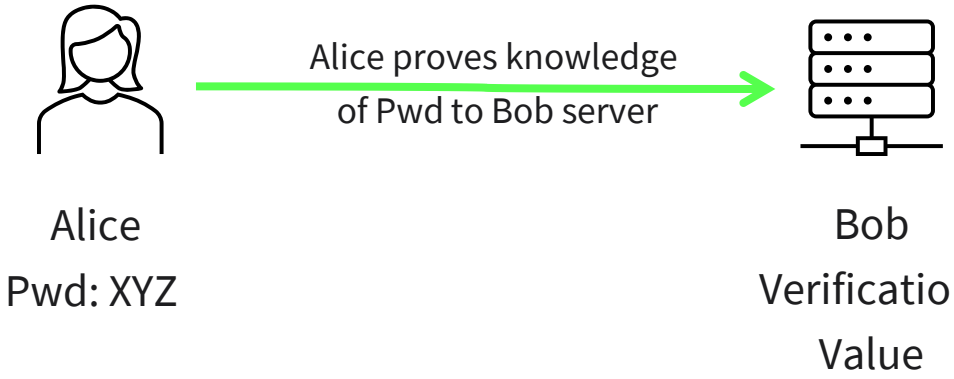
PAKE TYPES

Balanced PAKE Schemes



Mutual Authentication using the password

Augmented PAKE Schemes



Alice proves possession of the password

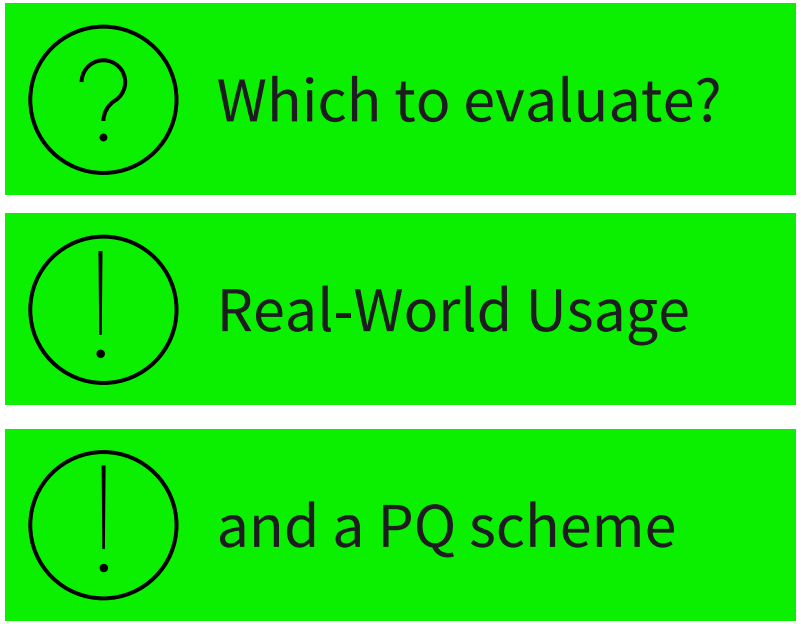
Bob uses a verification value

Bob cannot impersonate Alice to others

Pictograms from MS Archive, 2025

EVALUATED SCHEMES

- Nouri Alnahawi et al. (2025) discuss in a SoK paper:
  - 30 balanced schemes
  - 19 augmented Schemes



- Dragonfly – used in WPA3
- PACE – used in travel documents
- CPACE – used by Facebook Messenger
- OCAKE – as Post-Quantum Scheme and OEKE as comparison

Nouri Alnahawi, , David Haas, Erik Mauß, and Alexander Wiesmaier,2025 - SoK: PQC PAKEs Cryptographic Primitives, Design and Security – eprint, https://eprint.iacr.org/2025/119.pdf

EVALUATED SCHEMES

- Dragonfly – used in WPA3
- PACE – used in travel documents
- CPACE – used by Facebook Messenger
- OCAKE – as Post-Quantum Scheme and OEKE for comparison
- SRP – used in TLS
- SPAKE2+ - used by Apple HomeKit/Car Key and the Matter Protocol and SPAKE2 for comparison
- Transformation of Lyu et al. applied to OCAKE – as PQ scheme

IMPLEMENTATIONS

- Implementations for Android,
  - Using Kotlin/JAVA and
  - Bouncy Castle (Version 1.81)
  - Build for API 34 with ProGuard/R8 minification

