Scan Design and Secure Chips : Can They Work Together

David Hely, Frédéric Bancel, Marie-Lise Flottes, Bruno Rouzeyre

SAME 2005 Forum

Session 3 : Method Chip Design

Scan Design and Security: Can They Work Together?

David Hély[1&2], Frédéric Bancel[1], Marie-Lise Flottes[2], Bruno Rouzeyre [2]

[1] Smartcard Division, ST Microelectronics Rousset, France
[2] LIRMM/UMR 5506 CNRS, Université Montpellier II, France
david.hely@st.com

Abstract:

Scan based testing is one of the most used and powerful test technique since it provides full observability and controllability of the internal nodes of the IC. From a security point of view, the drawback of such testability capabilities is that using scan in secure chip, for instance in a cryptographic one, may seriously decrease the security level so that it makes this technique not acceptable. Some countermeasures have been proposed in order to secure the scan technique. In this paper we present secure scan countermeasures and show how scan and security can live together with a real example.

1. Introduction

Testing is primordial in order to reach a good level of quality, this being even more important when security is involved. Can we imagine to deliver to the customers supposedly secure chip which may fail due to process issues? No, of course, because a malfunction of the circuit can provoke major security vulnerabilities. Just imagine a stuck at fault on a path enabling access to the memory… Thus in addition to test for quality, test for security must also be considered. In order to reach a correct level of testability, secure IC designers may also imply design for testability principles in addition to design for security ones. However, while, design for testability principles increase both the observability and the controllability of the design, design for security induce to reduce both parameters to the minimum, protecting secret data processed on chip. Then, there is a need for new DfT approaches taking into consideration security requirements in order to make both testability and security live together.

In this paper, we will first present scan based attacks principles and the way they can be realized. Then, countermeasures securing the scan against such attacks will be described. Finally, a scan based attack will be applied on a cryptographic chip having a secure scan architecture in order to measure the efficiency of one of the proposed countermeasure.

2. Scan Chain Attacks

2.1 Scan chain attack principle

In [Yan04] and [Yan05], authors demonstrate how breaking both DES and AES algorithm using the scan chain, considering of course that secret keys registers are not part of the scan chain. Such attacks consist in unloading the scan chain at different time of the cryptographic process with different plaintexts. Comparing scan chain unloads bit to bit, hackers first retrieve the cryptographic registers in the scan chain (i.e. identify which bits of the scan chain corresponds to a register of the hardware implementation of the algorithm), then they retrieve secret keys by comparing and processing the value being handled during the encryption. The main concept of such attacks relies on the capability of the hacker to switch from test mode (the scan chain is activable) to system mode (cryptographic process is running).

2.2. The context

A trivial countermeasure to the scan chain attack consists in protecting scan chain port using additional authentication schemes. We consider in the following an on-chip test control bloc so that scan chain ports (scan_enable, scan_in and scan_out) are only accessible trough the controller. Making the controller usable only after a strong authentication of the test engineer provides thus a first countermeasure against scan chain attack. Figure 1 , describes the scan architecture we consider for the security evaluation.

2.3. Scan Attacks realization

From now, scan chain attack is more difficult to implement since hackers must either use the test controller as a test engineer or directly activate the scan chain by probing the scan chain signals directly. The first scenario is possible at the condition the hacker has been able to gain illicitly the authentication key or thanks to a malfunction of the authentication scheme. In this case, the hacker just switch form system mode to test mode just doing like a test engineer would do. In the second case, a bypass of the test controller can be imagined. Accessing the scan chain and thus all the scan flip-flops requires only two probes, one for the scan_enable in order to activate the scan chain, and an other one on the scan_output in order to observe the scan data flow. Of course probing current IC design is more and more difficult with the technology shrinking. Nevertheless with continuously empowering tools such as FIB or IDS PICA, such attacks may not be neglected. A probing attack aiming at setting the internal content of a register such for instance the one of the register acknowledging a correct authentication is some quite unrealistic. Indeed, this would require first to localize precisely the register in the design (with no access to the design data base this is almost like looking for a needle into a haystack), and secondly to use much more probes that can be realistically be set on an IC processed in state of the art technology since such registers are commonly protected using redundancy and checkers.

