Modified versions of ML-KEM based on Brazilian cryptographic resources

Vinícius Lagrota[1]
Beatriz L. Azevedo[2]
Mateus de L. Filomeno[2]
Moisés V. Ribeiro[2]

[1]CEPESC
[2]UFJF

17 de setembro de 2024

SBSeg24 SÃO JOSÉ DOS CAMPOS

CEPESC

Sumário

Introdução

- Algoritmos criptográficos são uma das principais armas para garantir **soberania nacional**, **segurança de informações sensíveis** e **proteção das infraestruturas críticas** contra ameaças cibernéticas e espionagem.
- Novo cenário de ameaça $\rightarrow$ computador quântico.
- Surge a necessidade de algoritmos pós-quânticos $\rightarrow$ concurso do NIST.
  - Padronização dos primeiros algoritmos este ano.
  - Apenas um Key Encapsulation Mecanism (KEM) padronizado $\rightarrow$ ML-KEM, baseado no CRYSTALS-Kyber.
    - ML-KEM-512, 768 e 1024.

SBSeg24 SÃO JOSÉ DOS CAMPOS

CEPESC

Introdução

Soberania nacional

- Nações soberanas desenvolvem sua própria criptografia.
- No Brasil, o CEPESC é um dos centros de referência para tal.
  - **Libharpia**, biblioteca utilizada nas eleições.
  - Conta com diversos algoritmos, como o ML-KEM e o Forró.
  - Forró é um algoritmo brasileiro baseado na arquitetura ARX.
  - Xote $\rightarrow$ versão acelerada do Forró utilizando duas matrizes de estado.
- A utilização do Forró e Xote fortalece a segurança nacional e soberania digital brasileira.

- Propor o ML-KEM baseado nas primitivas criptográficas simétricas Forró e Xote em substituição ao SHAKE.
- Investigar o desempenho do ML-KEM utilizando Forró e Xote e comparar com o SHAKE.

Sumário

ML-KEM instanciado com Forró e Xote

- Adaptações das primitivas simétricas $\rightarrow$ XOF, PRF e KDF.
  - XOF $\rightarrow$ gera dados que são parte da chave pública.
  - PRF $\rightarrow$ gera dados que são parte da chave privada e vetor de erros.
  - KDF $\rightarrow$ transforma o material de chaves em uma chave derivada.
- Utilização das funções disponibilizadas por Forró e Xote:
  - {Forro, Xote}.Keysetup($\cdot$).
  - {Forro, Xote}.IVsetup($\cdot$).
  - {Forro, Xote}.QR($\cdot$).
  - {Forro, Xote}.Encrypt($\cdot$).
- Criação de uma nova função:
  - {Forro,Xote}.GenerateBytes($\cdot$).

ML-KEM instanciado com Forró e Xote

Segurança

- Similar ao ML-KEM com SHAKE.
- Pequena melhora:
  - XOF no ML-KEM com SHAKE $\rightarrow$ 128-bits de segurança.
  - XOF no ML-KEM com Forró ou Xote $\rightarrow$ 256-bits de segurança.
- Os demais algoritmos introduzidos possuem o mesmo nível de segurança.

Algorithm {Forro, Xote}.XOF-absorb(st, ρ, i, j)

Input:

State matrix: $st \in \mathcal{U}^{4\times4}$
Seed: $\rho \in \mathcal{B}^{32}$
Nonce: $i, j \in \mathcal{B}^{4}$

Output:

State matrix: $st \in \mathcal{U}^{4\times4}$

Procedure:

$iv = i||j$
{Forro, Xote}.Keysetup(st, ρ)
{Forro, Xote}.IVsetup(st, iv)
{Forro, Xote}.QR(st)

Return:

State matrix: st

Algorithm {Forro, Xote}.XOF-squeeze(st, N)

Input:

State matrix: $st \in \mathcal{U}^{4\times4}$

Output length: $N \in \mathcal{U}$

Output:

Byte string: $out \in \mathcal{B}^*$

Return:

Byte string: $out :=$ {Forro, Xote}.GenerateBytes(st, N)

Algorithm {Forro, Xote}.PRF(r, i, N)

Input:

Seed: $r \in \mathcal{B}^{32}$

Nonce: $i \in \mathcal{B}$

Output length: $N \in \mathcal{U}$

Output:

Byte string: $out \in \mathcal{B}^{*}$

Procedure:

State matrix: $st \in \mathcal{U}^{4\times 4}$

IV: $iv \in \mathcal{B}^{32}$

$iv[0] = i$

{Forro, Xote}.Keysetup(st, r)