PRIMITIVES TO PROVIDE A SECURITY LEVEL OF 128 BITS

Scheme	Group	Cipher	Hash	Mapping	Others
Balanced Schemes					
OEKE	3072-bit MODP Group	AES-CBC	SHA256	-	-
OEKE (ECC)	secp256r1	AES-CBC	SHA256	-	-
OCAKE	ML-KEM512	AES-CBC	SHA256	-	PBKDF2-HMAC-SHA256
Dragonfly	secp256r1	-	SHA256	Hunting and Pecking	HKDF-SHA256
SPAKE2	secp256r1	-	SHA256		HKDF-SHA256
PACE(IM)	secp256r1	AES-CBC	SHA256	Integrated Mapping	PBKDF2-HMAC-SHA256
PACE(GM)	secp256r1	AES-CBC	SHA256	Generic Mapping	PBKDF2-HMAC-SHA256
CPACE	secp256r1	-	SHA256	Integrated Mapping	-
Augmented Schemes					
SRP	3072-bit MODP Group	-	SHA1/SHA256	-	-
SPAKE2+	secp256r1	-	SHA256	-	HKDF-SHA256/scrypt
aPAKE-PQC	ML-KEM512	AES-GCM	SHA256	-	-

MEASUREMENT SETUP

- Google Pixel 7 Pro with
  - Google Tensor G2 SoC
  - 12GB RAM
  - Android 16
- Connected via USB to a Windows PC

During the Tests:

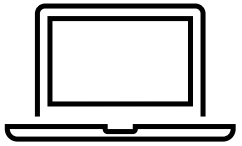
- Airplane mode on
- All connectivity disabled
- Adaptive Battery features / energy saving mode turned off
- Background processes set to zero

Conducted Microbenchmarking using Android Jetpack Benchmark Library (version 1.3.4)

Compute time and device-side energy is evaluated

Network overhead and radio power are not considered

BENCHMARKING



Compilation of Implementation and Tests

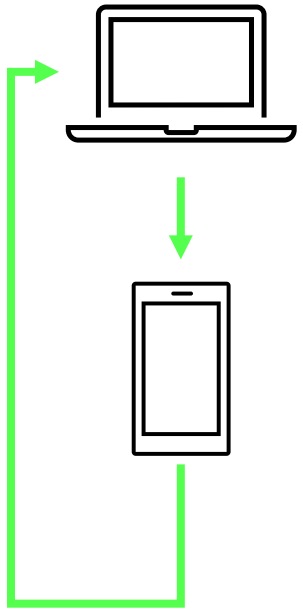
```
benchmarkRule.measureRepeated {
    val password = runWithTimingDisabled { tools.randomStringByKotlinRandom()}

    val clientMSG = client.initParty(password, ChannelID = "channelID", SessionID = "TestSession")
    val serverMSG = runWithTimingDisabled { server.initParty(password, ChannelID = "channelID", SessionID = "TestSession")}

    val clientKey= client.processMessage( YbEnc = serverMSG, server = false)
    val serverKey = runWithTimingDisabled {server.processMessage( YbEnc = clientMSG, server = true)}
    runWithTimingDisabled {
        assertTrue(clientKey.first contentEquals serverKey.first)
        assertTrue(clientKey.second contentEquals serverKey.second)}

}
```

