Cipherfix:
Mitigating Ciphertext Side-Channel Attacks in Software

Jan Wichelmann & Anna Pätschke, Luca Wilke, and Thomas Eisenbarth
August 11, 2023

32nd Usenix Security Symposium
Vulnerable Binary

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

Cipherfix Framework

Vulnerable Binary

Hardened Binary

Roadmap
• Run code in the cloud
Confidential Cloud Computing

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

Cipherfix: Mitigating Ciphertext Side-Channel Attacks in Software
- Constant-time swap operation
- Memory encryption should protect secret $b$
• Constant-time swap operation
• Memory encryption should protect secret $b$
• *Deterministic* memory encryption
• Constant-time swap operation
• Memory encryption should protect secret $b$
• *Deterministic* memory encryption $\Rightarrow$ ciphertext change leaks $b$
Enforcing Ciphertext Changes

- Add random mask on every write

encrypted Memory

\[ x \oplus \text{mask}_1 \quad y \oplus \text{mask}_2 \]
\[ y \oplus \text{mask}_1 \quad x \oplus \text{mask}_2 \]
Enforcing Ciphertext Changes

- Add random mask on every write
- Ciphertext always changes
Enforcing Ciphertext Changes

- Add random mask on every write
- Ciphertext always changes $\Rightarrow$ no leakage
From Vulnerable to Hardened Binary

Cipherfix Framework

- **Analysis**: Track secrets and find memory accesses
- **Mitigation**: Rewrite the binary and mask memory accesses

Cipherfix: Mitigating Ciphertext Side-Channel Attacks in Software
From Vulnerable to Hardened Binary

Cipherfix Framework

Vulnerable Binary

Analysis
Track secrets and find memory accesses

Mitigation
Rewrite the binary and mask memory accesses

Hardened Binary
From Vulnerable to Hardened Binary

Cipherfix Framework

Vulnerable Binary → **Analysis** → **Mitigation** → Hardened Binary

- **Analysis**: Track secrets and find memory accesses
- **Mitigation**: Rewrite the binary and mask memory accesses
• Back data with a *mask buffer*
• Back data with a *mask buffer*
• Store random mask in mask buffer
- Back data with a *mask buffer*
- Store random mask in mask buffer
- Mask is zero for public data
• Back data with a *mask buffer*
• Store random mask in mask buffer
• Mask is zero for public data
• Store secrecy information in separate secrecy buffer
• 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

<table>
<thead>
<tr>
<th>Performance</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdrand</td>
<td><img src="image" alt="Performance" /> <img src="image" alt="Security" /></td>
</tr>
<tr>
<td>XS128+</td>
<td><img src="image" alt="Performance" /> <img src="image" alt="Security" /></td>
</tr>
<tr>
<td>AES</td>
<td><img src="image" alt="Performance" /> <img src="image" alt="Security" /></td>
</tr>
</tbody>
</table>
Cipherfix: Mitigating Ciphertext Side-Channel Attacks in Software

Summary

Encrypted Memory

Analysis
Track secrets and find memory accesses

Mitigation
Rewrite the binary and mask memory accesses

Cipherx Framework

Vulnerable Binary

b = 0

b = 1

https://github.com/UzL-ITS/cipherfix
@JanWichelmann
@paetscan
Cipherfix: Mitigating Ciphertext Side-Channel Attacks in Software
Cipherfix: Mitigating Ciphertext Side-Channel Attacks in Software
Summary

Cipherfix Framework

Vulnerable Binary

- Analysis
  Track secrets and find memory accesses

- Mitigation
  Rewrite the binary and mask memory accesses

Hardened Binary

Cipherfix: Mitigating Ciphertext Side-Channel Attacks in Software

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

@JanWichelmann
@paetscan
Runtime Performance Overhead

<table>
<thead>
<tr>
<th>Cipher</th>
<th>AES</th>
<th>XS128+</th>
<th>rdrand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cipherfix-Fast</td>
<td>2.4x</td>
<td>2.7x</td>
<td>16.8x</td>
</tr>
<tr>
<td>Cipherfix-Base</td>
<td>3.9x</td>
<td>4.0x</td>
<td>17.3x</td>
</tr>
<tr>
<td>Cipherfix-Enhanced</td>
<td>5.1x</td>
<td>5.3x</td>
<td>17.5x</td>
</tr>
</tbody>
</table>

- Average performance overhead factor of multiple primitives
- Slowdown is restricted to few isolated code sections
- Store secrecy information in separate secrecy buffer
- All 1 for secret data, all 0 for public data
- Enforce minimum size of memory writes