MI-PB-2

Diferenciální odběrová analýza, možnosti obrany. Vliv operací, dat a šumu implementovaného šifrovacího algoritmu na naměřený signál spotřeby.

Vlastnosti signálu

Spotřeba logických hradel **závisí na aktivitě** obvodu -- **intenzitě výpočtu** a **vnitřních hodnotách**.

Komponenty bodu P v průběhu spotřeby:

- P_{op}: operačně závislá (typicky využito SPA)
- P_{data}: datově závislá (typicky využito DPA)
- $P_{\text{el. noise}}$: elektronický šum
- P_{const}: konstantní komponenta

$$P_{\text{total}} = P_{\text{op}} + P_{\text{data}} + P_{\text{el. noise}} + P_{\text{const}}$$

Pouze **část** $P_{\text{op}} + P_{\text{data}}$ je **využitelná pro útok:**

$$P_{\text{op}} + P_{\text{data}} = P_{\text{exploitable}} + P_{\text{sw. noise}}$$

($P_{\text{sw. noise}}$ - *switching noise* = datově nebo operačně závislá komponenta, která ale není využitelná zvolenou metodou útoku)

$$P_{\text{total}} = P_{\text{exploitable}} + P_{\text{sw. noise}} + P_{\text{el. noise}} + P_{\text{const}}$$

Signal to noise ratio:
$\text{SNR} = \frac{\text{var(signal)}}{\text{var(noise)}} = \frac{\text{var}(P_{\text{exploitable}})}{\text{var}(P_{\text{sw. noise}} + P_{\text{el. noise}})}$
Závisí na metodě útoku
Hodně měření, průměrování naměřených průběhů, snižuje SNR

Differential Power Analysis (DPA)

DPA: více naměřených průběhů spotřeby, analýza v jednotlivých bodech časové osy přes všechny

průběhy

Základní myšlenka:

- Zvolit vnitřní hodnotu šifry v, která **závisí na datech a tajném klíči:** $v = f(d, k)$
- **Naměřit průběhy spotřeby:** tvorba matice $\mathbf{T} = (t_{ij})$ během šifrování dat d_i (j udává počet vzorků v jednom průběhu)
- Sestavit matici $\mathbf{V} = (v_{ij})$ **hypotetických hodnot uvnitř šifry** pro všechny možné hodnoty klíče a hodnoty vstupních dat
- Pomocí modelu spotřeby vytvořit z matice $\mathbf{V}$ matici $\mathbf{H} = (h_{ij})$ **hypotetické spotřeby** (aplikace modelu spotřeby na naždou hodnotu v matici $\mathbf{V}$)
- **Statisticky vyhodnotit,** která hypotetická hodnota klíče nejlépe sedí na naměřené hodnoty v každý individuální čas

DPA na AES

Vnitřní hodnota šifry:
Typicky $v = \mathrm{SBOX}(d \oplus k)$ (hodnota po prvním AddRoundKey a SubBytes)

Naměření průběhů spotřeby:
Matice $\mathbf{T}$

Matice hypotetických vnitřních hodnot:
$\mathbf{V}$, kde $\mathbf{V}_{ij} = \mathrm{SBOX}(d_i \oplus k_j)$

Možné modely spotřeby:

- Hammingova váha: $HW(x) = \#\,\text{jedniček v binárním zápisu } x$
- Hammingova vzdálenost: $HD(x, y) = HW(x \oplus y)$ (počet rozdílných bitů)
- Single-bit: hodnota jednoho zvoleného bitu (např LSb)

Matice hypotetické spotřeby:
$\mathbf{H}$, kde $\mathbf{H}_{ij} = \mathrm{model}(\mathbf{V_{ij}})$

Vyhodnocení modelu spotřeby a naměřených průběhů:

- Pearsonova korelace matice $\mathbf{H}$ a matice $\mathbf{T}$ ($\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sqrt{\mathrm{var}X \mathrm{var}Y}}$) -- klíč je tam, kde je maximální
- Rozdíl průměrů (při binárních modelech, např. single bit):
  $\mathbf{M}_1 = (m_{1ij})$, kde $m_{1ij} =$ průměr počtu jedniček v každém sloupci matice $\mathbf{V}$

$\mathbf{M}_0 = (m_{0ij})$, kde $m_{0ij} =$ průměr počtu nul v každém sloupci matice $\mathbf{V}$
Statistika $\mathbf{R} = \mathbf{M}_1 - \mathbf{M}_0$