BENCHMARKING



Compilation of Implementation and Tests

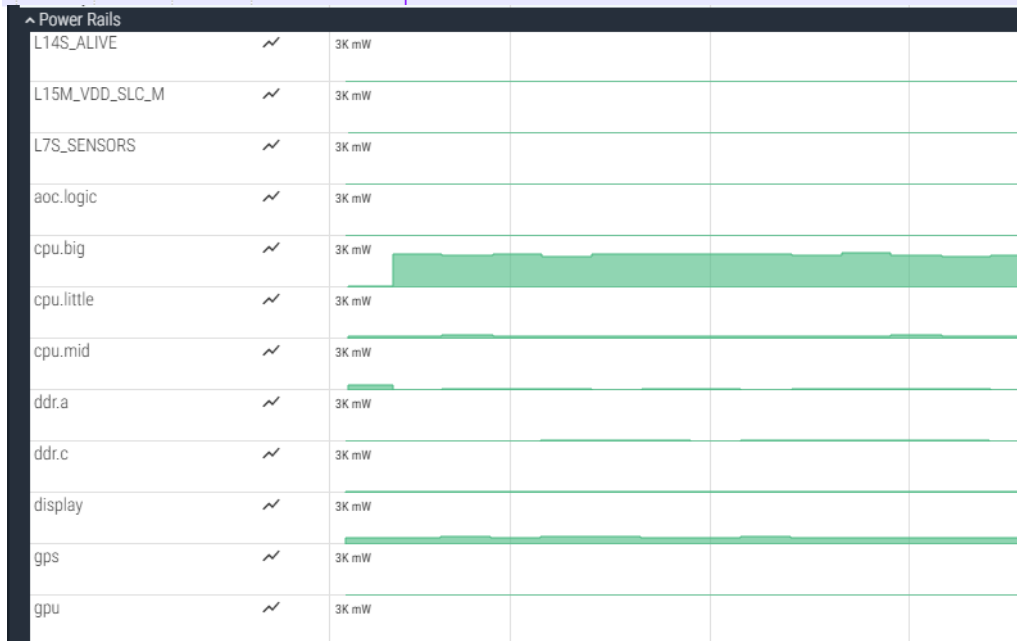
Instrument tests via PowerShell

Microbenchmarking (50 runs per test)
Perfetto Trace per Test

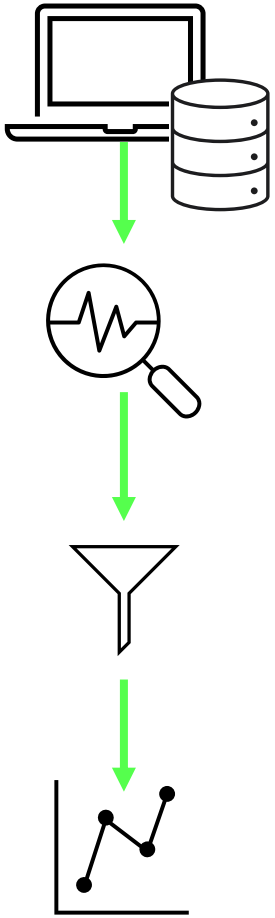
Download JSON file with Results and Perfetto Traces

Repeat 100 times for performance measurements
Repeat 250 times in randomized test order
for energy measurements


```
{
    "name": "mbpbCPACE",
    "params": {},
    "className": "com.libs.powerrails_tests.PowerBe
    "totalRunTimeNs": 10122064620,
    "metrics": {
        "timeNs": {
