

Pinpointing the Side-Channel Leakage of Masked AES Hardware Implementations

Stefan Mangard

Graz University of Technology

Kai Schramm

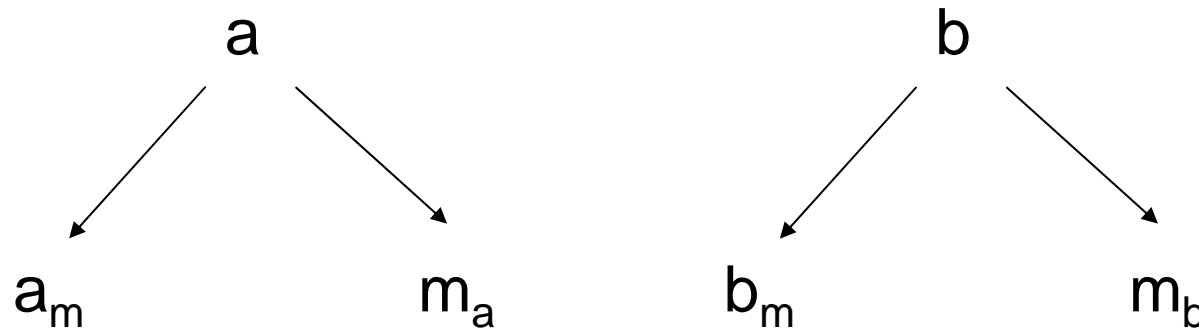
Ruhr University Bochum



- Masking Schemes
- Leakage of Masking Schemes in General
- Leakage of Masked Multipliers
- Leakage of a Masked AES S-Box
- Countermeasures
- Conclusions

Masking Schemes

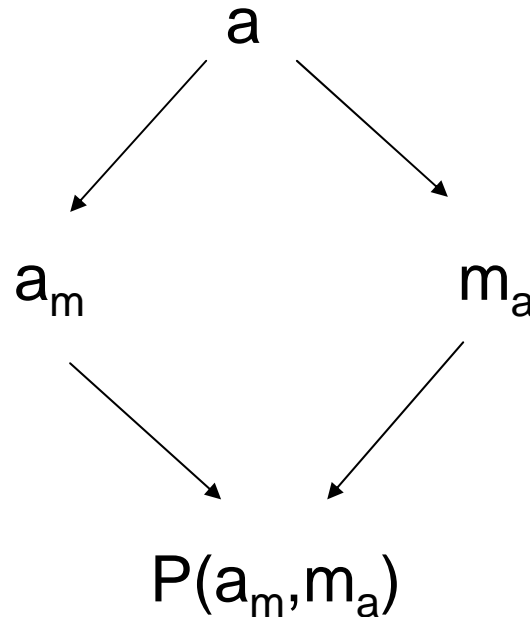
- Goal: Make the power consumption independent of the intermediate values of the cryptographic algorithm



$$I_1 = f_1(a_m, m_a, b_m, m_b), \dots, I_n = f_n(a_m, m_a, b_m, m_b)$$

- If all intermediate values are pairwise independent of the unmasked values, the scheme is provably security

