Challenge-response based secure test wrapper for testing cryptographic circuits

Amitabh Das, Miroslav Knežević, Stefaan Seys, Ingrid Verbauwhede
Department of Electrical Engineering ESAT/SCD-COSIC
Katholieke Universiteit Leuven
Leuven-Heverlee, Belgium
Amitabh.Das@esat.kuleuven.be

Abstract—Cryptographic circuits need a special test infrastructure due to security constraints. Typical Design for Testability (DFT) methods, such as scan chains, as applied to most ASICs can not be applied directly to cryptographic chips. These methods, though providing the highest testability, open backdoors or side-channels for attackers to extract secret keys or Intellectual Property information from the core. Past approaches at secure test modified the existing design or on-chip DFT structure and was not suited for System on Chip (SoC) integration testing. This work seeks to address the tradeoff between security, testability and test area overhead by presenting a challenge-response based Secure Test Wrapper structure, suitable for testing IP cores in a SoC environment. This scheme incorporates the KATAN lightweight block-cipher in IEEE 1500 Standard Test Wrapper and as such provides an extremely compact locking and unlocking mechanism for the standard scan chains. The overhead to include this secure mechanism is restricted to about 9% compared to a standard scan and test wrapper. The main contributions of this research are full testability, high security and scalability of the proposed design.

Keywords—Secure Test; SoC integration testing; Crypto-Graphic Circuits; Design for Testability; Scan Chains; Test Wrapper; challenge-response; Test Protocol

I. INTRODUCTION

Testing of cryptographic chips is of primary importance to ensure that they meet the security criteria for which they have been designed. They need the highest testability achievable to ensure that all faults in the circuits are unearthed.

Testability and security are always at odds with each other. There is an inherent tradeoff between the two. If testability is increased, security levels go down, or if security is enhanced in the testing process, then testability suffers. Therefore it is often difficult to provide security and testability in the same design.

Scan chains provide the best testability, but are very insecure, since they can be exploited by an attacker to shift out secret information from the cryptographic cores while in test mode [6, 7]. Built-In-Self-Test (BIST) provides inherent security, as the test response patterns are not available externally. Test patterns are generated internally by a Pseudo Random Pattern Generator (PRPG), and there is internal comparison of the compacted test responses created in a Multiple Input Signature Register (MISR) with a stored golden signature of a fault-free circuit.

However, BIST has relatively high test area overhead since it requires the PRPG, the MISR and the BIST controller. In BIST, the generated internal test patterns are (pseudo) random and cannot be fully controlled. Scan chains, on the other hand, give us the freedom to choose the test vectors. In our work, the flexibility of scan chains has been combined with secure Test Wrappers (which provide both test access and test isolation during scan pattern application) to address this issue of the tradeoff between security and testability. Moreover, the IEEE 1500 Test Wrapper used in our design provides a standard for wrapping multi-vendor cores in a SoC environment.

SoC integrators often use embedded cores which may be procured externally from various Intellectual Property (IP) vendors. The security of the SoC depends on its resistance from attacks exploiting the existing on-chip DFT, targeted at extracting the secret information stored in it. Cryptographic IP cores, such as Advanced Encryption Standard (AES) implementations, must especially be protected from such attacks. Past approaches at secure Design for Testability (DFT) techniques are suitable when the chip with its modified test structure is used individually. When used as an IP module in another SoC chip, they are mostly unusable.

Existing works on secure testing are mostly ad-hoc solutions. They are based on inherently less-secure Linear Feedback Shift Registers (LFSRs) or provide security by obscurity, which is not considered a good secure design practice. This is elaborated in Section II of the paper. To the best of our knowledge, this is the first paper which proposes a secure test method based on cryptographic primitives.

The proposed design is based on the Secure Test Wrapper first introduced in [9]. The scheme had some problems. It was insecure, as it used LFSRs (with secret polynomial) for generating the key to unlock wrapper, and sent the test key in plaintext to the chip for comparison. LFSR length and polynomial was kept secret (security by obscurity) and even non-irreducible polynomials of LFSRs were used to increase resistance to brute-force attacks. It also modifies the standard IEEE 1500 test wrapper boundary cell introducing additional flip-flops, increasing the test area overhead.

The scheme presented here preserves the existing structure of Standard Test Wrapper boundary cell, and is secure (does not use LFSRs or send the key in plaintext). It also uses standard scan chains for thorough testing of the crypto cores. In this manner, the current work seeks to address the tradeoff between security, testability and test area overhead. The design is highly scalable as it does not affect the existing scan-chain DFT present within the SoC, and only adds some modules.
The remainder of the paper is arranged as follows. Section II states the vulnerability of scan chains to active and passive attackers, and then briefly presents the state-of-the-art in secure test methods proposed by various researchers, along with the problems related to their implementation and usefulness for a secure SoC integration environment. An analysis of these schemes, with the parameters of testability, security and area overhead has also been done. The standard IEEE 1500 Test Wrapper Cell is introduced in section III. Then, the proposed secure test wrapper structure along with a description of its constituent modules is presented in detail in section IV. The test protocol is also introduced. In section V, a security analysis will be done. Section VI presents the Key Management problem and the possible solutions through enhancements in the basic test protocol. A comparison with existing secure self-test methods is done in section VII. Implementation and results are discussed in section VIII, and the paper is concluded in section IX.

II. ANALYSIS OF STATE-OF-THE-ART SECURE TESTING APPROACHES

Though scan-chain test approach provides the best controllability to the test engineer, scan chains can be used by an attacker to read chip internal data, stored secret information and determining the position of all the scan elements in a chain. Only a few pins need to be externally controlled and monitored – the scan-in, the scan enable, and the scan output. By observing the scan chain output during a cryptographic process and then by iterating this process, the secret information can be deciphered through knowledge of the encryption algorithm. Also, deliberate faults can be introduced into the scan chain by the attacker to compromise the secret information [7].

