Hamming Weight Attacks on Cryptographic Hardware – Breaking Masking Defense

Marcin Gomułkiewicz (TU Wrocław)

Mirek Kutyłowski (TU Wrocław and CC Signet)

Modern ciphers

Modern ciphers use certain operations, more complicated than XORs, as their building blocks:

- modular addition (very common: IDEA, MARS, RC5/6, Twofish, ...)
- modular multiplication (quite common: IDEA, MARS, ...)
- ...

Side channel analysis; countermeasures

- each algorithm must be somehow implemented
- implementation in software or hardware
- hardware implementations often cause secret leakage
- popular countermeasure, *masking*: combining intermediate results a with random value r:

$$a+K=((a+r)+K)-r$$

goal: addition with the subkey on a random argument, any side channel characteristic of addition is random

Hamming weight assumptions

		1	1	1				1	1	1		←	*internals* (I)
	0	0	1	0	1	1	0	0	1	0	1	←	*first operand* (O_1)
+	1	0	1	1	1	0	1	0	0	1	1	←	*second operand* (O_2)
	1	1	1	0	0	1	1	1	0	0	0	←	*result* (R)

Hamming weight of binary number x: $|x|$ - amount of bits set to 1, in our example: $|I| = 6, \ |O_1| = 5, \ |O_2| = 7, \ |R| = 6$.

- standard assumption: $|O_1|$ and/or $|O_2|$ and/or $|R|$ - known to an attacker
- our assumption: $|I| + |O_1| + |O_2| + |R|$ - known to an attacker

Consequences of a new Hamming weight assumption

Observation: I heavily depends on O_1 and O_2.

- one of operands, say O_1, may be the (sub)key K we wish to find
- if another operand, O_2 is chosen at random (from uniform distribution), then R's distribution is also uniform, thus:
  1. distributions of $|O_2|$ and $|R|$ are easy to find; they do not depend on K
  2. distribution of $|I|$ depends on K only

Attack possibilities

- distribution of $|I|$ depends on K only
- but: dependence might be complicated
  and therefore **useless** for deriving the subkey
- **main point:**
  the dependence can be very well suited for a successful attack
- corollary: take care when implementing addition in hardware!

Properties of addition

Lemma: If $K = k_{n-1}k_{n-2}\ldots k_1k_0$ is added to random value chosen from uniform distribution, we expect to see C carry bits, where:

$$C = \sum_{i=0}^{n-1} k_i - 2^{1-n}K$$

Conclusion: We expect $|I| + |K| + |O_2| + |R|$ to be close to:

$$2\sum_{i=0}^{n-1} k_i - 2^{1-n}K + n$$

(obviously, n is known; typically $n = 16, 32$)

Properties of the formula

- expected value of the total Hamming weight can be quite well approximated as the mean value obtained for independent experiments,
- since n is known, it can be removed from the value

$$2\sum_{i=0}^{n-1} k_i - 2^{1-n}K + n$$

- the number

$$2\sum_{i=0}^{n-1} k_i - 2^{1-n}K$$

has some leading bits corresponding to the sum of key bits **followed by the binary representation of K!**

Key property

- obviously the Hamming weight depends on the key, but
- dependence is extremely useful for cryptanalysis:
  a part of the binary representation of the weight is the key itself
- moreover: the key is represented by almost the most significant bits

An attack: concept

- perform a large number of additions, collect Hamming weight data
- find the key bits the formula
- possible problem: errors and measurement inaccuracies?

Influence of errors

- problem: errors in such side channel data are unavoidable,
- different kind of errors: measurement inaccuracies, errors caused by randomization
- errors' impact on our formula: is it somehow "continuous", or maybe even small error can cause large changes?

Influence of errors (2)

- for our analysis we use (large) sums only
- if errors are independent, their sum can be very well approximated by Central Limit Theorem
  deviations from the expected value of the sum of k experiments
  (which is $k \cdot E$)
  oscillate around $\sqrt{k} \cdot E$
- $\Rightarrow$ for large k, errors do not influence the leading bits of the sum
- choose k large enough so that the errors do not influence at least some positions corresponding to key bits

Vulnerability of popular algorithms

- IDEA: 2^{20} samples and 2^{37} work (average), tradeoffs possible
- Twofish 128: 2^{44} samples and $\leq 2^{63}$ work (average), tradeoffs possible

Vulnerability of popular algorithms: theory

- Twofish 192 / 256: 2^{44} samples and 2^{95} / 2^{127} work, tradeoffs possible
- MARS: 2^{44} samples suffices to find 320 out of 1280 bits of expanded key, tradeoffs possible
- RC5/6 with r rounds operating on n-bit long strings: with equipment of indefinite accuracy at most $2^{\frac{n}{2}}(2+2r)$ samples would allow us to duplicate encrypting device or decipher messages

Conclusions and open problems:

- even if the analysis reveals only leading bits of subkeys it may happen that these key bits reconstruct almost the whole key
  $\Rightarrow$ be careful with key schedule if using addition!
- addition is particularly well suited for this kind of attack,
  other operations?
- masking does not prevent the attack, it even helps by making input to addition fully random!
  masking that prevents attacks based on analysis of a single event may facilitate attacks based on global behavior.