$$a = a_m \oplus m_a$$

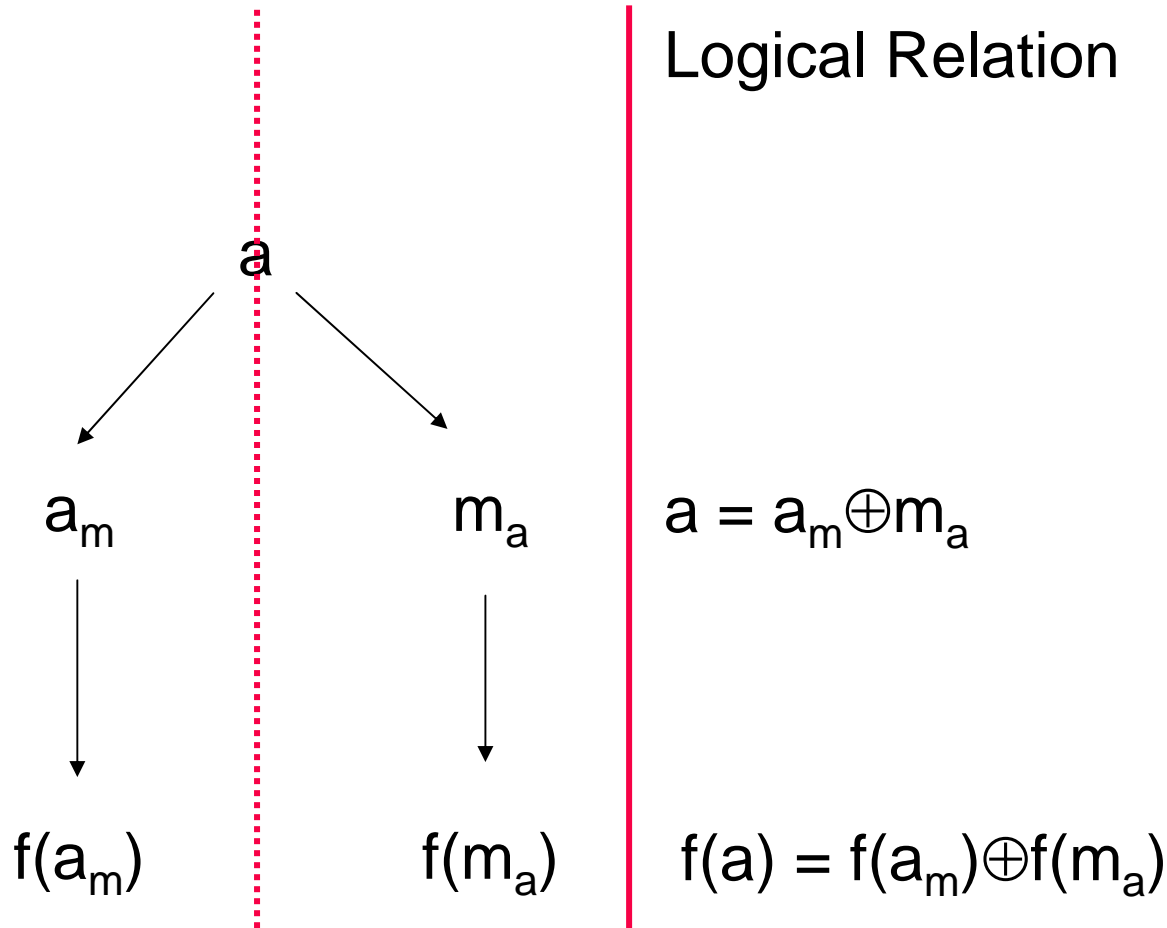


Algorithm

Device

$P(a_m, m_a)$

- Power consumption $P(a_m, m_a)$ depends on a
- The degree of the dependency is influenced by many factors

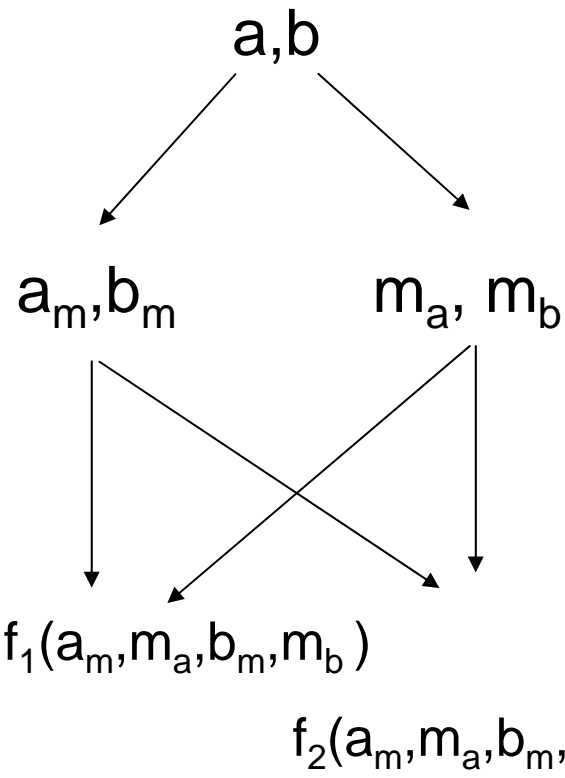


Additive Power Consumption

$$a = a_m \oplus m_a$$

$$P(a_m, m_a) = P(a_m) + P(m_a)$$

- The average power consumption is equal for all possible values a
- This holds for all power consumption functions $P()$
- The statistical distribution of the power consumption is not equal for all values a , *i.e.* it is not independent of a !
- Dependency can be exploited by templates, test on other parameters, ...
- Non-linear preprocessing (e.g. squaring) of the power consumption $\text{pre}(P(a_m, m_a))$ can be used to make the average of $P(a_m, m_a)$ depend on a