On the other side, probing the scan chain requires only two probes and identifying a scan flip-flop on the layout. Indeed once the flip-flop have been identified probing the scan_enable and observing the flip-flop output gives full access to the flip-flop data preceding the one being probed in the scan chain. Once a scan flip flop is identified, it may also be possible to identify the whole scan chain by following connections until the chip scan output.

3. Countermeasures

3.1 Test Mode countermeasure

As described above, if the hacker is able to use the test controller as a test engineer would do, scan chain data are directly accessible. Since scan based attacks rely on the possibility to switch from system mode and test mode so that scan data have closest link with secret key processed by the cryptographic algorithm, it has thus been proposed in [Yan04] and [HEL05] to modify the test controller so that even if one can identify himself as a test engineer, data which flow through the scan chain have no links with the cryptographic data processed in system mode. Such a countermeasure is simply realised by adding a reset process in the test controller finite state machine. Thus once the authentication has been passed, first the

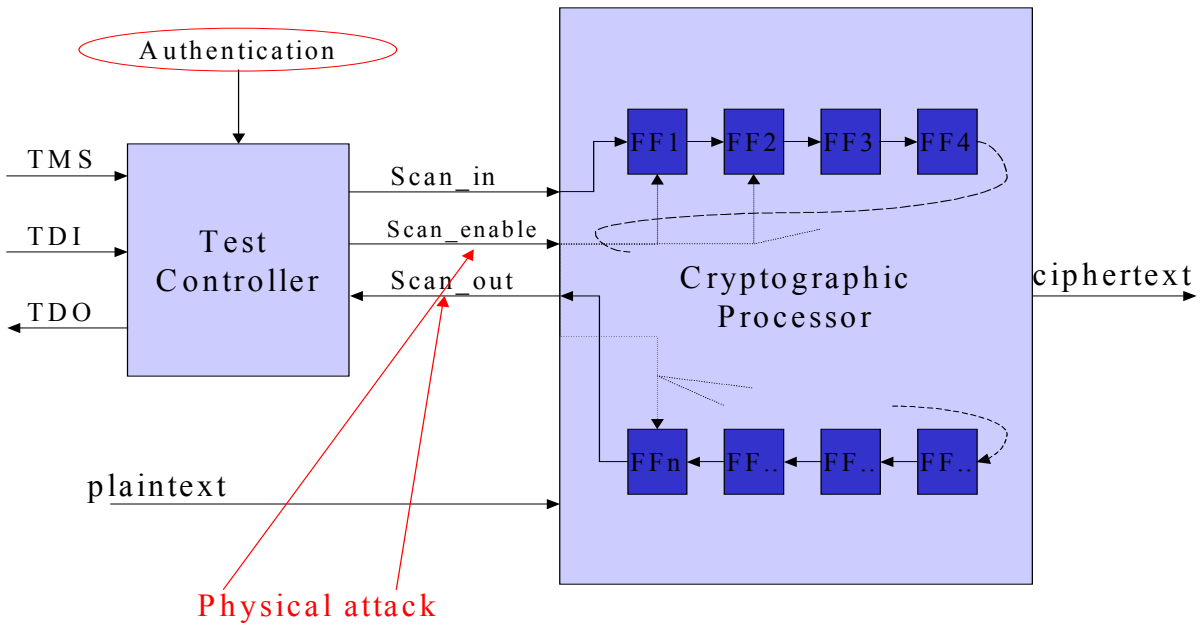

Figure 1: Scan architecture and possible attack

cryptographic part of the chip is reset. Then this reset is verified and if this one is correct, then scan operations are possible, each time scan is activated it is checked that the reset has been correctly done.

