UNIVERSITÄT ZU LÜBECK
INSTITUTE FOR IT SECURITY

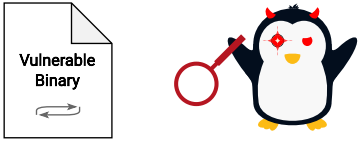
Cipherfix:
Mitigating Ciphertext Side-Channel Attacks in Software

Jan Wichelmann & **Anna Pätschke**, Luca Wilke, and Thomas Eisenbarth

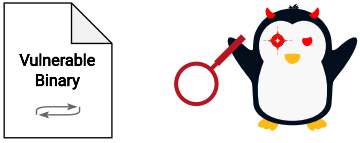
August 11, 2023

32nd Usenix Security Symposium

Roadmap

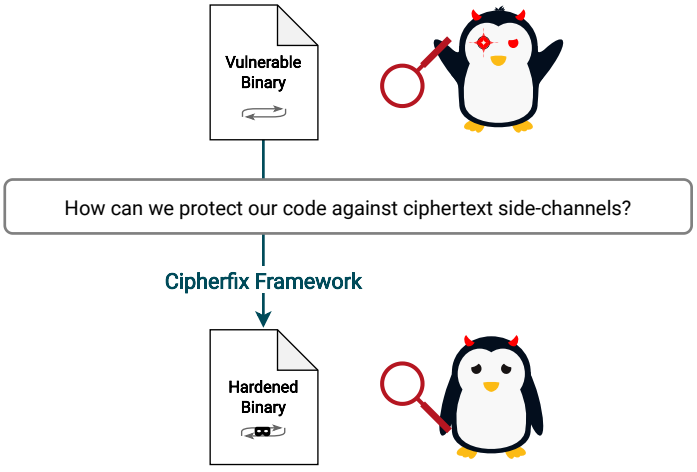


Roadmap

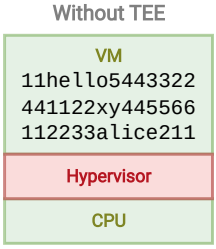


How can we protect our code against ciphertext side-channels?

Roadmap

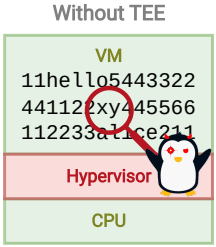

Vulnerable Binary
How can we protect our code against ciphertext side-channels?
Cipherfix Framework
Hardened Binary

Confidential Cloud Computing



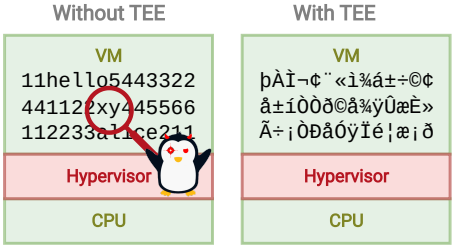
- Run code in the cloud

Confidential Cloud Computing



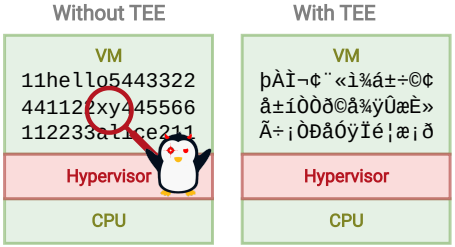
- Run code in the cloud

Confidential Cloud Computing



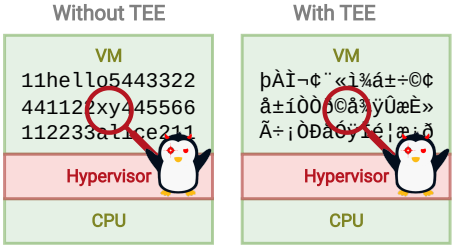
- Run code in the cloud
- Trusted Execution Environment (TEE) encrypts code and data