{Forro, Xote}.IVsetup(st, iv)

Return:

Byte string: $out :=$ {Forro, Xote}.GenerateBytes(st, N)

Algorithm {Forro, Xote}.KDF(kr, N)

Input:

Key material: $kr \in \mathcal{B}^{64}$

Output:

Shared secret: $ss \in \mathcal{B}^{32}$

Procedure:

State matrix: $st, st_1, st_2 \in \mathcal{U}^{4\times4}$

IV: $iv \in \mathcal{B}^{32}$

$iv := 0$

{Forro, Xote}.Keysetup(st_1, kr)

{Forro, Xote}.IVsetup(st_1, iv)

{Forro, Xote}.Keysetup($st_2, kr[32]$)

{Forro, Xote}.IVsetup($st_2, iv + 1$)

$st := st_1 \oplus st_2$

Return:

Shared secret: $ss :=$ {Forro, Xote}.GenerateBytes(st, N)

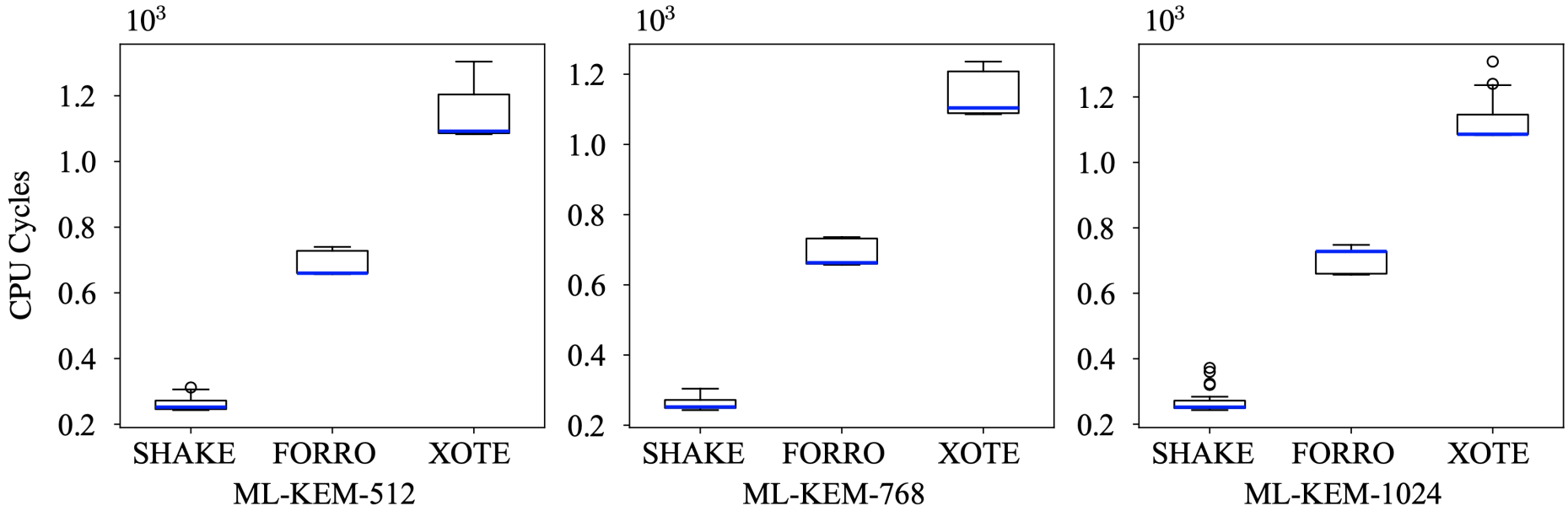
Sumário

Resultados

- Análise de desempenho do ML-KEM com SHAKE, Forró e Xote:
  - Apresentação em boxplot $\rightarrow$ 101 amostras.
  - Cada amostra é a mediana de 10001 iterações de cada função para cada nível de segurança.
- Resultados divididos entre:
  - Funções núcleos $\rightarrow$ XOF-absorb, XOF-squeeze, PRF e KDF.
  - Funções compostas $\rightarrow$ Geração do par de chaves, encapsulação e decapsulação.

