Automated Detection of Single-Trace Side-Channel Vulnerabilities in Constant-Time Cryptographic Code

Ferhat Erata†, Ruzica Piskac†, Victor Mateu‡, and Jakub Szefer†

†Yale University · ‡Technology Innovation Institute

July 3, 2023

Post Quantum Cryptography: Kyber's message encoding

```c
void poly_frommsg(poly *r,
      const uint8_t msg[KYBER_INDCPA_MSGBYTES]) {
  unsigned int i, j;
  int16_t mask;
  for (i = 0; i < KYBER_N / 8; i++) {
   for (j = 0; j < 8; j++) {
     mask = -(int16_t)((msg[i] >> j) & 1);
     r->coeffs[8*i + j] = mask & ((KYBER_Q+1)/2);
    }
  }
}
```

Listing: CRYSTALS-Kyber's message encoding, attacked by [Steffen et al., 2021, Ravi et al., 2020]

- The number of cases of the mask value is 2: -1 (0xFFFF) and 0 (0x0000).
- The complete shared secret can be extracted from one single trace only.

Kyber's message encoding – Instruction-level

```
...
ldr    r2, [r7,#0]
ldr    r3, [r7,#20]
add    r3, r2
ldrb   r3, [r3,#0]
mov    r2, r3
ldr    r3, [r7,#16]
asr.w  r3, r2        ⋆
and.w  r3, r3, #1    ⋆
negs   r3, r3        ⋆
...
```

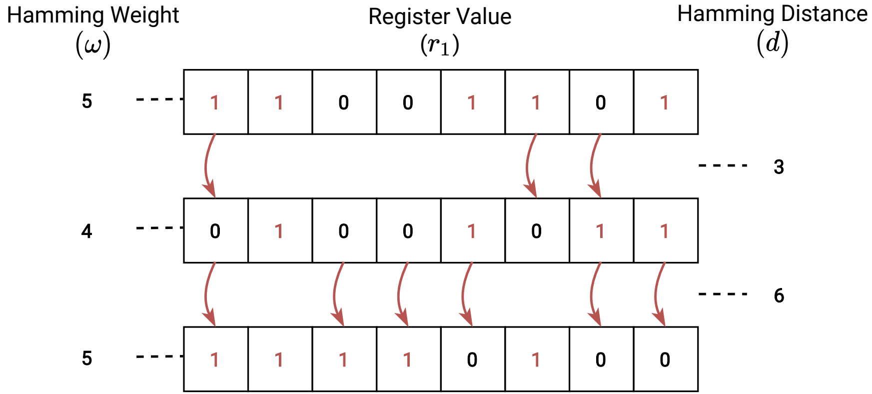
Listing: Partial disassembly at −O0

```
...
...
ldrb    r2, [r3,#0]
sbfx    r2, r2, #0, #1   ⋆
strh    r2, r2, #6144
and.w   r2, r2, [r0,#0]
strh.w  r2, [r0,#512]
strh.w  r2, [r0,#1024]
strh.w  r2, [r0,#1536]
...
...
```

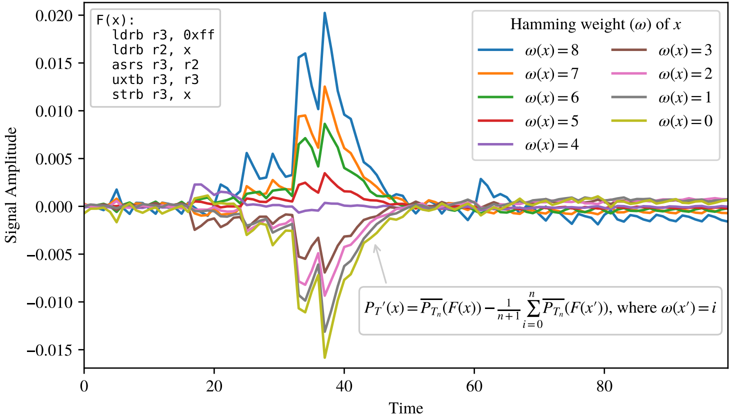
Listing: Partial disassembly at −O3

- The shared secret can be read from an oscilloscope display directly with the naked eye with optimization turned off (-O0);
- When optimizations are enabled (-O3), the attack requires template-based attack, but the attack still works on single power traces.