Logical Relation

$$a = a_m \oplus m_a$$
$$b = b_m \oplus m_b$$

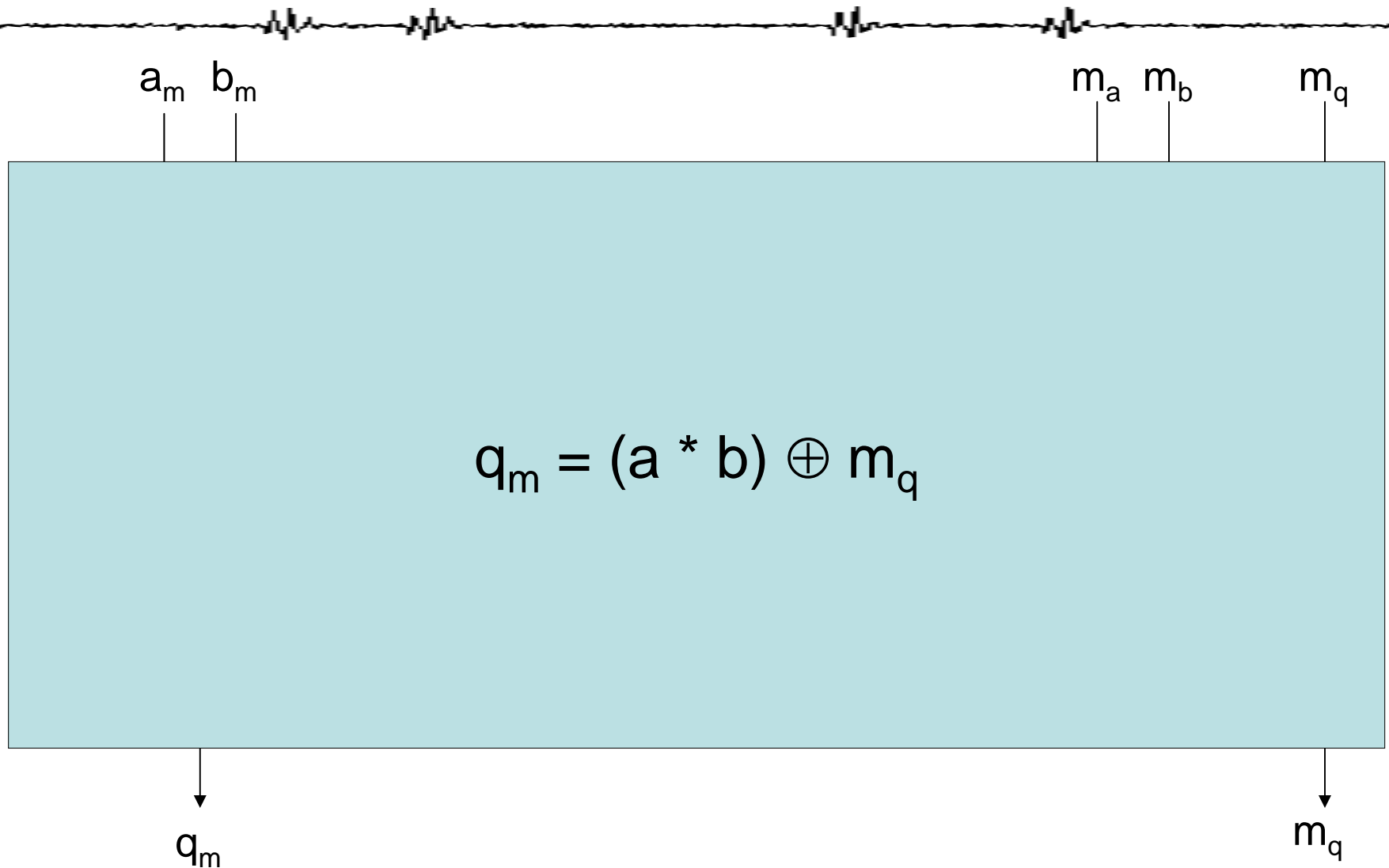
$$f(a, b) = f_1(a_m, m_a, b_m, m_b) \oplus f_2(a_m, m_a, b_m, m_b)$$