            "minimum": 1384947.6756756757,
            "maximum": 1469565.4189189188,
            "median": 1444707.1148648649,
            "coefficientOfVariation": 0.016432729711
            "runs": [
                1450385.0135135136,
                1391238.689189189,
                1463612.6621621621,
                ....
```

BENCHMARKING



Processing via Python

Extract timings per run

Extracted value for cpu.big power consumption

Remove outliers

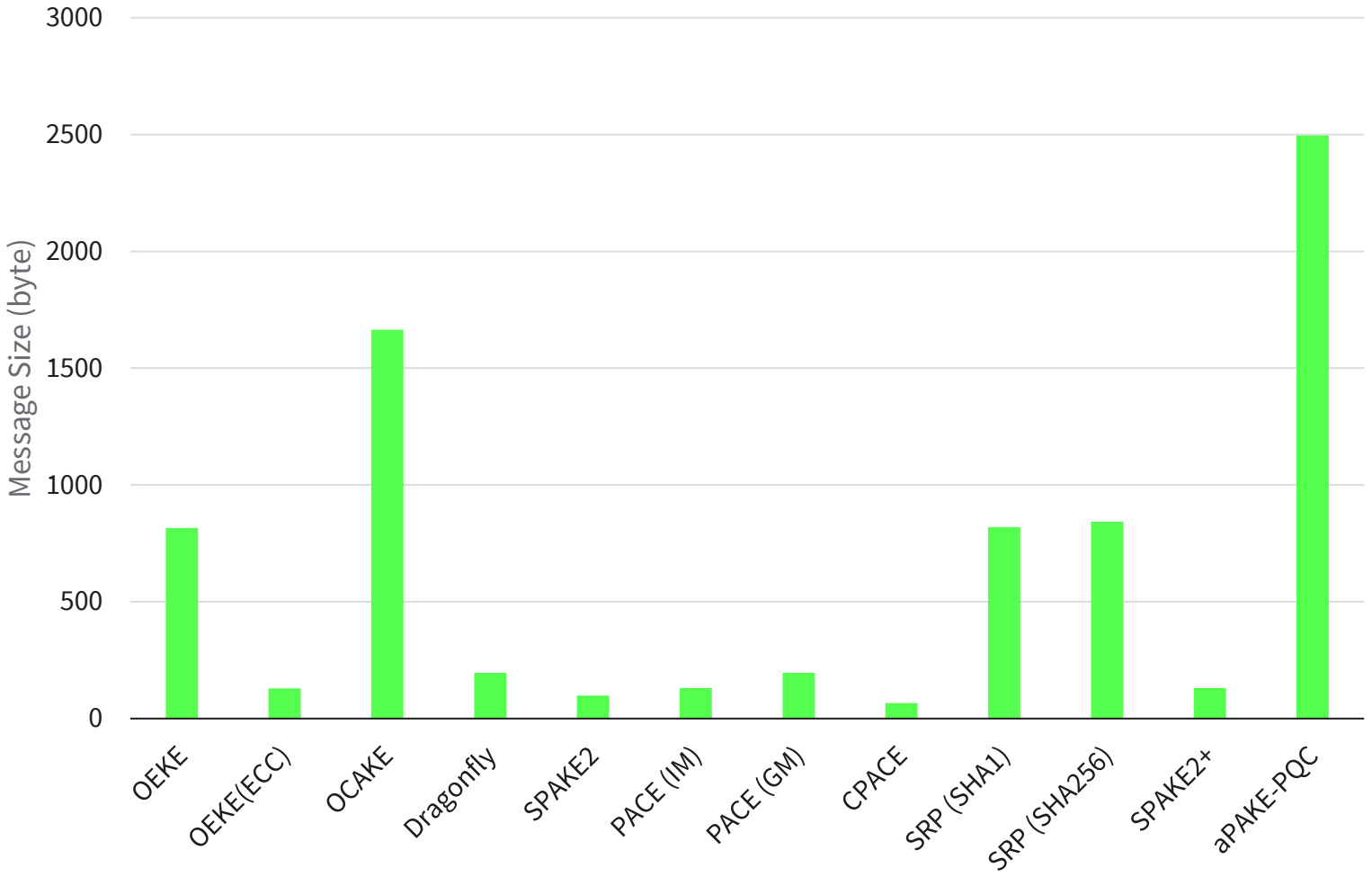
>10x mean value of current measurements for performance