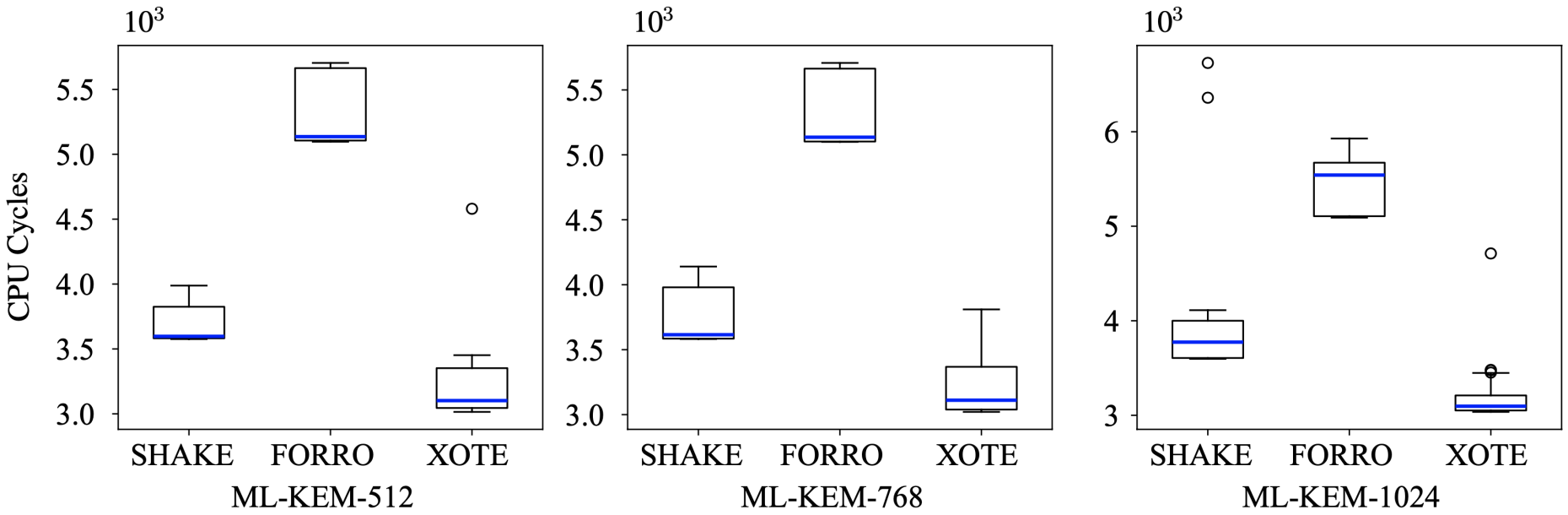
Resultados

Boxplot da função núcleo XOF-absorb



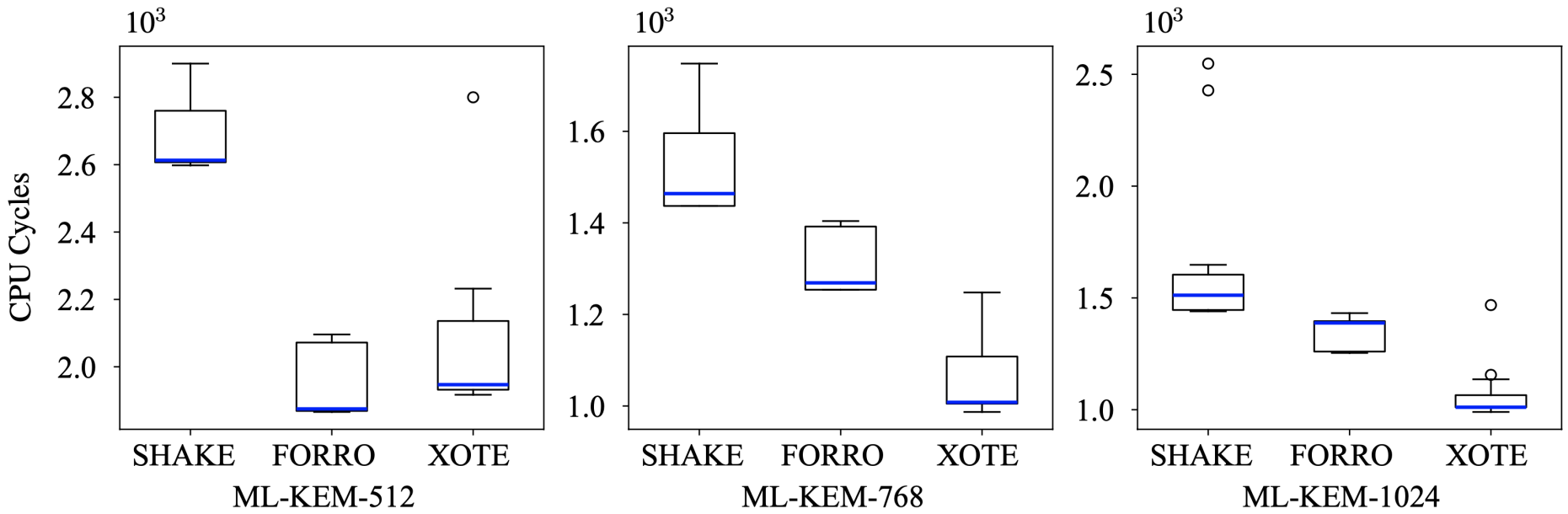
Resultados

Boxplot da função núcleo XOF-squeeze



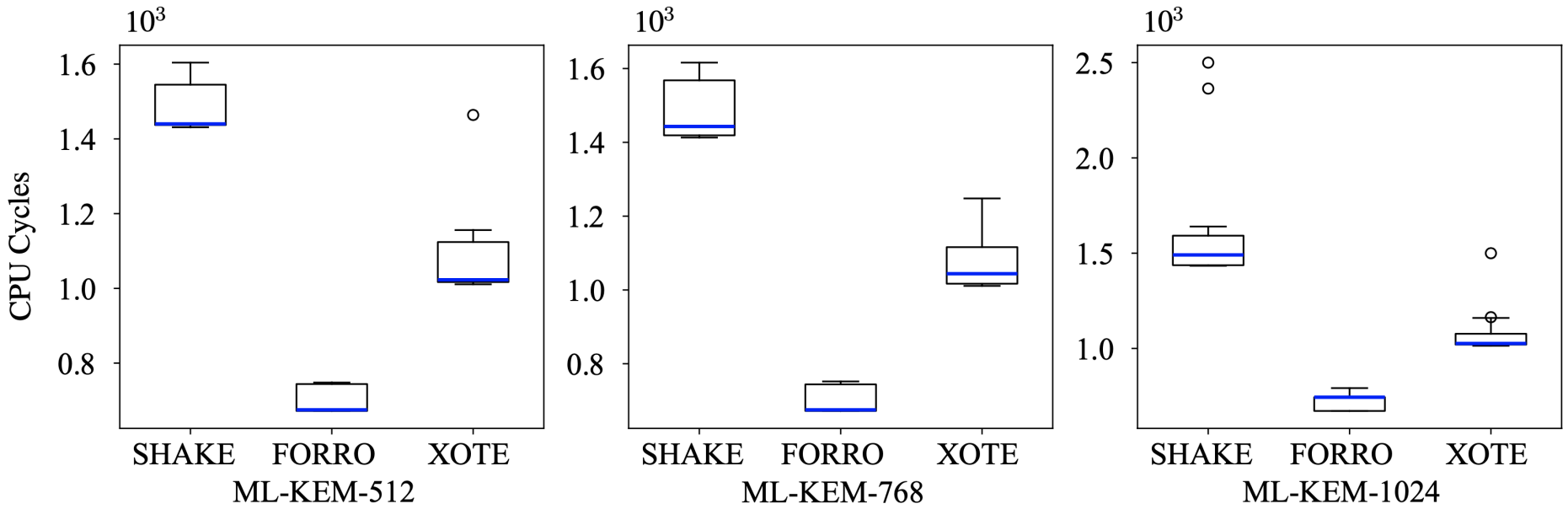
Resultados

Boxplot da função núcleo PRF



Resultados

Boxplot da função núcleo KDF



Resultados