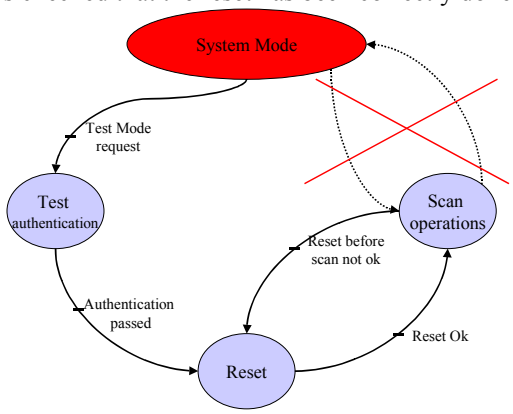

Figure 2: Test Controller Modification

In [Hel05], we propose to adapt the IEEE 1149.1 test controller, so that a state is added into the finite state machine managing the reset step required for security. Such a countermeasure makes then the scan operations secure even if one has access to all features dedicated only to test engineers.

3.2 System mode countermeasures

Since the scan chain can also be accessible by physical attack bypassing the test controller, the previous countermeasure does not provide any protection against such attack. It is then necessary to secure the scan architecture at the chain level. Protecting the scan chain at this level can be made according two different points of view. On one hand designers can decide to not allow any scan chain activation in system mode, a system thus checks scan chain shiftings in system mode[Hel05]. On the other hand, one can make scan chain unloads of scan chain during system not exploitable by hackers by modifying the scan chain structure in system mode [Hel04].

The first countermeasure consists in checking the scan_enable signal in system mode. Indeed, since any scan chain activation relies on a switching of this one from 0 to 1, checkers are inserted into the scan_enable tree. These ones check that the scan enable root is stuck at 0 during system mode and that all the branches have always the same value (indeed, in case the probe is placed after a buffer, the checker at the root of the signal would not be able to detect anything).

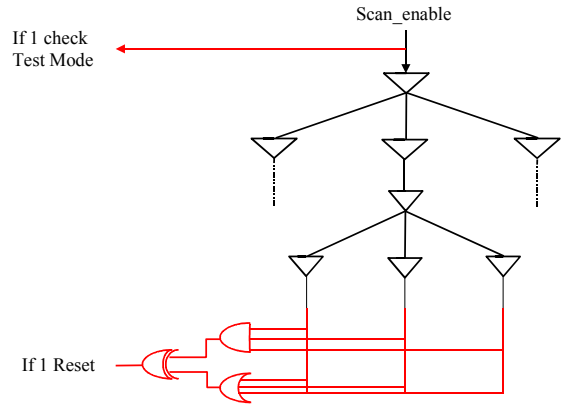

Figure 3: Scan_enable protection

The second proposed countermeasure consists in modifying the scan chain structure while in system mode. Since scan based attacks rely on comparing different unloads bit to bit, in system mode the scan chain is divided into segments which are dynamically connected together randomly. The proposed scan chain scrambling countermeasure dynamically re-orders the bit-sequence of a scan chain during unauthorized scan out operations. Since between to unloads the scan chain order has been dramatically modify, it is not possible to compare them bit to bit. controls the order of the scan chain elements in such a manner that:

- When the scan mode has been securely reached (before the fuses are blown and after a strong authentication for instance), the scan chain elements order is fixed to a predetermined order.
- When the chip is not in test configuration, the order of scan chain elements changes at a given frequency.

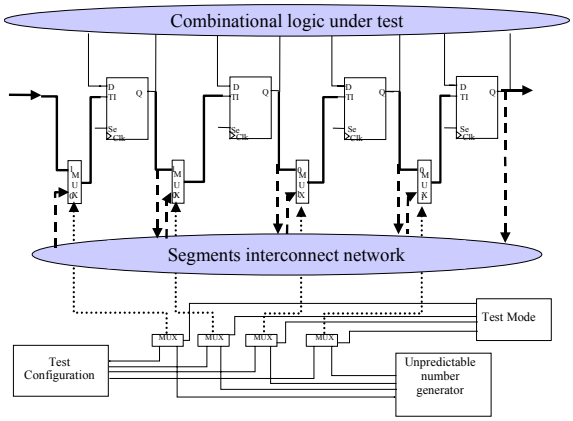

Figure 4: Scan chain scrambling

In order to perform such scrambling multiplexers are inserted between scan chain segments. The test input of the i^{th} segment is fed by the output of multiplexer; the inputs of the multiplexer can either come from the $(i-1)^{th}$ segment (in test mode) or from any of the segments connected to this multiplexer through the scrambler. A scrambling