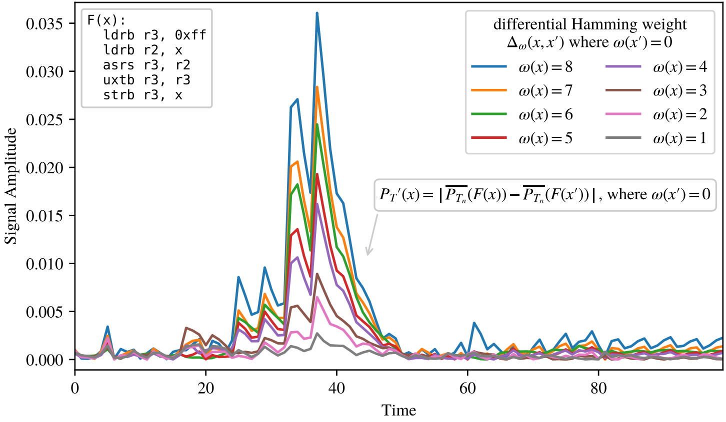
Leakage Models: Data Dependent Power Consumption



Hamming Weight - ASR inst. - ARM Cortex-M4F3



Differential Hamming Weight - ASR inst. - ARM Cortex-M4F3



- **Input:** A binary executable or Region of Interest & Marking Secrets;
- **Output:** A set of leakage locations (PoIs) and their corresponding leakage values.
- **Output:** Test Vector Generation.

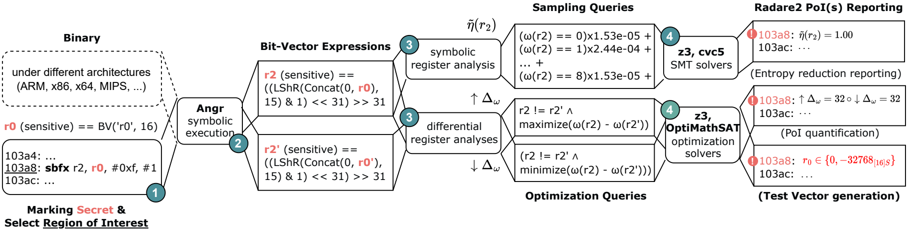


Figure: Pascal: Power Analysis Side Channel Attack Locator – Tooling Workflow

Differential Register Analysis

- Example of conditional addition written in a constant-time style using masking.

$$
\begin{aligned}
&f(sum_{[8]S},\ x_{[8]S}):\\
&\quad mask := (x - 64) \gg 7\\
&\quad sum := sum + (\sim mask \wedge x)
\end{aligned}
$$

- Dissassembly of the binary code & forward taint analysis

state	lifted binary code	symbolic store σ
S_0	$f(sum_{0[8]}, x_{0[8]})$	$\sigma_0 := \{x_0 = \beta_{[8]} \wedge sum_0 = \lambda_{[8]}\}$
S_1	$r_0 := x_0 - 64$	$\sigma_1 := \{\sigma_0 \wedge r_0 = x_0 - 64\}$
S_2	$r_1 := r_0 \gg 7$	$\sigma_2 := \{\sigma_1 \wedge r_1 = f_{asr}(r_0, 7)\}$
S_3	$r_2 := \sim r_1$	$\sigma_3 := \{\sigma_2 \wedge r_2 = \sim r_1\}$
S_4	$r_3 := r_2 \wedge x_0$	$\sigma_4 := \{\sigma_3 \wedge r_3 = r_2 \wedge x_0\}$
S_5	$r_4 := sum_0 + r_3$	$\sigma_5 := \{\sigma_4 \wedge r_4 = sum_0 \wedge r_3\}$

