Masking Does Not Protect Against Fault Attacks

Arnaud.Boscher@spansion.com

Helena.Handschuh@spansion.com

SPANSION®

Masking techniques on AES

- Cryptographic algorithms are susceptible to power analysis attacks, i.e. measure power consumption, do some statistical analysis and extract the secret keys.
- Masking : for each AES algorithm execution, add a *random vector u* to the plaintext bytes, *a random vector w* to the secret key bytes, and start the computations with u, and w.
- compute the algorithm for both the « masked plaintext » and the « mask » independently
- at the end, add both parts together and recover the correct ciphertext

Example: Boolean masking for AES

- **Everything goes fine except for the non-linear Sboxes (table look-up implementation).**
- **Recompute each Sbox such that** $S_u(m \oplus u) = S(m)$.
- **Remask with v at the output:** $S^v_u(m \oplus u) = S(m) \oplus v$.
- **Close-up look on last-round computation on masked state s (ignore Shiftrow here):**
- $c = ((S^v_u(s) \oplus (k \oplus w)) \oplus v) \oplus w$
- **Consider 2 different runs with the same plaintext, same key, but different random vectors u,v,w.**
- **Ciphertexts will be the same .**

Effects of a single bit-flip e_j in the masked state s

- **Introduce a single bit-flip e_j in masked state s of one of the runs.**
- $c = ((S^{v}_{u}(s) \oplus (k \oplus w)) \oplus v) \oplus w$
- $c = ((S^{v^*}_{u^*}(s^*) \oplus (k \oplus w^*)) \oplus v^*) \oplus w^*$
- $c^* = ((S^{v^*}_{u^*}(s^* \oplus e_j) \oplus (k \oplus w^*)) \oplus v^*) \oplus w^*$

- **Add the correct and the faulty ciphertexts.**
- $c \oplus c^* = ((S^{v}_{u}(s) \oplus (k \oplus w)) \oplus v) \oplus w \oplus ((S^{v^*}_{u^*}(s^* \oplus e_j) \oplus (k \oplus w^*)) \oplus v^*) \oplus w^*$
- $c \oplus c^* = (S^{v}_{u}(s) \oplus v) \oplus (S^{v^*}_{u^*}(s^* \oplus e_j) \oplus v^*)$

- $c \oplus c^* = S_u(m \oplus u) \oplus S_{u^*}((m \oplus u^*) \oplus e_j)$.
- $c \oplus c^* = S_u(m \oplus u) \oplus S_{u^*}((m \oplus e_j) \oplus u^*)$.
- $c \oplus c^* = S(m) \oplus S(m \oplus e_j)$

- **even though the faulty computation S(m ⊕ ej) never actually took place.**

Key Recovery

- $c \oplus c^* = S(m) \oplus S(m \oplus e_j)$
- **Apply differential cryptanalysis on S or S^{-1} and recover one last round subkey byte (a few values may be left here)**
- **Requires 16 faulty ciphertexts plus their correct versions.**
- **Plus some exhaustive search or a second faulty ciphertext for each byte since a few different keys remain after the filtering stage.**
- **This example requires a very strong and precise fault model**
- **Can be generalised to weaker fault models, other rounds, other masking techniques but the conclusion remains the same...**
- **We have plenty more examples ;-)**

Helena.Handschuh@spansion.com