controller generates the control signals of the multiplexers inserted between the scan chain segments. During the test mode, a test key allows to certify the validity of the mode of operation. The scrambler controller reads this key and generates adequate control signals in order to connect the scan chain segment in the appropriate and fixed order. In any other mode of operation, or when the test key is not valid, the scrambling controller sends random values to the multiplexer control inputs.

4. Application on scan based attack

4.1 The secure architecture

In this section we propose to apply the attack described in [Yan04] to a chip having a secure scan architecture. In order to overcome both kind of attacks described in section 2, the architecture contains a secure test controller with the reset feature and a scan chain scrambler in order to protect the scan chain in system mode. A correct authentication signal disables the scrambler mechanism. When an incorrect value is applied, however, the scan chain scrambler is enabled. The scan chain segments are randomly connected together according to the value applied on *Unpredicr_data* from the random number generator. Most of present crypto chips include an embedded hardware random number generator used for generating secret keys. This generator can be reused for providing true random numbers to the scrambler block.

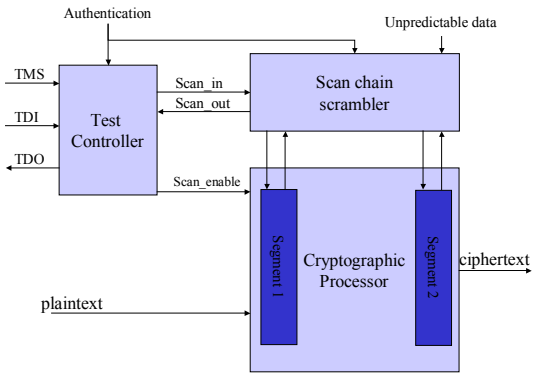

Figure 5: secure scan architecture

The scan chain scrambling parameters are chosen as follows:

- The scan chain is divided into two segments
- The segments connections are refreshed at the frequency F_{scram}, This frequency is set by the designer at the conception phases of the scrambler feature.
- According to the value *Unpredict_data*, which drives the muxes between the segments, the scan chain order changes as described in figure 6. If the signal *test_key* is correctly provided, the value *unpredict_data* is fixed to "000" and thus the connections are fixed.

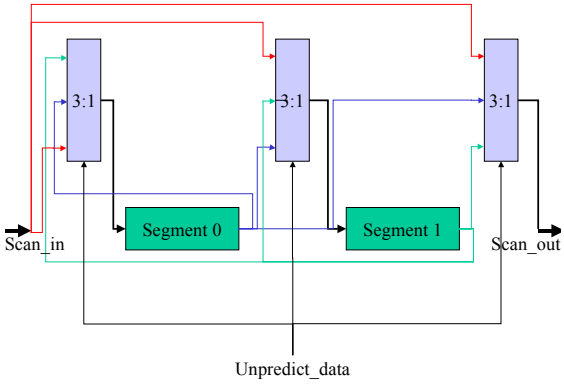

Figure 6: scan chain with scrambling facilities

4.2 Resistance to the attack

Since the efficiency of the attack presented in section 4 relies on the capability of determining the scan chain structure, we propose to validate the scan chain scrambling based countermeasure on this particular step of the attack. In the following sections we are going to retrieve the position of an R register bit in the scan chain by applying the methodology described in [Yan04]. We then apply two different plaintexts (PT^1 and PT^2) having one bit difference so that $PT^1_{10} \neq PT^2_{10}$. Unloading the scan chain after two clocks cycles, one difference should be observed between the two bit streams corresponding to the $IP_{10}=R\_reg_{17}$. Thus, in the following sections we try to retrieve this register for both cases where the scan chain scrambler is on and off.