Differential Register Analysis

- Dissassembly of the binary code & forward taint analysis

state	lifted binary code	symbolic store σ
S_0	$f(sum_{0[8]}, x_{0[8]})$	$\sigma_0 := \{x_0 = \beta_{[8]} \wedge sum_0 = \lambda_{[8]}\}$
S_1	$r_0 := x_0 - 64$	$\sigma_1 := \{\sigma_0 \wedge r_0 = x_0 - 64\}$
S_2	$r_1 := r_0 \gg 7$	$\sigma_2 := \{\sigma_1 \wedge r_1 = f_{asr}(r_0, 7)\}$
S_3	$r_2 := \sim r_1$	$\sigma_3 := \{\sigma_2 \wedge r_2 = \sim r_1\}$
S_4	$r_3 := r_2 \wedge x_0$	$\sigma_4 := \{\sigma_3 \wedge r_3 = r_2 \wedge x_0\}$
S_5	$r_4 := sum_0 + r_3$	$\sigma_5 := \{\sigma_4 \wedge r_4 = sum_0 \wedge r_3\}$

- Relational Symbolic Execution over Fixed-Size Bit-Vectors

$$\varphi_{SC_{S_2}} \triangleq \overbrace{r_1 \neq r_1'}^{\text{disjoint 2-secrets}} \wedge \underbrace{r_1 = (\beta - 64) \gg 7}_{\text{symbolic register } r_1} \wedge \overbrace{r_1' = (\beta' - 64) \gg 7}^{\text{self-composition of } r_1}$$

Differential Register Analysis

- Relational Symbolic Execution over Fixed-Size Bit-Vectors

$$\varphi_{SC_{S_2}} \triangleq \overbrace{r_1 \neq r_1'}^{\text{disjoint 2-secrets}} \wedge \underbrace{r_1 = (\beta - 64) \gg 7}_{\text{symbolic register } r_1} \wedge \overbrace{r_1' = (\beta' - 64) \gg 7}^{\text{self-composition of } r_1}$$

- Single-Objective Optimization Queries

state	Δ_ω objective function	optimization query
S_2	maximize $\Delta_\omega(r_1, r_1')$	$\varphi_{SC_{S_2}} \wedge max(\lvert\omega(r_1) - \omega(r_1{}'))\rvert)$
	minimize $\Delta_\omega(r_1, r_1')$	$\varphi_{SC_{S_2}} \wedge min(\lvert\omega(r_1) - \omega(r_1{}'))\rvert)$

- Reporting of Points of Interest & Quantification of the Differential Behavior

state	lifted binary code	$\Delta_\omega(weight)$	$d(distance)$
S_1	$r_0 := x_0 - 64$	$\Delta_{\uparrow\downarrow\Delta_\omega} = 8$	$\Delta_{\uparrow\downarrow\, d} = 7$
S_2	$r_1 := r_0 \gg 7$ ⋆	$\Delta_{\uparrow\downarrow\Delta_\omega} = 0$	$\Delta_{\uparrow\downarrow\, d} = 0$
S_3	$r_2 := \sim r_1$ ⋆	$\Delta_{\uparrow\downarrow\Delta_\omega} = 0$	$\Delta_{\uparrow\downarrow\, d} = 0$
S_4	$r_3 := r_2 \wedge x_0$	$\Delta_{\uparrow\downarrow\Delta_\omega} = 8$	$\Delta_{\uparrow\downarrow\, d} = 7$
S_5	$r_4 := sum_0 + r_3$	$\Delta_{\uparrow\downarrow\Delta_\omega} = 8$	$\Delta_{\uparrow\downarrow\, d} = 7$

Symbolic Register Analysis

ω classes and probabilities for $\mathbb{F}_8$

ω_i	0	1	2	3	4	5	6	7	8
$\lvert\omega_i\rvert$	1	8	28	56	70	56	28	8	1
$\mathbb{P}_{\omega_i}$	.004	.031	.109	.219	.273	.219	.109	.031	.004

- An approximate model to obtain the entropy of destination registers to quantify the leakage over the single-trace.