test runs with fewer than 2 energy measurement points

Compute results

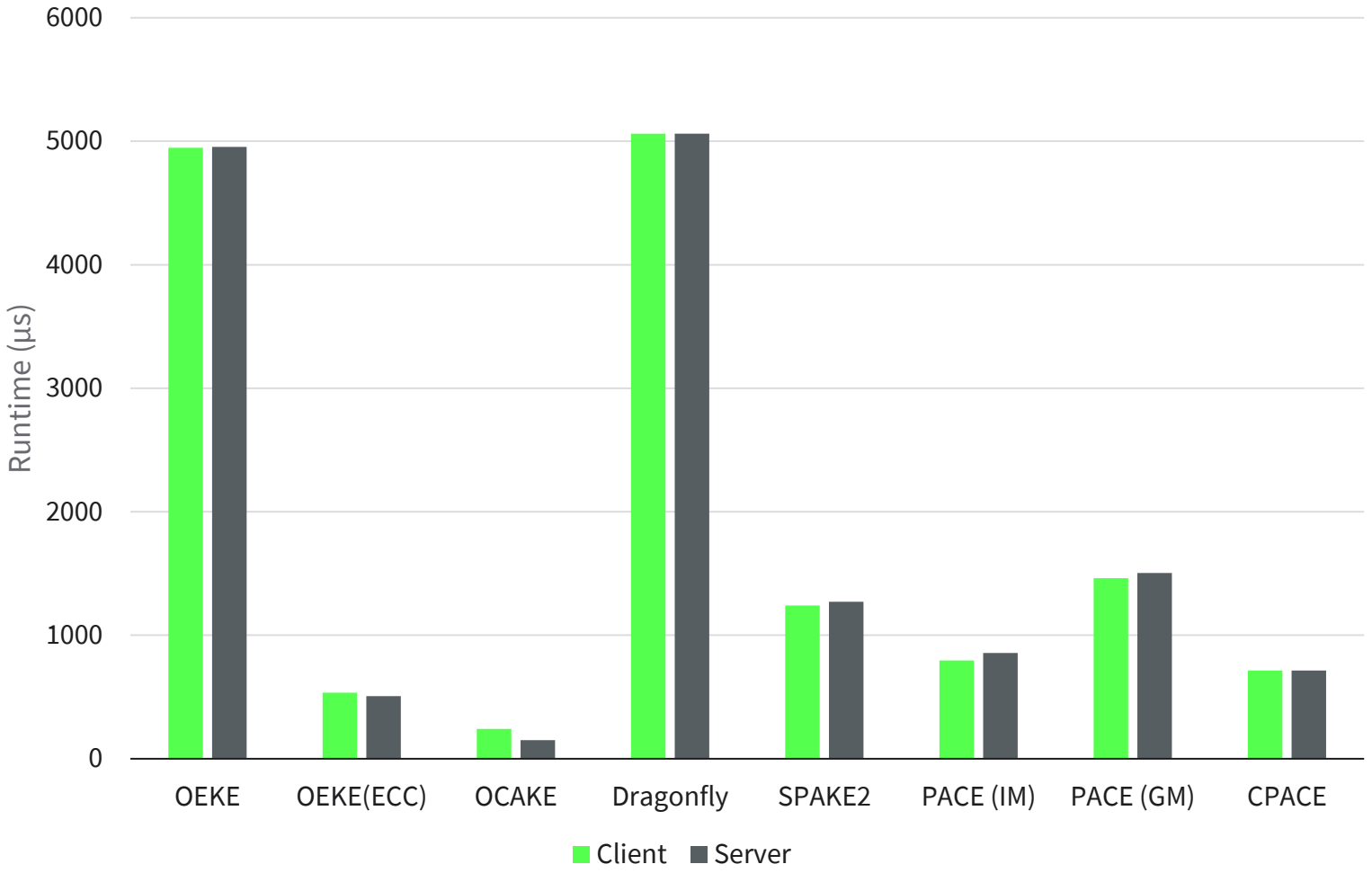
- on average 4,993 data points per performance tests
- 222 energy consumption results per test