Confidential Cloud Computing



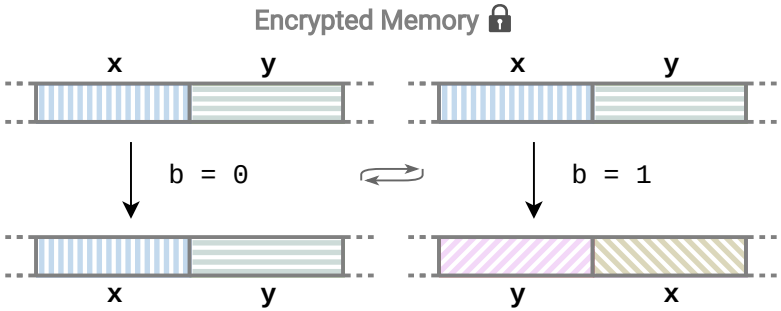
- Run code in the cloud
- Trusted Execution Environment (TEE) encrypts code and data
- Constant-time code protects against timing side-channels

Confidential Cloud Computing



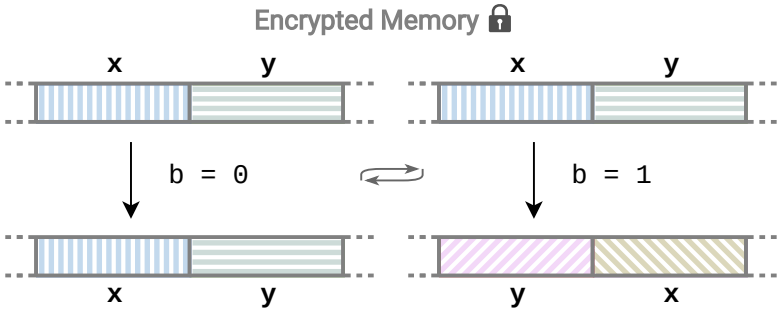
- Run code in the cloud
- Trusted Execution Environment (TEE) encrypts code and data
- Constant-time code protects against timing side-channels

Ciphertext Side-Channel



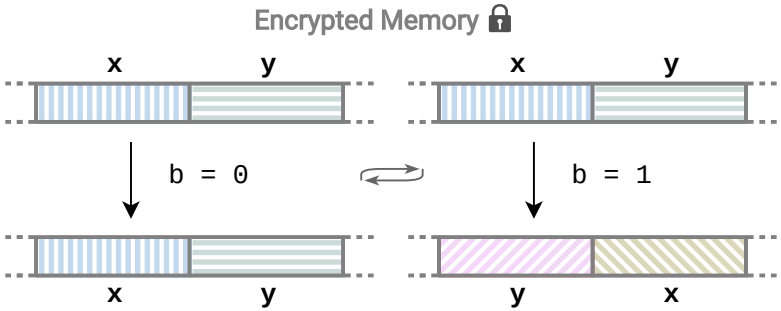
- Constant-time swap operation

Ciphertext Side-Channel



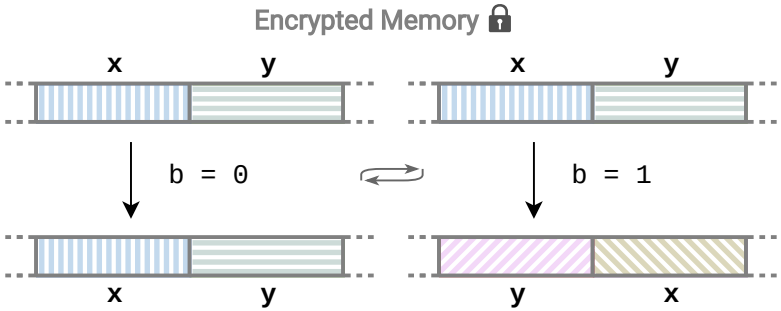
- Constant-time swap operation
- Memory encryption should protect secret b

Ciphertext Side-Channel



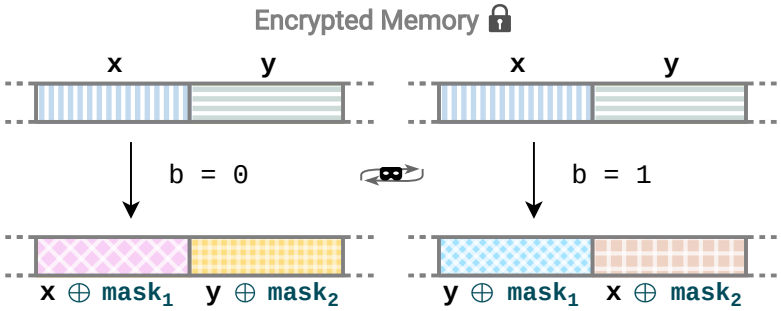
- Constant-time swap operation
- Memory encryption should protect secret b
- *Deterministic* memory encryption