Power Consumption

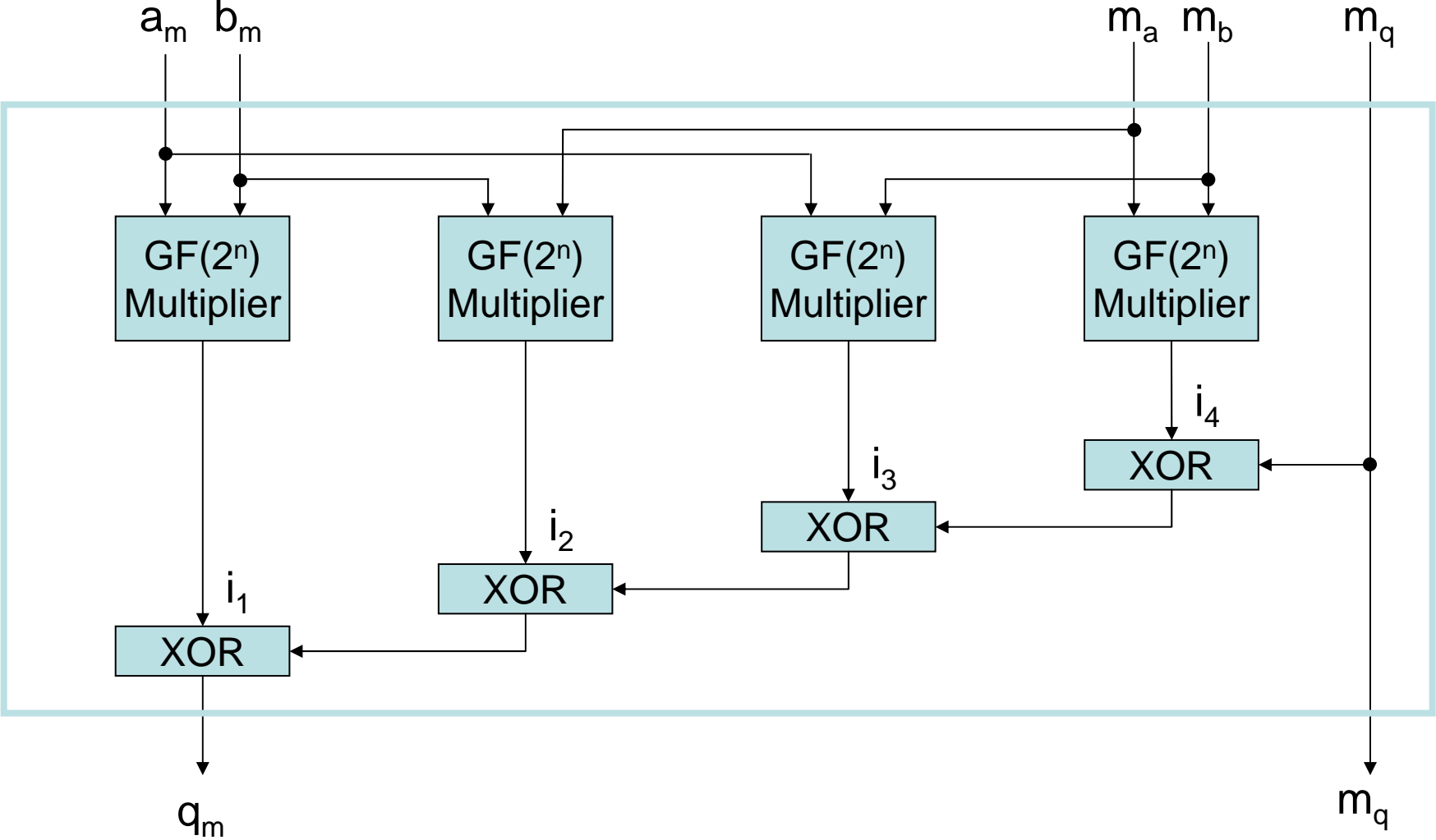
$$P(a_m) + P(m_a) + P(b_m) + P(m_b)$$



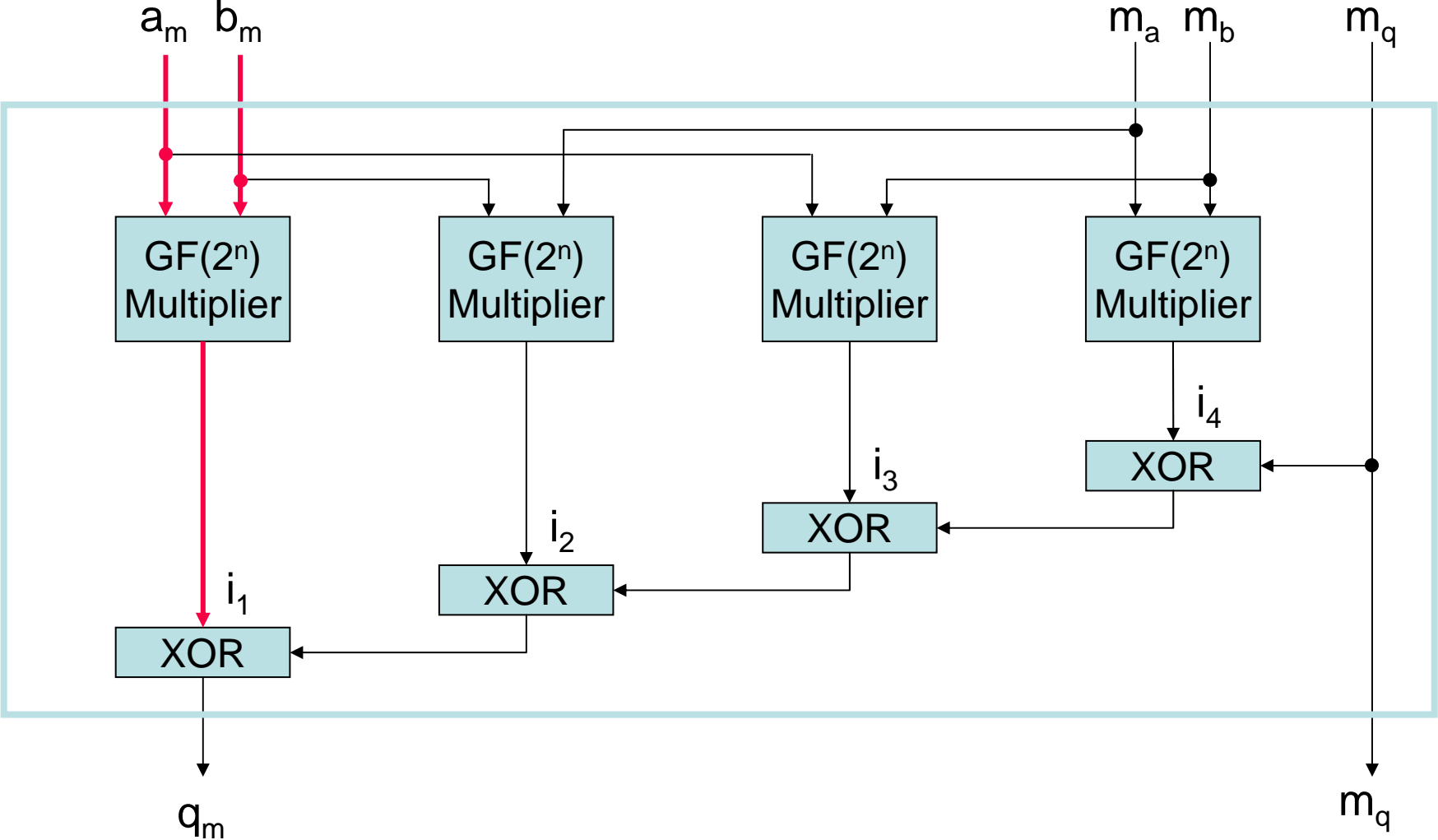
Example: Masked Multiplier



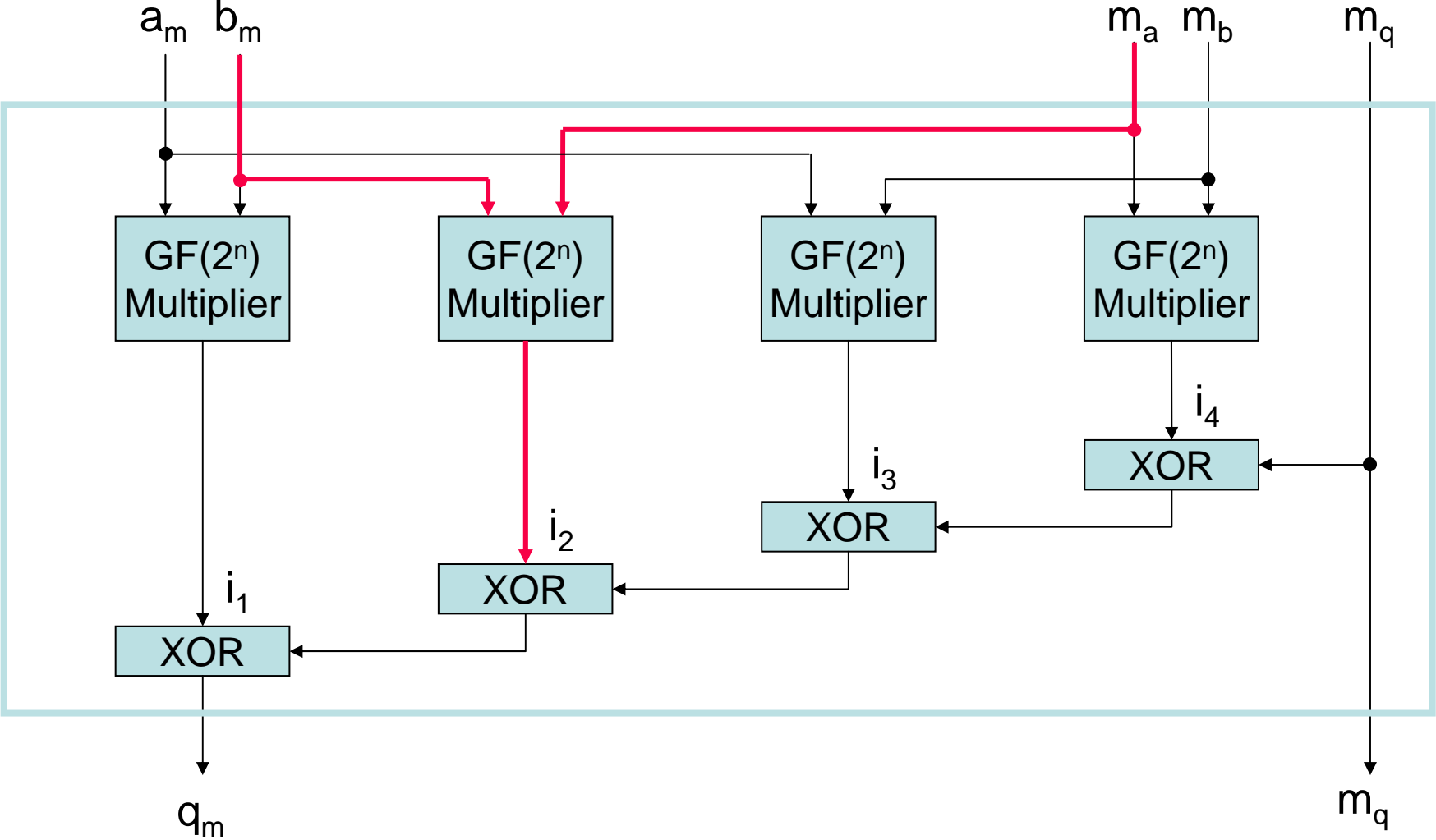
Example: Masked Multiplier



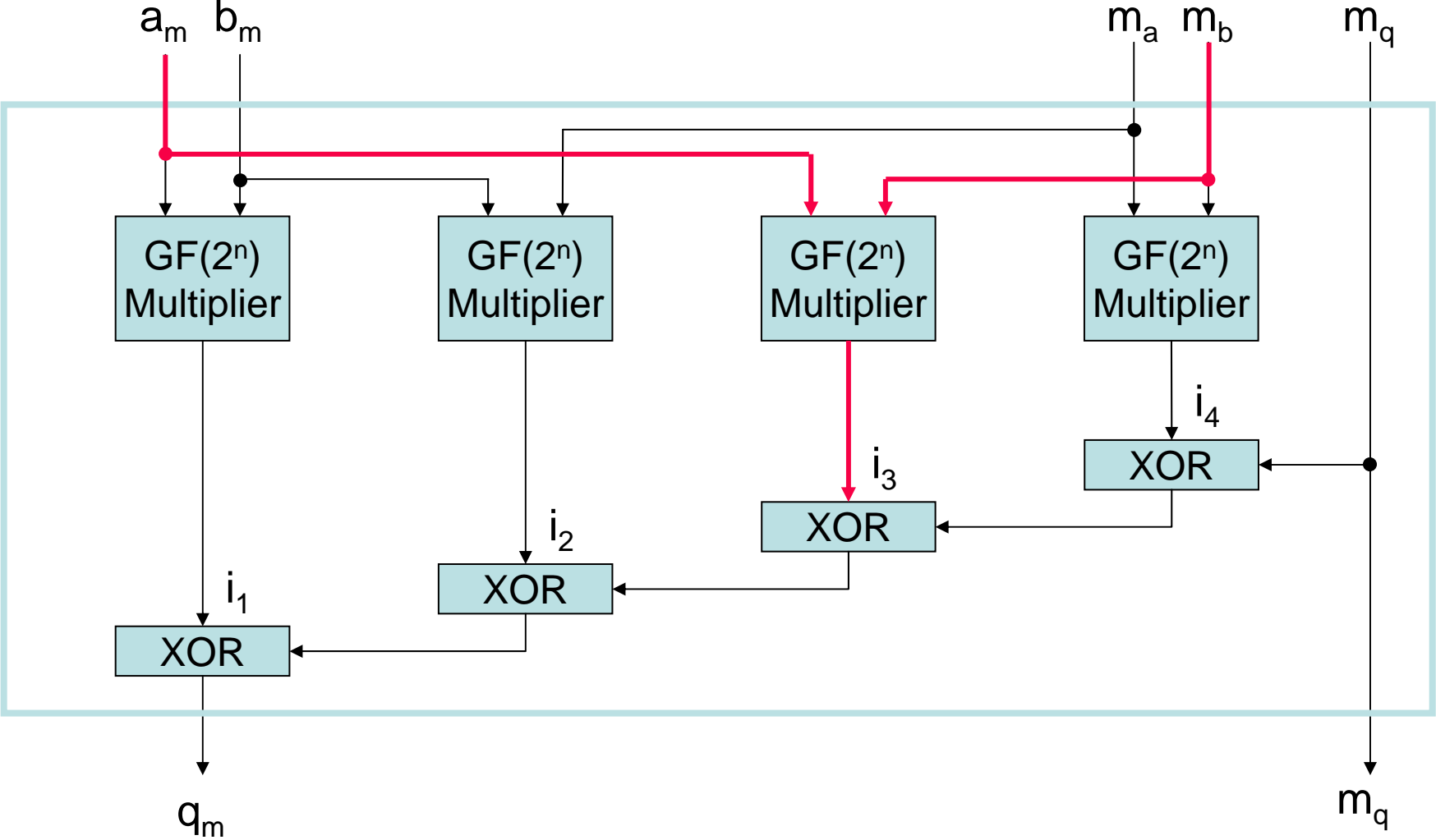
Example: Masked Multiplier



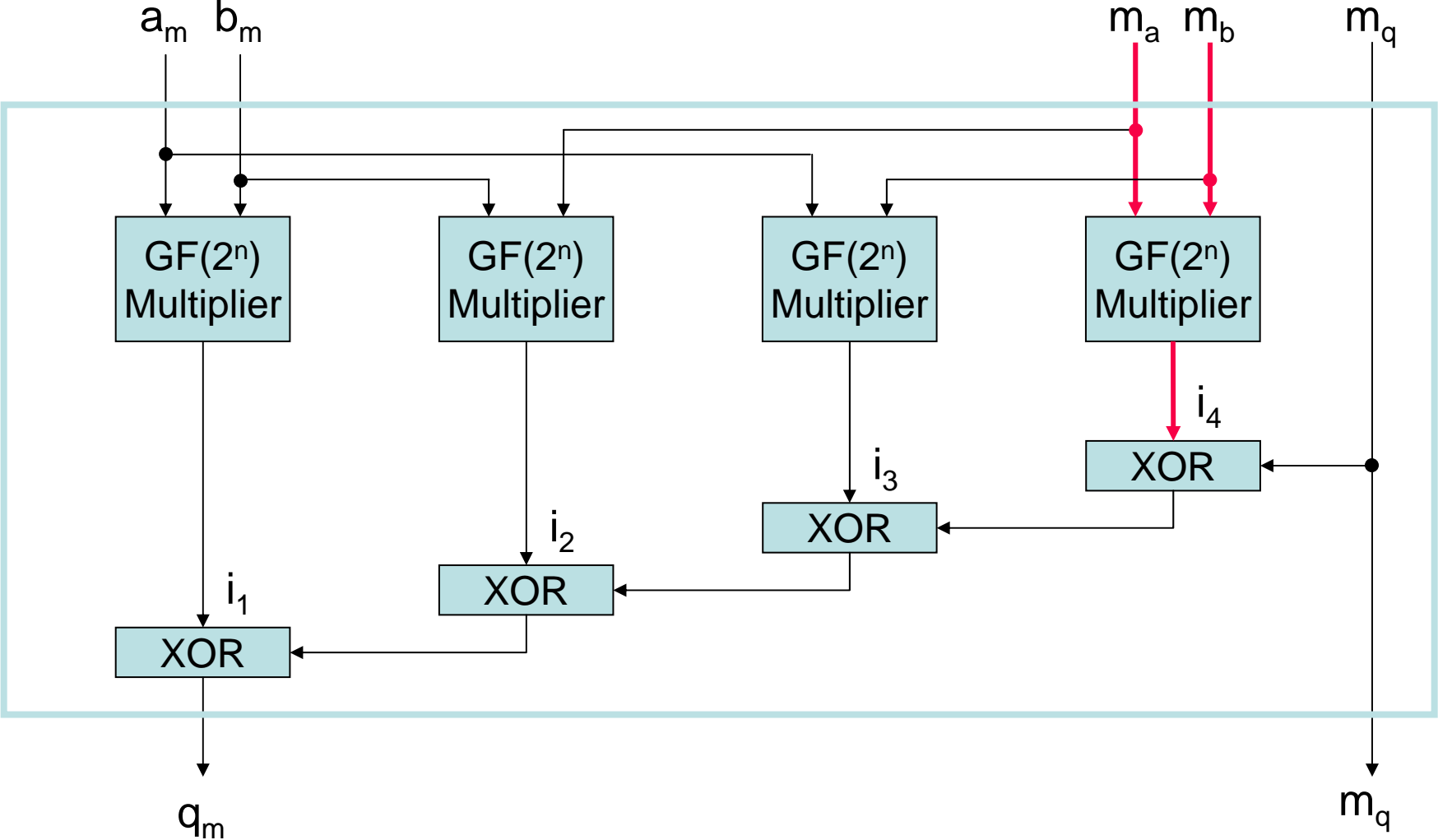
Example: Masked Multiplier



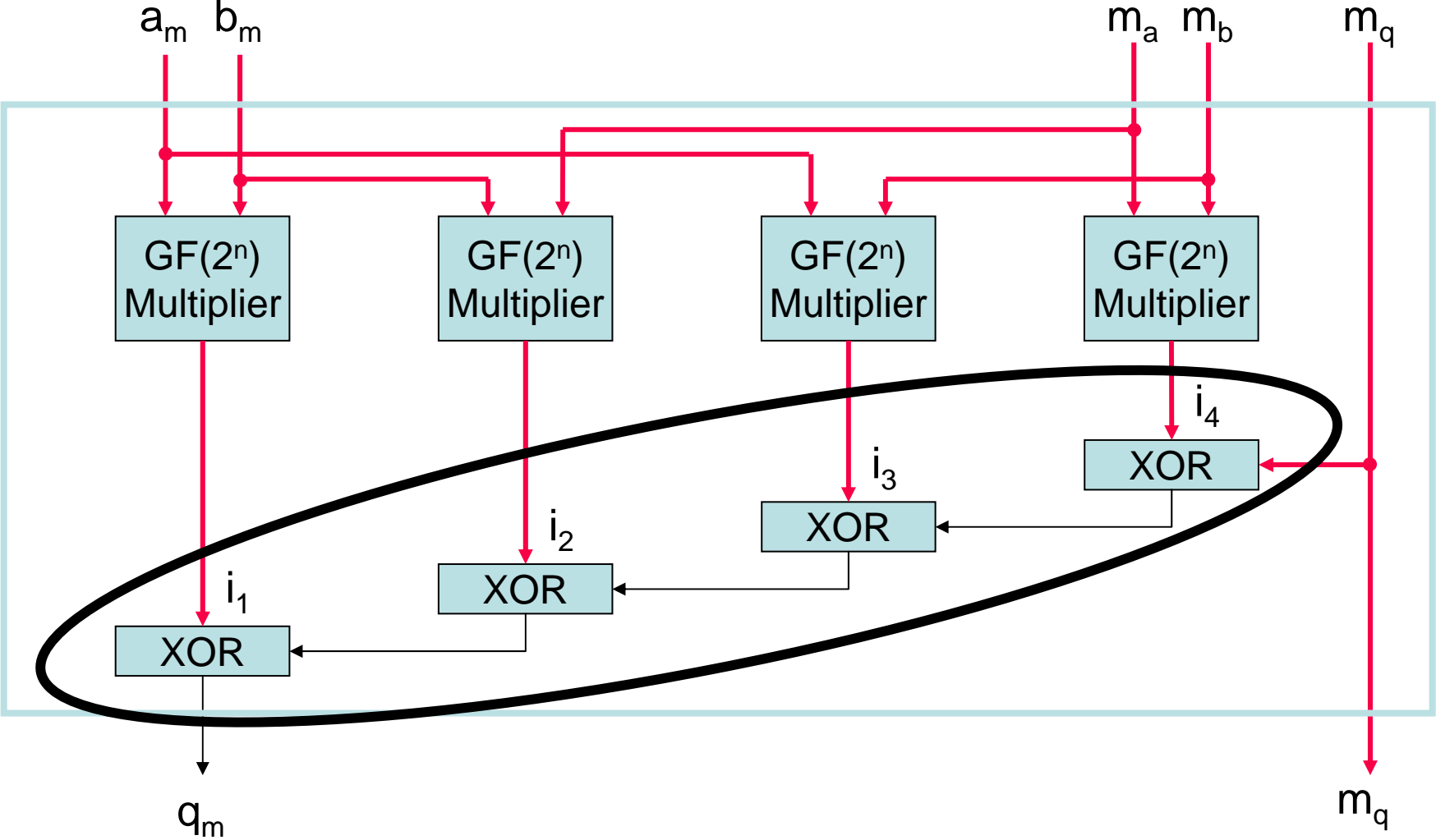
Example: Masked Multiplier