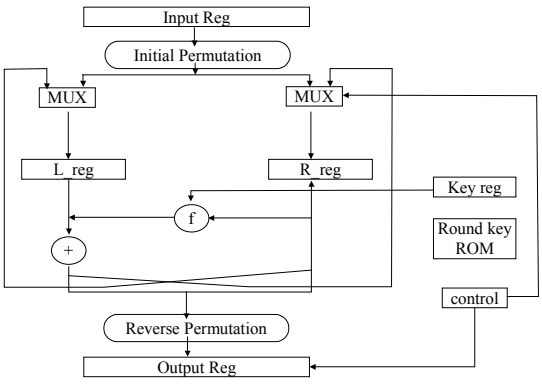

Figure 7: DES architecture

In this last case, it is assumed that the signal *Auth* has been correctly set according to the key dedicated to test engineer. The scan chain scrambler is thus off and the segments are

connected together corresponding to the configuration for the one the test engineer has processed the pattern generation. We perform the attack described above with plaintexts $PT^1_{63:0}$ and $PT^2_{63:0}$ so that $PT^1_{10} \neq PT^2_{10}$. After two clock cycles the scan chain is unload for both plaintexts. Comparing the two bit streams, we found one bit difference at position 16 which corresponds to $IP_{10}=R\_reg_{17}$. Bit 17 of R register has been successfully located in the scan chain.
Now, the scan chain is activated whereas a bad authentication key is provided (i.e. the authentication signal is not set to the correct value). Thus, the scrambler module is on. The signal muxes are controlled by a random number generator, the value is refreshed at the frequency $F_{scram}=1/4F_{scan}$, thus the segments connections change every 4 system clock cycles.
We apply plain-texts PT1 and PT2 as previously done. Comparing both A-2 and B-2 we should observe a difference at position 16. In fact we observe 6 differences. Moreover these differences occur at position 18-19-20-21-85 and 133.
Since the signal *Unpredict_data* is aleatory, we perform the attack again (*Unredict_data* is thus different), comparing pattern 1 and pattern 2this time we observe 6 differences at position 6-23-95-121-124-133.
Thus, even if the attack with the same plain-text is performed several times, it is hardly possible to correlate the bit-streams together in order to retrieve the correct bit difference.

4.3 Design costs

Of course such a protection has a cost in terms of area and power consumption. Using Synopsys tools such as PrimePower and Design compiler [Syn], we have quantified the increase for these parameters. Of course, the cost depends on the number of the segments and the frequency for which the connexions are refreshed. In this case, considering a two segment implementation, the scan dedicated circuitry area is increased by 15%, which corresponds of an increase of 1.8% of the area of the des module. Concerning the power consumption, we compare the power consumption during a encryption when the scrambling is on and when it is off. The power consumption of the des chip during the encryption of a plaintext is increased by less than 1%. This results does not take into account the activity of the random number generator.

5. Conclusions

Inserting scan into a secure design implies new approaches of the method. It has been shown that security must be taken into consideration both at the protocol level requiring a modification of the test controller and at the scan chain level, with modification of the scan chain implementation. Eventually, applying ones of these countermeasure has also proven that at an acceptable cost, scan and security can live together.

References

[Yan04] B. Yang, K. Wu and R. Karri, Scan Based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard, International Test Conference, pp.339-344, 2004

[Yan05] B. Yang, K. Wu and R. Karri, Secure Scan: A Design-for-Test Architecture for Crypto Chips, Design Automation Conference, pp 135-140, 2005

[Hel04] D. Hély, F. Bancel, ML Flottes, B. Rouzeyre, M. Renovell and N. Bérard, Scan Design and Secure Chip, IEEE International On-Line Testing Symposium pp.219-226, 2004

[Hel05] D. Hély, F. Bancel, ML Flottes, B. Rouzeyre, Test Control for Secure Scan Designs, European Test Symposium, pp - , 2005

[Syn] Synopsys, Design Compiler Data Sheet 2003, http://www.synopsys.com/products/logic/dc_expert_ds.pdf