EXCHANGED MESSAGES



- Elements are encoded as byte array
- ECC uses compressed points
- AES-encrypted messages include IV
- No further message overhead was added

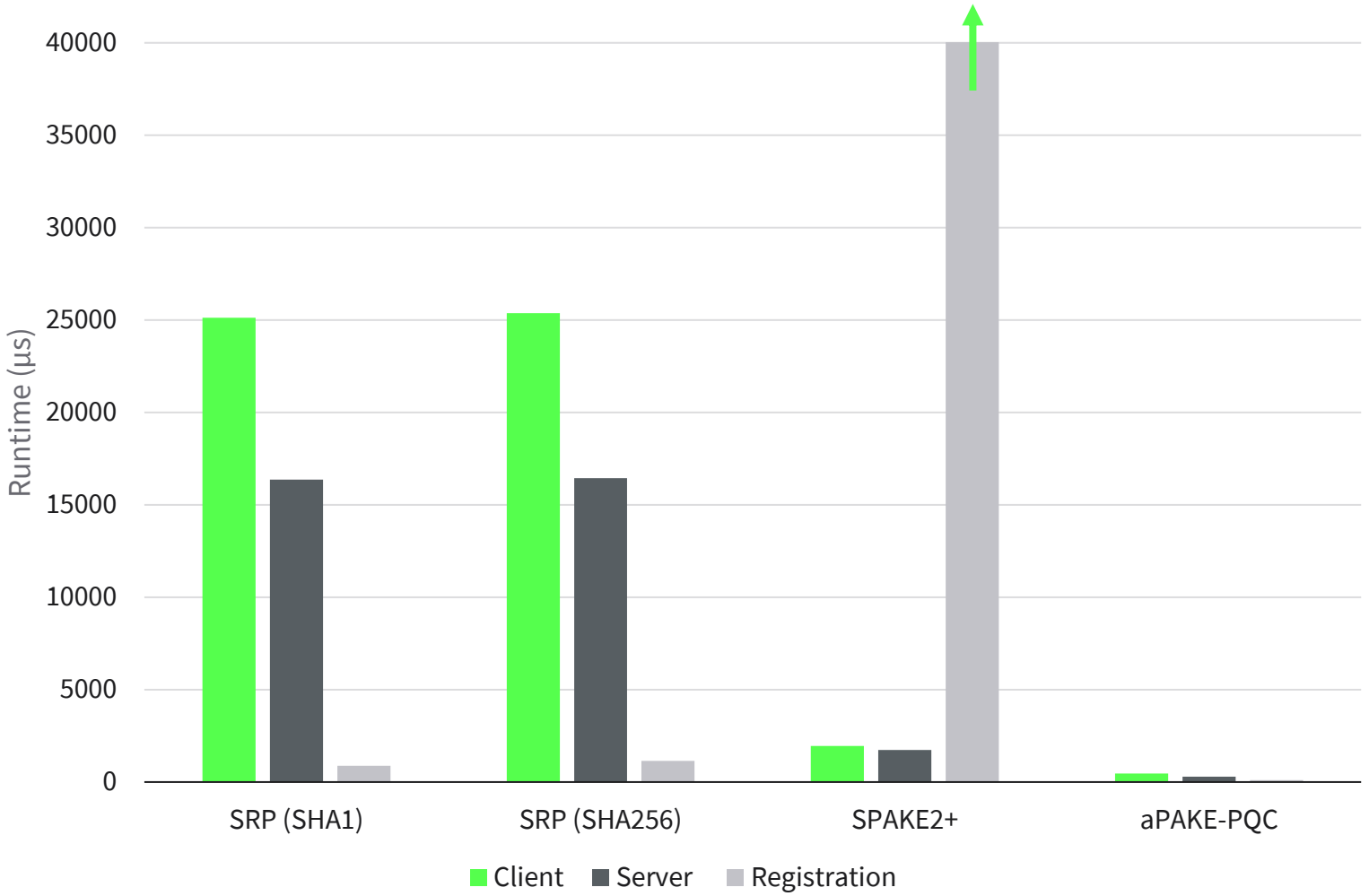
PERFORMANCE RESULTS – BALANCED SCHEMES



Performance impact:

- OEKE uses 3072-bit MODP Group – not secp256r1
- Dragonfly uses constant-time Hunting & Pecking (40 tries); non-constant-time is ~4× faster but not recommended
- Integrated Mapping is more efficient than Generic Mapping, plus figures ignore network latency