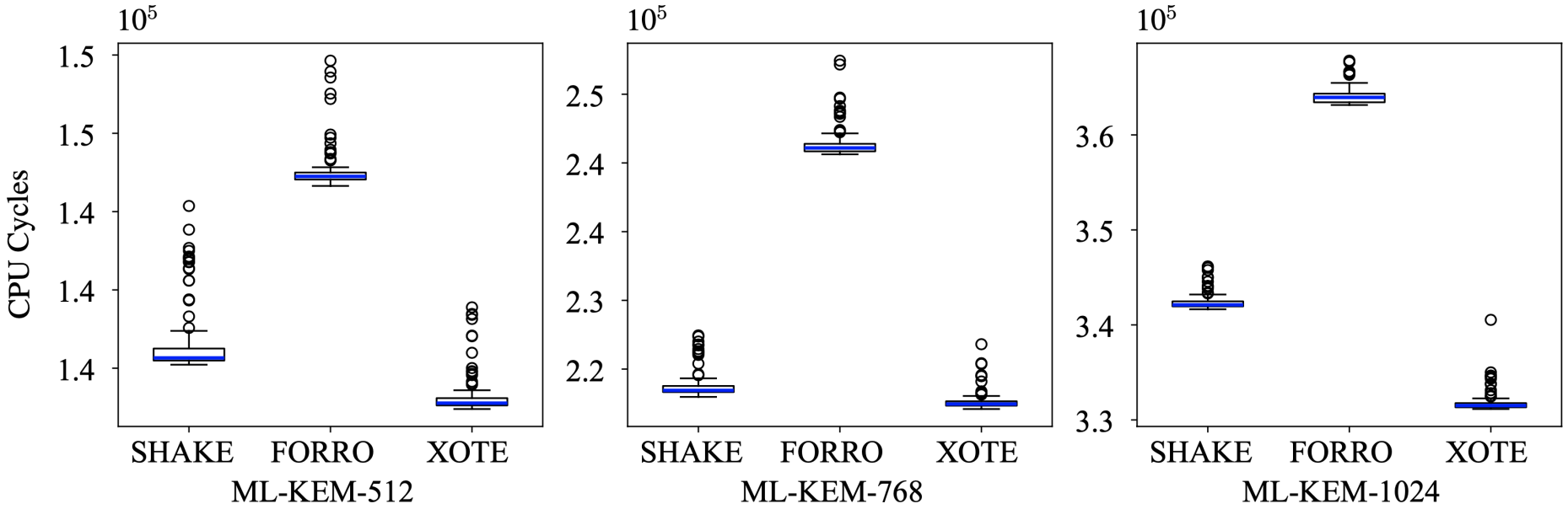
Utilização das funções núcleo pelas funções compostas

	XOF-absorb	XOF-squeeze	PRF	KDF
Geração do par de chaves	K^2	K^2	$2K$	0
Encapsulação	K^2	K^2	$2K+1$	1
Decapsulação	K^2	K^2	$2K+1$	1

- $K \in \{2, 3, 4\}$ se refere ao nível de segurança do ML-KEM-512, -768, -1024, respectivamente.