Ciphertext Side-Channel



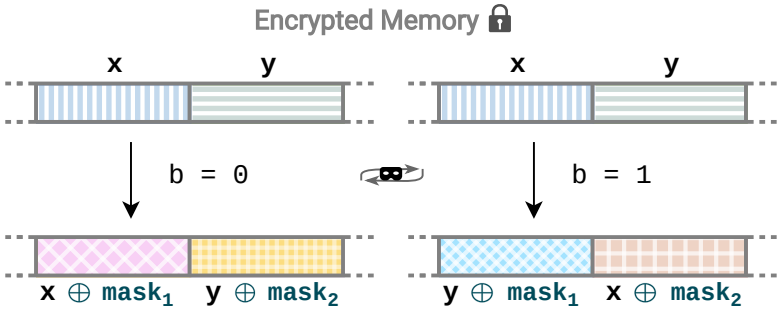
- Constant-time swap operation
- Memory encryption should protect secret b
- *Deterministic* memory encryption ⇒ ciphertext *change* leaks b

Enforcing Ciphertext Changes



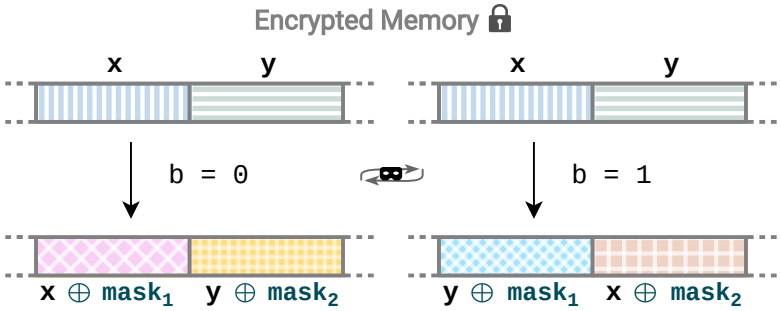
- Add random mask on every write

Enforcing Ciphertext Changes



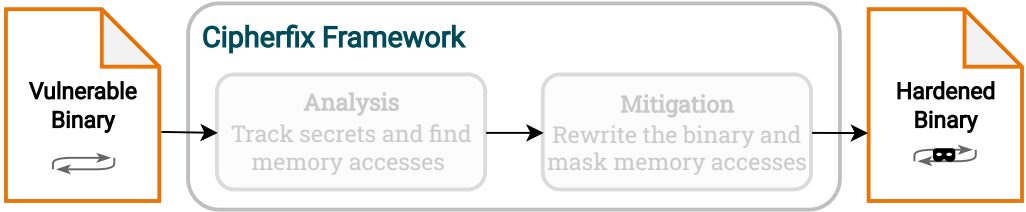
- Add random mask on every write
- Ciphertext always changes