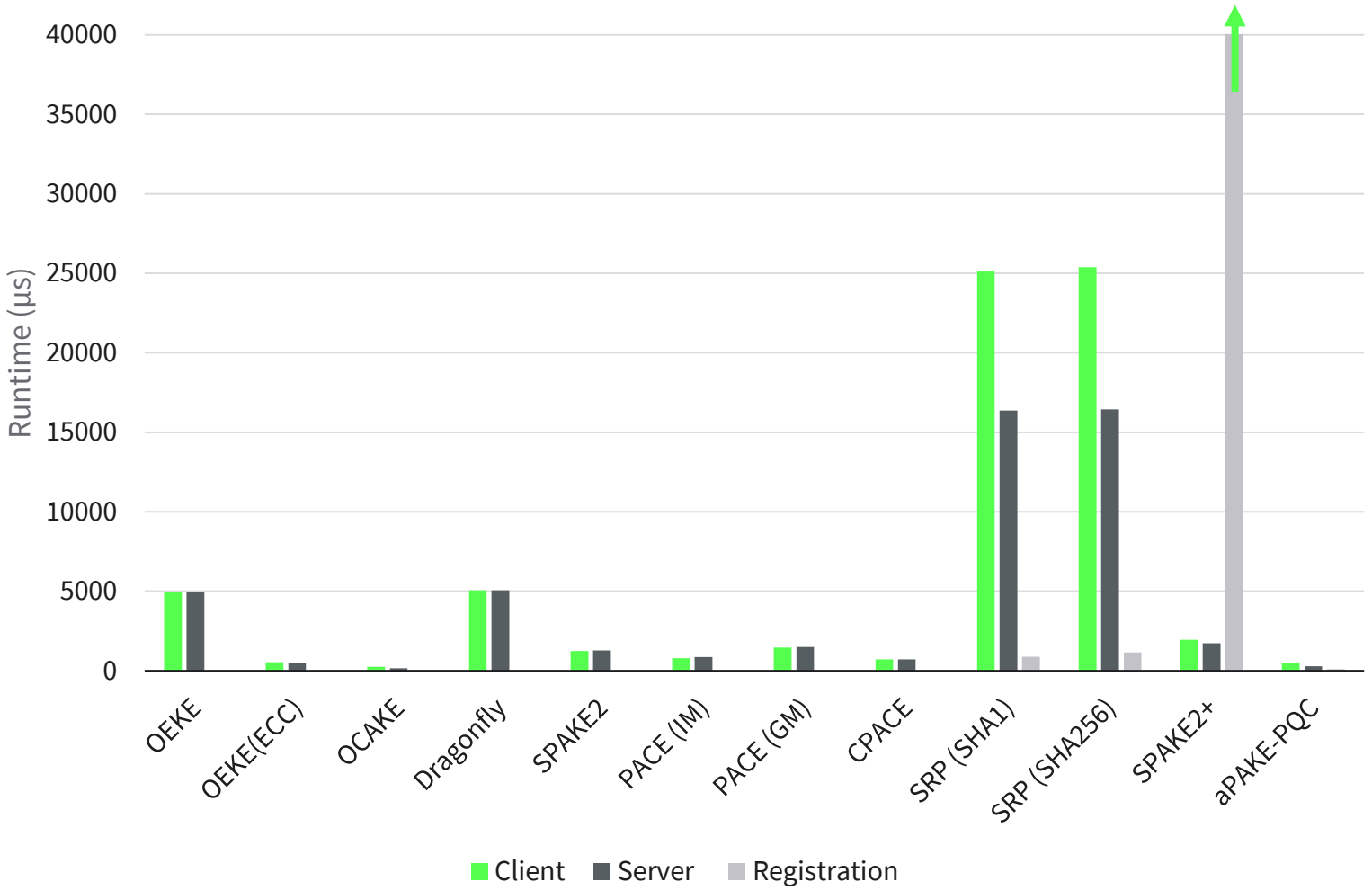
PERFORMANCE RESULTS – AUGMENTED SCHEMES



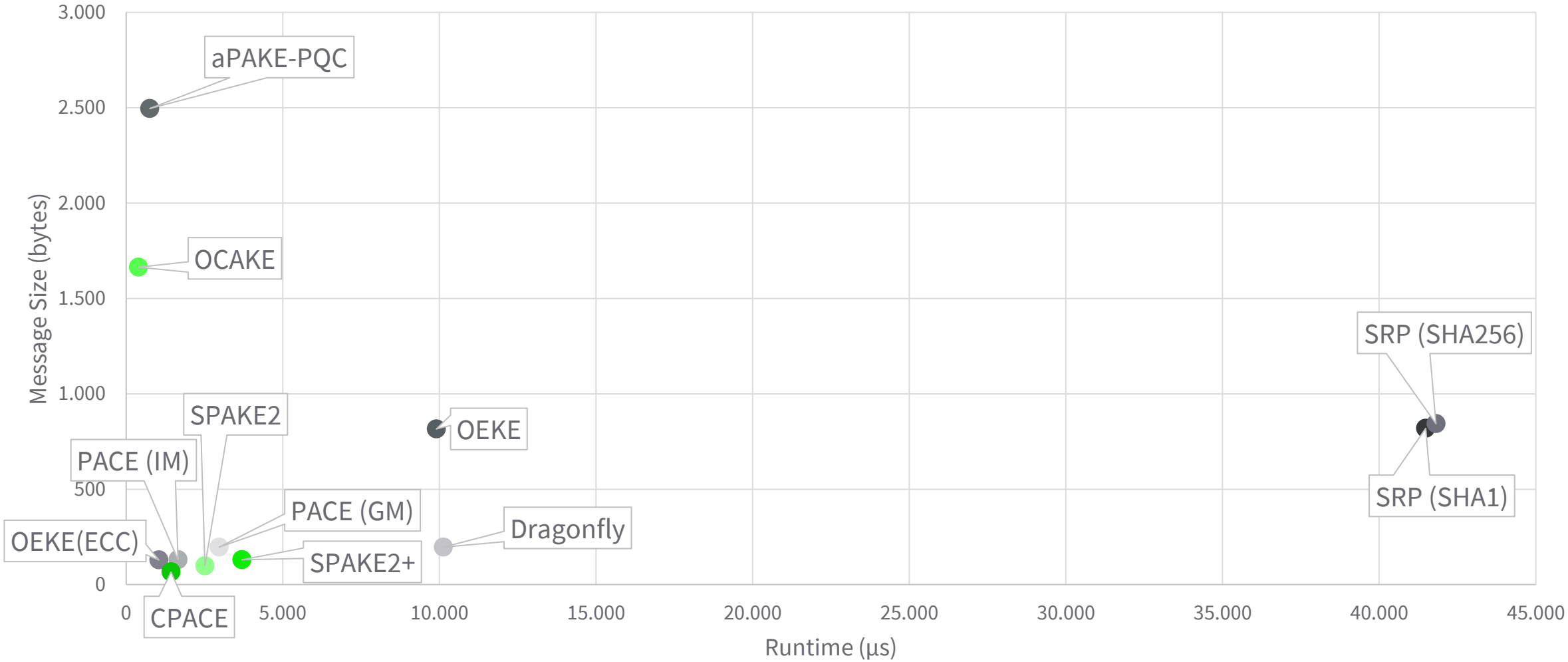
Performance impact:

- Registration of SPAKE2+ uses scrypt
  - Parameters (32768, 8, 1)
- SRP uses 3072-bit MODP Group – not secp256r1, limited impact of the used hash-function