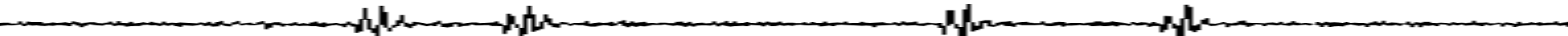


Example: Masked Multiplier

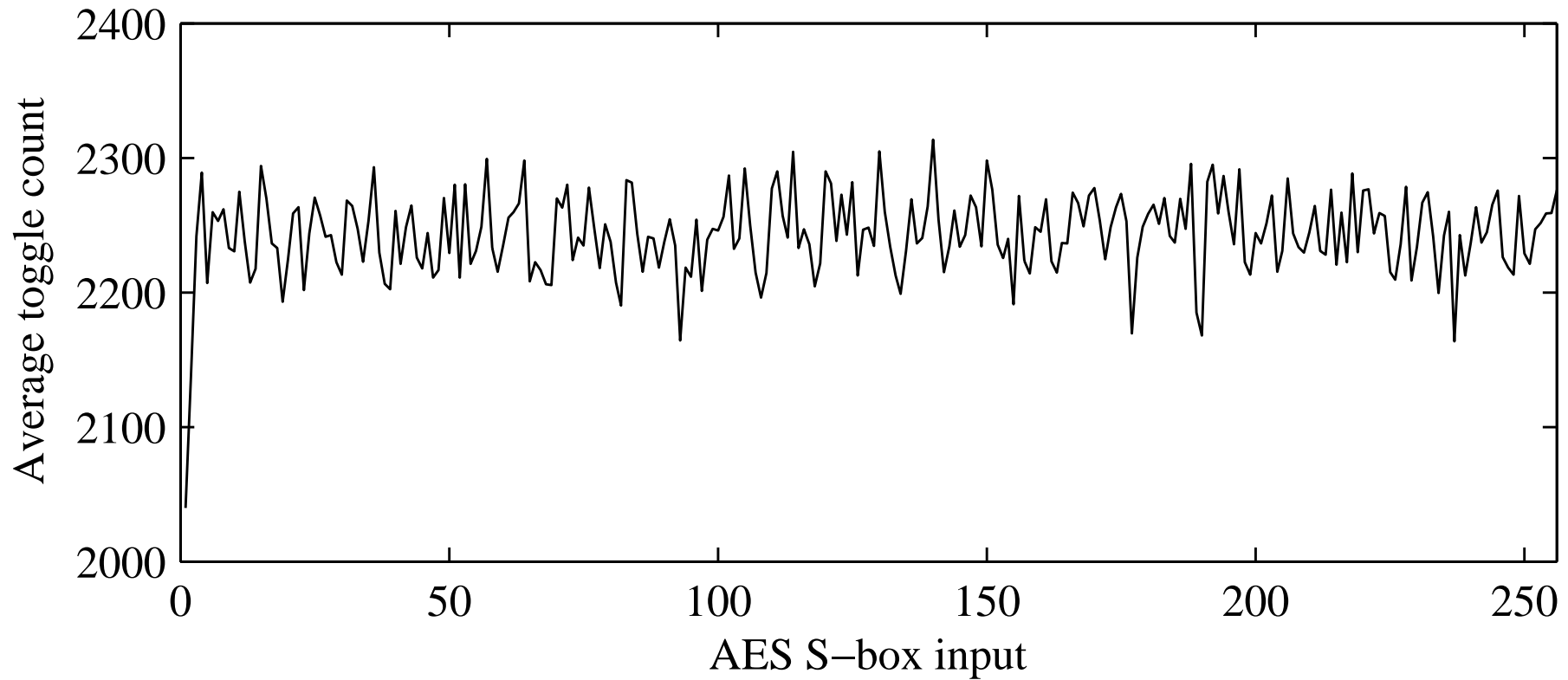


Example: Masked Multiplier



- 
- The power consumption of the multipliers is independent of the unmasked values a , b , and q
 - XOR gates switch whenever an input switches
 - If two inputs change within a short period of time, no transition occurs at the output. The input transitions are absorbed.
 - Arrival times of the signals at the XOR gates depends on the unmasked values
 - The power consumption depends on the unmasked values

Average number of Transitions of an AES S-Box



- Whenever masked value and mask arrive at the same gate there is a potential problem
- The number of transitions that occur at the output depend on the joint distribution of the arrival times of the masked value and the mask
- Data-dependent power consumption transitions also occur in parts that are connected to the gates processing masks and masked values -> avalanche effect

- Data-dependent “absorption” of transitions needs to be avoided
- Absorption of transitions can for example be prevented by
 - Dual-rail pre-charge circuits
 - Enable signal
- There still occurs parallel processing of masked values and masks

- The power consumption of masked implementations depends on the corresponding unmasked values
- Distinguish between sequential, parallel, and joint processing of masked values and masks
- For sequential and parallel processing no general optimal preprocessing has been proposed so far
- Joint processing typically leads to more significant leakage
- Countermeasures to protect implementations of joint processing are for example dual-rail encoding and enable signals

www.dpabook.org

Stefan Mangard, Elisabeth Oswald, and Thomas Popp