Enforcing Ciphertext Changes

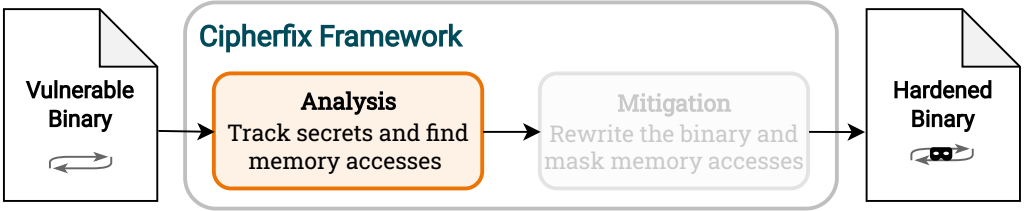


- Add random mask on every write
- Ciphertext always changes ⇒ no leakage

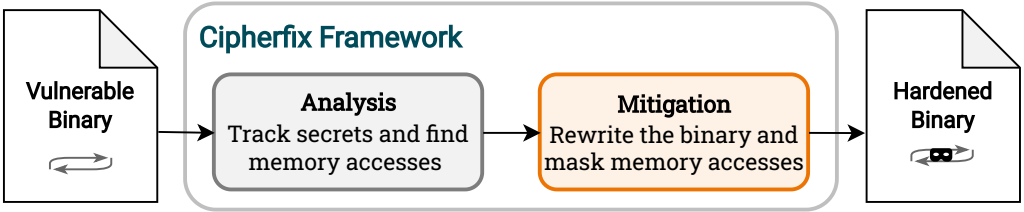
From Vulnerable to Hardened Binary



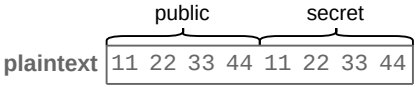
From Vulnerable to Hardened Binary



From Vulnerable to Hardened Binary

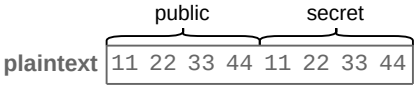


Cipherfix-Fast



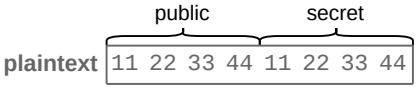
- Back data with a *mask buffer*

Cipherfix-Fast



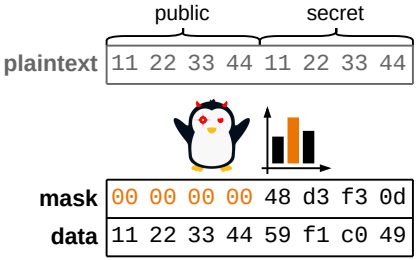
- Back data with a *mask buffer*
- Store random mask in mask buffer

Cipherfix-Fast



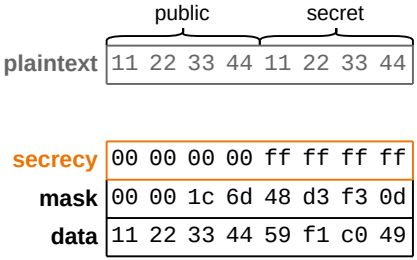
- Back data with a *mask buffer*
- Store random mask in mask buffer
- Mask is zero for public data

Cipherfix-Fast



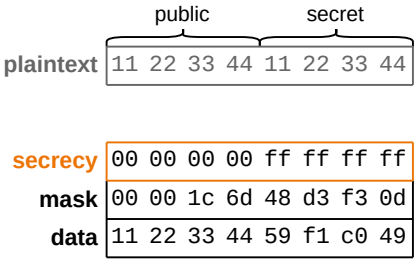
- Back data with a *mask buffer*
- Store random mask in mask buffer
- Mask is zero for public data

Cipherfix-Base



- Store secrecy information in separate *secrecy buffer*

Cipherfix-Base



- Store secrecy information in separate *secrecy buffer*
- All 1 for secret data, all 0 for public data

Mask Generation

- `rdrand`: Secure

Mask Generation

- `rdrand`: Secure and slow

Mask Generation

- `rdrand`: Secure and slow
- `XS+`: Well-known `XorShift128+` RNG

Mask Generation

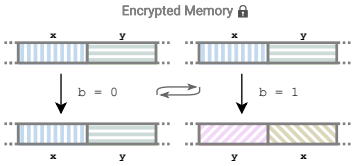
- `rdrand`: Secure and slow
- `XS+`: Well-known `XorShift128+` RNG
- AES: Custom RNG based on one AES round

Mask Generation

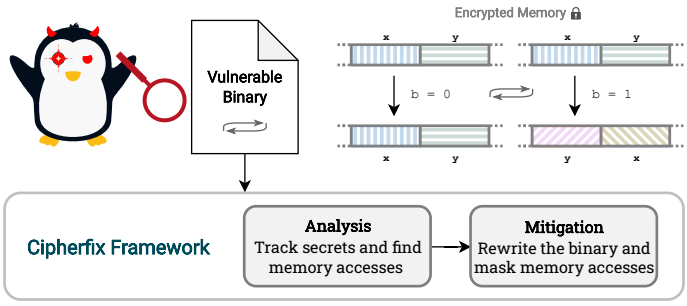
- `rdrand`: Secure and slow
- `XS+`: Well-known `XorShift128+` RNG
- `AES`: Custom RNG based on one AES round