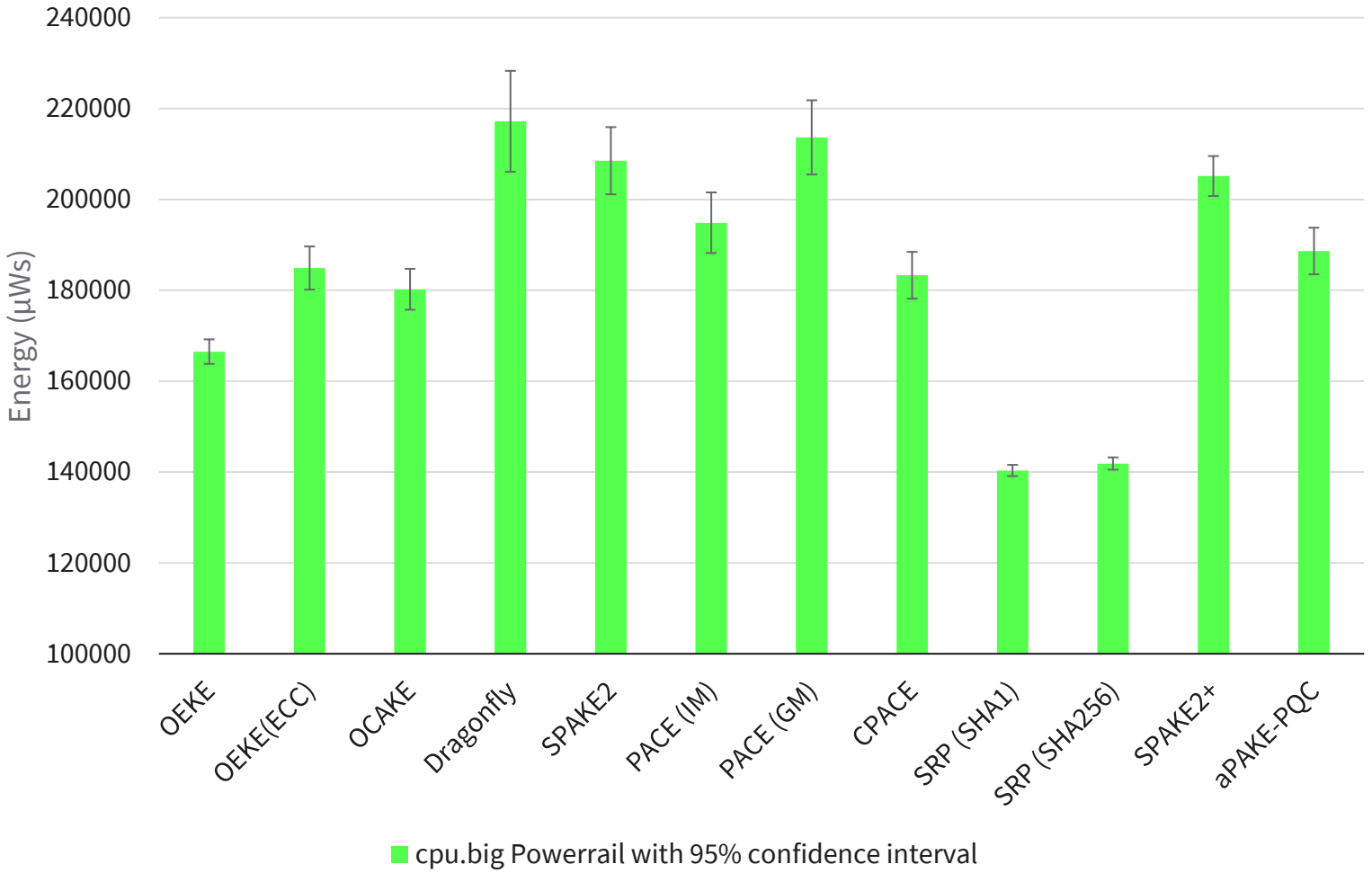
PERFORMANCE RESULTS – WHOLE PICTURE



PERFORMANCE VS MESSAGE SIZE



ENERGY CONSUMPTION



THREATS TO VALIDITY / LIMITATIONS

- Test on a single device only
- Use of Bouncy Castle – how about other libraries?
- No (explicit) use of hardware accelerations
- Network overhead and power consumption is not considered

- Energy consumption measurements on cpu.big only
  Trace alignment errors caused discards
- Outliers of performance measurements are discarded

CONCLUSIONS & FUTURE WORK

- Mapping functions, primitives, and protocol class impact runtime and energy
- Memory-hard KDFs dominate augmented registration; tuning parameters trades security vs performance
- SRP shows lower big-core energy, ECC and KEM broadly similar
- Implementing efficient balanced and augmented PQC-PAKE is possible at cost of larger messages

Future Work:

- Expand the experiments to other hardware
- (Explicitly) use Hardware-accelerated cryptographic instructions
- Analyze the impact of side-channels

THANK YOU

Jörn-Marc Schmidt and Alexander Lawall
IU International University of Applied Sciences
Erfurt, Germany

joern-marc.schmidt@iu.org
alexander.lawall@iu.org