- Vzdálenost průměrů: Rozdíl průměrů dělený směrodatnou odchylkou

Obrana proti DPA - skrývání

Cíl: skrýt P_{op} a P_{data} tak, aby bylo minimalizováno $P_{\text{exploitable}}$

Spotřeba energie by měla být nezávislá na operacích a vnitřních datech:

- Konstantní spotřeba
- Náhodná spotřeba

Skrývání v čase

DPA potřebuje zarovnané průběhy spotřeby $\Rightarrow$ úmyslné rozházení průběhů (zpřeházení operací, vkládání dummy operací -- náhodných NOPů, vícero HW hodin)

Útok:

- Zarovnání naměřených průběhů -- pattern matching
  - Volba patternu (části průběhu)
  - Pattern se hledá ve všech průbězích, ty se pak posunou, aby na sebe patterny seděly
- Trace preprocessing:
  - Integrace průběhů -- součet několika po sobě jdoucích hodinových cyklů
  - Convoluce, FFT, ...

Skrývání v amplitudě

Vyovnání nebo randomizace spotřeby pro všechny operace $\Rightarrow$ snížení SNR

Minimalizace signálu:
$\mathrm{var}(P_{\text{exploitable}}) \to 0$
Všechny operace potřebují stejné množství energie -- speciální knihovny

Maximalizace šumu:

$\mathrm{var}(P_{\text{sw. noise}} + P_{\text{el. noise}}) \to \infty$

Náhodná aktivita hradel převýší spotřebu operací

Několik operací paralelně

Generátory šumu

Útok:

Útok na speciální logiku (chyby v paměťových buňkách, logických funkcích)

Maskování

Randomizací způsobuje nezávislost spotřeby na vnitřních hodnotách

Logické maskování:

- Aplikace logické funkce s náhodnou maskou na vnitřní hodnotu (např. vnitřní hodnota $\oplus$ maska)
- Lineární funkce: $f(x \oplus m) = f(x) \oplus f(m)$
- Nelineární funce: SBOX -- $S_m(x \oplus m) = S(x) \oplus m'$, kde m je vstupní maska a m' výstupní
  - Při sekvenčním zpracovávání hodnot $x \oplus m$ a m hrozí vyrušení: $HD(m, x \oplus m) = HW(x)$

Aritmetické maskování:

- Podobné jako logické, ale využívá aritmetické funkce:
  - multiplikativní homomorfismus RSA: $(ab)^d \equiv a^d b^d \pmod n$
  - ekvivaltentní exponenty RSA: $c^d \equiv c^{d+k\varphi(n)} \pmod n$

Implementace:

- Maskovací tabulky
- Random precharging:
  - inicializovat registr R na $R = m$
  - vnitřní hodnota nahrána do registru: $R = v$
  - únik $HD(v, m) = HW(v \oplus m)$

AES:

- AddRoundKey: klíč maskován m, data do AddRoundKey maskovány $k \oplus m$
- SubBytes: upravený SBOX -- $S_m(x \oplus m) = S(x) \oplus m'$
- ShiftRows: nemaskuje se

- MixColumns: jiná maska pro každý řádek stavu (v operaci se xoruje, takže by se jedna maska vyrušila)

Útok:

- Second order DPA:
  - Místo jedné vnitřní hodnoty v je pracováno s dvěma (maskovanými stejnou neznámou maskou).
  - Vnitřní hodnoty se musí objevit v různých operacích
  - Preprocessing průběhů:
    - Hodnoty jsou v různých hodinových cyklech: kombinace bodů
    - Hodnoty jsou ve stejném cyklu: Preprocessing každého bodu zvlášť

Kombinace bodů:

- Na průběhu odhadnout interval, kde se nejspíš nachází dvě sledované operace s hodnotami u a v
- Kombinace bodů intervalu pomocí funkce $\mathrm{pre}(t_x, t_y) \Rightarrow$ vznikne matice $\tilde{\mathbf{V}}$ (kombinace každého bodu s každým)
- Pokud pre symetrická, stačí část nad diagonálou
- Návrhy na pre:
  - $t_x \cdot t_y$
  - $|t_x - t_y|$
  - $(t_x + t_y)^2$
  - FFT
- S maticí $\tilde{\mathbf{V}}$ se provede klasicky DPA jako s $\mathbf{V}$ -- využití sdruženého rozdělení dvou hodnot, v DPA je s nimi pracováno jako s jednou $w = u \oplus v$