Security of assembly programs against fault attacks on embedded processors - Abstract

Nicolas Moro

This thesis focuses on the security of embedded programs against fault injection attacks. Due to the spreadings of embedded systems in our common life, development of countermeasures is important.

Keywords : fault injection attacks ; electromagnetic injection ; fault model ; verified countermeasures ; assembly ; instruction skip

1 Assembly-level fault model

First, a fault model based on practical experiments with a pulsed electromagnetic fault injection technique has been built. The experimental results show that the injected faults were due to the corruption of the bus transfers between the Flash memory and the processor?s pipeline. Such faults enable to perform instruction replacements, instruction skips or to corrupt some data transfers from the Flash memory. Some works about the definition of the fault model have been published in [1].

2 Fault-tolerance countermeasure

Although replacing an instruction with another very specific one is very difficult to control, skipping an instruction seems much easier to perform in practice and has been observed very frequently. Furthermore many simple attacks can carried out with an instruction skip. A countermeasure that prevents such instruction skip attacks has been designed and formally verified with model-checking tool. The countermeasure replaces each instruction by a sequence of instructions. An example of the use of this countermeasure for a `bl` subroutine call instruction is presented in listing 1. Some works about the definition of this countermeasure and its formal verification have been presented in [2] and a detailed version has been published in [3].

Listing 1: Replacement sequence for a `bl` instruction

```
    adr r12,return_label  ; update of the return pointer
    adr r12,return_label
    add lr, r12, #1      ; the last bit of the pointer is set to 1
    add lr, r12, #1      ; (specification for Thumb mode)
```

```
  b   function          ; branch to the subfunction
  b   function

return_label            ; destination of the return pointer
```

3 Experimental evaluation of the countermeasure

However, this countermeasure does not protect the data loads from the Flash memory. To do this, it can be combined with another assembly-level countermeasure that performs a fault detection. A first experimental test of these two countermeasures has been achieved, both on isolated instructions and complex codes from a FreeRTOS implementation. The proposed countermeasure appears to be a good complement for this detection countermeasure and allows to correct some of its flaws. Some works about the experimental evaluation of these two countermeasures have been published in [4].

References

[1] N. Moro, A. Dehbaoui, K. Heydemann, B. Robisson, and E. Encrenaz, "Electromagnetic Fault Injection: Towards a Fault Model on a 32-bit Microcontroller", in *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, Santa Barbara, California, USA: IEEE, Aug. 2013, pp. 77–88. DOI: 10.1109/FDTC.2013.9.

[2] K. Heydemann, N. Moro, E. Encrenaz, and B. Robisson, "Formal verification of a software countermeasure against instruction skip attacks", in *2nd Workshop on Security Proofs for Embedded Systems (PROOFS)*, Santa Barbara, California, USA, 2013.

[3] N. Moro, K. Heydemann, E. Encrenaz, and B. Robisson, "Formal verification of a software countermeasure against instruction skip attacks", *Journal of Cryptographic Engineering*, vol. 4, no. 3, pp. 145–156, Feb. 2014. DOI: 10.1007/s13389-014-0077-7.

[4] N. Moro, K. Heydemann, A. Dehbaoui, B. Robisson, and E. Encrenaz, "Experimental evaluation of two software countermeasures against fault attacks", in *2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, Arlington, Virginia, USA: IEEE, May 2014, pp. 112–117. DOI: 10.1109/HST.2014.6855580.