ω-class sampling model

$$\tilde{\eta}(r) = -\sum_{i=0}^{n} \mathbb{P}_{\omega_i}(r) \cdot \log_2 \mathbb{P}_{\omega_i}(r), \text{ where } r \in \mathbb{F}_n$$

- An approximate model to obtain the entropy of destination registers to quantify the leakage over the single-trace.

ω-class sampling model

$$\tilde{\eta}(r) = -\sum_{i=0}^{n} \mathbb{P}_{\omega_i}(r) \cdot \log_2 \mathbb{P}_{\omega_i}(r), \text{ where } r \in \mathbb{F}_n$$

state	machine code		$\tilde{\eta}(entropy)$	$\Delta_\omega(weight)$	$d(distance)$
S_1	$r_0 := x_0 - 64$		$\tilde{\eta}(r_0) = 2.54$	$\Delta_{\uparrow\downarrow\Delta_\omega} = 8$	$\Delta_{\uparrow\downarrow\, d} = 7$
S_2	$r_1 := r_0 \gg 7$	$\star$	$\tilde{\eta}(r_1) = 1.00$	$\Delta_{\uparrow\downarrow\Delta_\omega} = 0$	$\Delta_{\uparrow\downarrow\, d} = 0$
S_3	$r_2 := \sim r_1$	$\star$	$\tilde{\eta}(r_2) = 1.00$	$\Delta_{\uparrow\downarrow\Delta_\omega} = 0$	$\Delta_{\uparrow\downarrow\, d} = 0$
S_4	$r_3 := r_2 \wedge x_0$		$\tilde{\eta}(r_3) = 2.54$	$\Delta_{\uparrow\downarrow\Delta_\omega} = 8$	$\Delta_{\uparrow\downarrow\, d} = 7$
S_5	$r_4 := sum_0 + r_3$		$\tilde{\eta}(r_4) = 2.54$	$\Delta_{\uparrow\downarrow\Delta_\omega} = 8$	$\Delta_{\uparrow\downarrow\, d} = 7$

Post Quantum Cryptography: Kyber's message encoding

```
void poly_frommsg(poly *r,
      const uint8_t msg[KYBER_INDCPA_MSGBYTES]) {
  unsigned int i, j;
  int16_t mask;
  for (i = 0; i < KYBER_N / 8; i++) {
   for (j = 0; j < 8; j++) {
     mask = -(int16_t)((msg[i] >> j) & 1);
     /* $\uparrow \Delta_\omega = 16 \circ \downarrow \Delta_\omega = 16 \parallel \uparrow \Delta_d = 16 \circ \downarrow \Delta_d = 16 \parallel \tilde{\eta} = 1.00$ */
     r->coeffs[8*i + j] = mask & ((KYBER_Q+1)/2);
    }
  }
}
```

Listing: CRYSTALS-Kyber's message encoding, attacked by [Steffen et al., 2021, Ravi et al., 2020]

- The mask value can be either 0x0000 or 0xFFFF; therefore, the number of cases of the mask value is 2: -1 (0xFFFF) with $\omega = 16$ and 0 (0x0000) with $\omega = 0$.

Post Quantum Cryptography: Dilithium poly. generation

```
void poly_challenge(poly *c,
            const uint8_t seed[SEEDBYTES]) {
  ...
  for (i = 0; i < 8; ++i) signs |= (uint64_t)buf[i] << 8 * i;
  pos = 8;
  for (i = 0; i < N; ++i) c->coeffs[i] = 0;
  for (i = N - TAU; i < N; ++i) {
    do {
      if (pos >= SHAKE256_RATE) {
        shake256_squeezeblocks(buf, 1, &state); pos = 0;}
      b = buf[pos++];
    } while (b > i);
    c->coeffs[i] = c->coeffs[b];
    c->coeffs[b] = 1 - 2 * (signs & 1);   /* ↑ Δω = 31 ∘ ↓ Δω = 31 */
    signs >>= 1;
  }
}
```

Listing: CRYSTALS-Dilithium polynomial generation, attacked by [Karabulut et al., 2022]

- How many negative and positive coefficients the private polynomial has can leak: -1 (0xFFF...F) with $\omega = 32$ and 1 (0x000...1) with $\omega = 1$