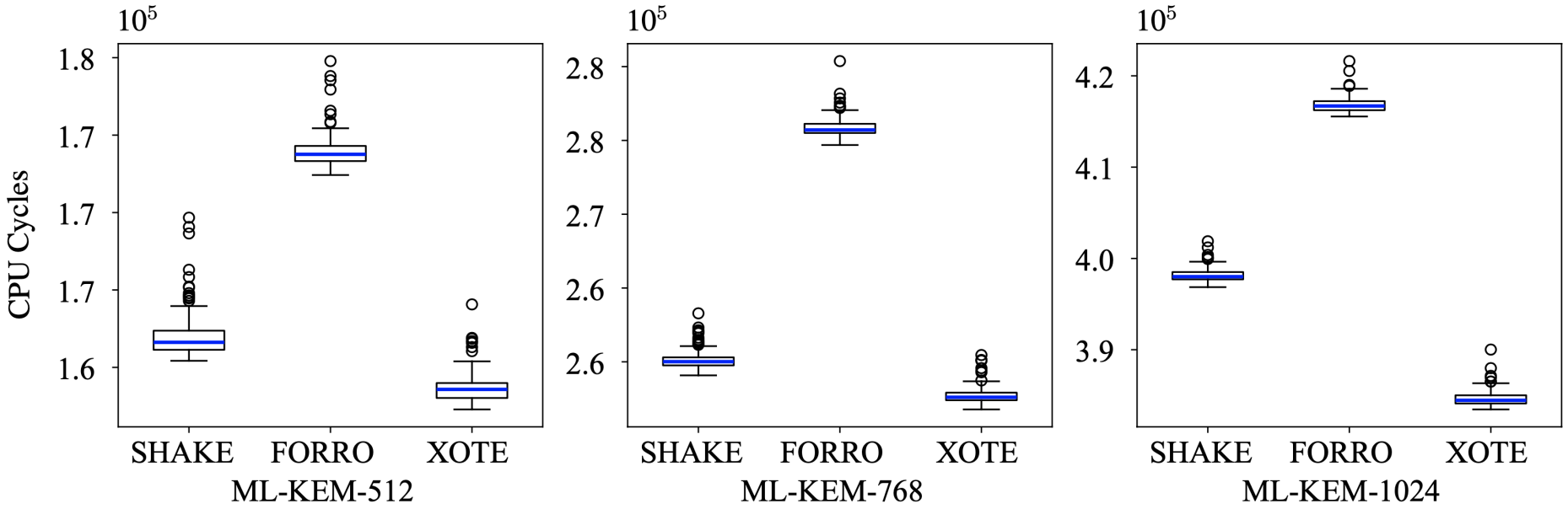
Resultados

Boxplot da função composta de geração do par de chaves



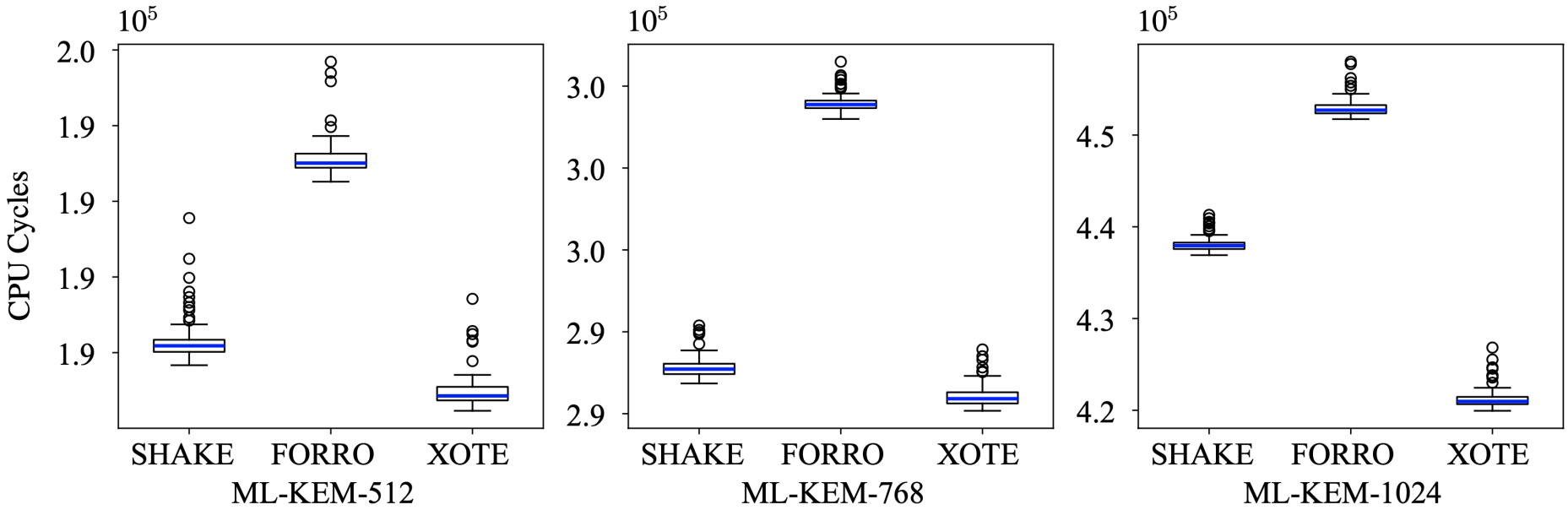
Resultados

Boxplot da função composta de encapsulação



Resultados

Boxplot da função composta de decapsulação



Resultados

Melhora de desempenho comparado ao SHAKE

	Algoritmo	Forró	Xote
ML-KEM-512	Geração do par de chaves	-4.13%	1.03%
	Encapsulação	-3.66%	0.92%
	Decapsulação	-3.22%	0.88%
ML-KEM-768	Geração do par de chaves	-7.90%	0.44%
	Encapsulação	-6.04%	0.93%
	Decapsulação	-5.61%	0.63%
ML-KEM-1024	Geração do par de chaves	-6.38%	3.10%
	Encapsulação	-4.70%	3.40%
	Decapsulação	-3.37%	3.88%

Sumário

Conclusões

- Substituição das primitivas simétricas do ML-KEM.
- Experimentos numéricos mostraram que a geração de chaves, encapsulamento e decapsulamento utilizando o ML-KEM com o Xote apresenta **ganhos em desempenho**.
  - Quando o Forró é utilizado, apresenta pior desempenho que o SHAKE e Xote.

- Trabalhos futuros:
  - Comparação entre AES acelerado em hardware com implementações AVX2 do Forró e Xote.

SBSeg24 SÃO JOSÉ DOS CAMPOS

CEPESC

Muito obrigado!

vinicius.lagrota@gmail.com
beatriz.azevedo@estudante.ufjf.br
mateus.lima@engenharia.ufjf.br
mribeiro@engenharia.ufjf.br