	Performance	Security
rdrand		
XS128+		
AES		

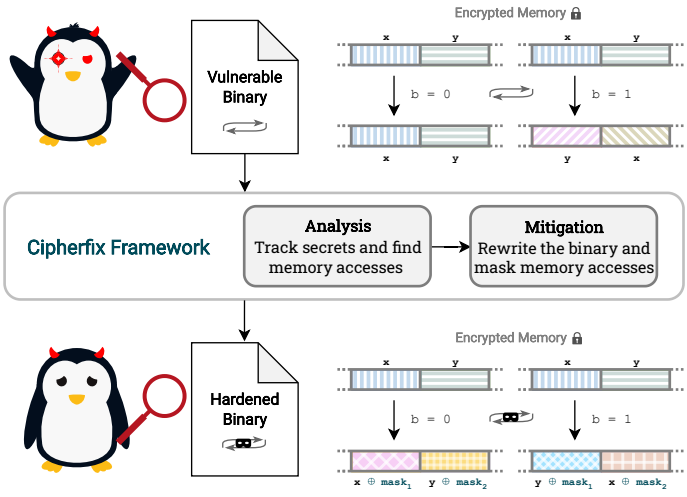
Summary



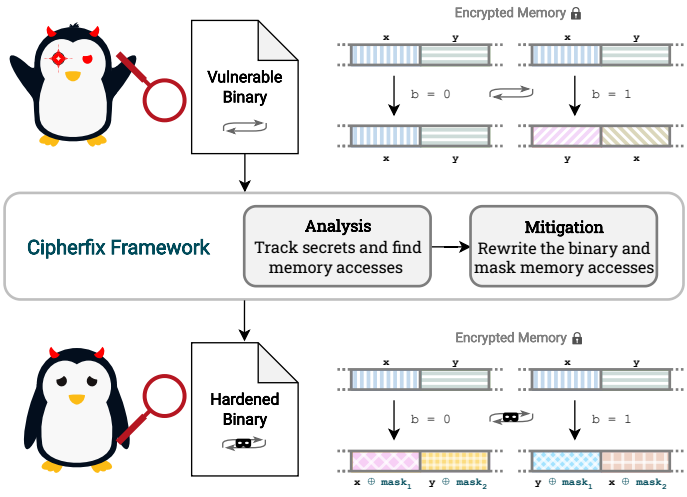
Summary


Vulnerable Binary
Encrypted Memory
x
y
x
y
b = 0
b = 1
x
y
y
x
Cipherfix Framework
Analysis
Track secrets and find memory accesses
Mitigation
Rewrite the binary and mask memory accesses

Summary


Vulnerable Binary
Encrypted Memory
x
y
b = 0
b = 1
Cipherfix Framework
Analysis
Track secrets and find memory accesses
Mitigation
Rewrite the binary and mask memory accesses
Hardened Binary
Encrypted Memory
x ⊕ mask1
y ⊕ mask2
y ⊕ mask1
x ⊕ mask2

Summary



https://github.com/UzL-ITS/cipherfix

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

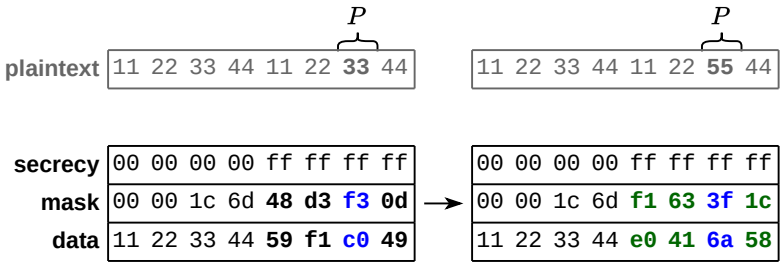
@JanWichelmann
@paetscan

Runtime Performance Overhead

	AES	XS128+	rdrand
Cipherfix-Fast	2.4x	2.7x	16.8x
Cipherfix-Base	3.9x	4.0x	17.3x
Cipherfix-Enhanced	5.1x	5.3x	17.5x

- Average performance overhead factor of multiple primitives
- Slowdown is restricted to few isolated code sections

Cipherfix-Enhanced



- Store secrecy information in separate *secrecy buffer*
- All 1 for secret data, all 0 for public data
- Enforce minimum size of memory writes