Lightweight Cryptography: SPECK's ARX-box

```c
// Rotate left for 16 bit registers.
#define ROTL(x, n) (((x) << n) | ((x) >> (16-(n))))
// Rotation and Addition.
void A(uint16_t* l, uint16_t* r) {
  (*l) = ROTL((*l), 9);
  (*l) += (*r);
  (*r) = ROTL((*r), 2);
  (*r) ^= (*l);
}
```

Listing: SPECK's ARX-box Implementation attacked by [Yan and Oswald, 2019]

- Maximum $\Delta_\omega = 2$ and minimum $\Delta_\omega = 0$.

Vulnerability Detection: SPECK's ARX-box

```
1  ldrh r2, [r0]              ; arg1;
2  lsrs r3, r2, 7             ; $\uparrow \Delta_\omega = 9 \circ \downarrow \Delta_\omega = 0$
3  orr.w r3, r3, r2, lsl 9    ; $\uparrow \Delta_\omega = 25 \circ \downarrow \Delta_\omega = 0$
4  uxth r3, r3                ; $\uparrow \Delta_\omega = 16 \circ \downarrow \Delta_\omega = 0$
5  strh r3, [r0]              ; arg1
6  ldrh r2, [r1]              ; arg2
7  add r3, r2
8  strh r3, [r0]              ; arg1
9  ldrh r2, [r1]              ; arg2
10 lsrs r3, r2, 0xe           ; $\uparrow \Delta_\omega = 2 \circ \downarrow \Delta_\omega = 0$
11 orr.w r3, r3, r2, lsl 2    ; $\uparrow \Delta_\omega = 18 \circ \downarrow \Delta_\omega = 0$
12 uxth r3, r3                ; $\uparrow \Delta_\omega = 18 \circ \downarrow \Delta_\omega = 0$
13 strh r3, [r1]              ; arg2
14 ldrh r2, [r0]              ; arg1
15 eors r3, r2                ; $\uparrow \Delta_\omega = 18 \circ \downarrow \Delta_\omega = 0$
16 strh r3, [r1]              ; arg2
17 bx lr
```

Listing: Full disassembly of ARX-box, attacked by [Yan and Oswald, 2019]

- Over the set of 2^{16} numbers, the register `r3` at line 10 can take only one of $\{0, 1, 2, 3\}$ and therefore this reduces the number of traces required for the attack.

contributions

1. Developed Pascal, an advanced analysis tool, for the detection and quantification of potential single-trace power side-channel vulnerabilities at the binary level.
2. Introduced two novel register analysis techniques that utilize relational symbolic execution for power side-channel analysis.
3. Conducted a comprehensive systematic analysis of power side-channel attacks documented in the literature to validate the effectiveness of Pascal.
4. Identified 30 distinct vulnerabilities present in a wide range of cryptographic schemes.

https://arxiv.org/abs/2304.02102

Technology Innovation Institute

References

[Karabulut et al., 2022] Karabulut, E., Alkim, E., and Aysu, A. (2022).
Single-trace side-channel attacks on ω-small polynomial sampling.
Cryptology ePrint Archive, Report 2022/494.
https://ia.cr/2022/494.

[Ravi et al., 2020] Ravi, P., Bhasin, S., Roy, S. S., and Chattopadhyay, A. (2020).
Drop by drop you break the rock - exploiting generic vulnerabilities in lattice-based pke/kems using em-based physical attacks.
Cryptology ePrint Archive, Report 2020/549.
https://ia.cr/2020/549.

[Steffen et al., 2021] Steffen, H. M., Kogelheide, L. J., and Bartkewitz, T. (2021).
In-depth analysis of side-channel countermeasures for crystals-kyber message encoding on arm cortex-m4.
Cryptology ePrint Archive, Report 2021/1307.
https://ia.cr/2021/1307.

[Yan and Oswald, 2019] Yan, Y. and Oswald, E. (2019).
Examining the practical side channel resilience of arx-boxes.
Cryptology ePrint Archive, Report 2019/335.
https://ia.cr/2019/335.