These attacks are referred to as scan-based side channel attacks, as the scan chain opens a side channel during test mode, through which secret information inside the chip leaks. There have been successful attacks on hardware implementations of DES and AES, using only the scan chains [6, 7].

Scan chains may be permanently disabled after testing of the chip (by blowing some fuses, for instance) before being used in a product, but then the in-field testability of the chip is lost. Also, testing after a code or firmware upgrade, would not be possible in such a case.

Previous work on secure test is now presented briefly. They need modification to the existing structure of the on-chip DFT and thus are not suitable for a SoC integration scenario.

The first approach is based on randomizing the scan sequence. A LFSR performs a pseudo-random selection of scan chains to be loaded at a time, controlled through a MUXed structure. Instead of serially transmitting the bit stream through the scan registers, the process is randomized. This scheme is also known as scan chain scrambling [1]. This technique, though relatively secure, can be subject to attacks targeting the LFSR using Berlekamp-Massey Algorithms [12]. Also, the area overhead of scan chain scrambler can be high for a complex circuit.

The next method is the ‘Lock and Key Technique’ where the scan chains are divided into a number of sub-chains. The access to these sub-chains is controlled through a Test Security Controller (TSC) which randomizes their operation in case of an unauthorized access by an attacker, making reverse-engineering of the internal structure based on scan responses difficult. An FSM controls the behavior and current mode of the TSC; a test key comparator together with a test controller (TC) is used for returning a secure or insecure result; the LFSR selects a single sub-chain during a scan operation and controls the output multiplexer; and the decoder translates the output of the LFSR into a one-hot enable scheme. Communication between each of the components is kept to a minimum to reduce routing costs and the overall size of the TSC [2]. This method still has high area overhead due to the extensive TSC required in the design. Also, there are security issues associated with the pseudo-random pattern generated by the Linear Feedback Shift Register (LFSR).

Yet another method is the Design for Secure Test (DFST) which is an ad-hoc solution targeting the round structure of AES chips in particular. Since in AES, the rounds consisting of the operations – Substitute Bytes, Shift Rows, Mix Columns and Add Round Key – are identical, they should have identical test responses as well, when fed with the same input test pattern. The responses are then compared in a special comparator for parity [3]. The idea of this scheme though simple, provides an ad-hoc solution only for AES crypto chips which have a regular round structure with some form of unrolling. This method would not work for testing crypto chips with an asymmetric structure.

The approach taken in [4] is to reset the chip and remove all traces of any secret information or cryptographic algorithm execution in test mode. Here access to test features is not prevented, but it is made useless in retrieving secret information. The chip controller of a standard IEEE P1149.1 Boundary Scan is modified such that once the chip is in the test mode, the activation of the test feature (scan in and out operations) is only possible after an initialization process aimed at protecting secret information. Once the chip is in the normal operation mode, no data from the scan operations remain. This provides a full-proof security mechanism, but is not suitable for cryptographic applications where the key or any other secret information needs to be stored on-chip. This can also prevent in-field testability and has test area overhead of 12%.

The ‘Flipped Scan Tree’ architecture [5] which has the advantage of lower test time and hardware overhead has inverters introduced at the scan-in input of some of the Scan D Flip-Flops. The location of the flipped scan flip-flops in the scan tree architecture is known only to the designer and the SoC Tester, and completely unknown to an attacker, who cannot interpret the generated test patterns to seek useful information, without this knowledge. This method though not consuming high test area overhead, has issues with the communication of the test structure between the designer and SoC Tester. Without this knowledge, testing of the chip with this custom test structure is not possible. This can be interpreted as security through obscurity, which may not be considered a good secure design practice. Also, in spite of an unknown test structure, it is possible to decipher it, by observing enough number of test inputs and corresponding responses, using a new kind of differential attack independent of scan chain architecture [18]. The flipped scan tree method has been fixed to a certain extent using the dynamic variable scan approach, through the use of state-dependent scan Flip-Flops, as presented in [19]. However, it still involves security through obscurity by hiding the location of these special scan flip-flops. Moreover, the security depends on the number of replacements of ordinary scan flip-flops with the state-dependent scan flip-flops, and hence involves an area overhead
of up to 16% to ensure higher security. Another approach at addressing the trade-off between security and testability by developing secure scan chains using extended de Bruijn graphs can be found in [24].

Embedded Deterministic Test (EDT) is used in the popular Mentor Graphics Test Compression tool TestKompress. It can provide security through efficient test compression techniques using a Ring Generator architecture. Here, the scan cell data is encoded during pattern generation by solving a series of linear equations, using XOR Compactor with XOR Masking [8]. A theoretical analysis of the security provided by an EDT addition is provided in [20]. However, it does not address the issue of the standard Ring Generator structure known to the attacker. Mentor Graphics, in a recent whitepaper [13] presented the secure features of EDT, claiming it to be suitable for testing of secure chips, if the polynomials used to create the EDT hardware is customized and kept secret. However, the security of this scheme (the underlying linear equations and the structure of the Ring Generators) needs to be further analyzed, before it can be converted into a secure testing strategy.

III. STANDARD IEEE 1500 TEST WRAPPER

IEEE 1500 standard Test Wrapper boundary cell [14], one type of which is shown in Fig. 1, is used to provide both test access and test isolation to the core and the external user-defined logic during testing. The core test input (cti) is the test input to the wrapper cell. It can come from either a primary input (if the cell is the first cell in the wrapper chain) or the cti signal of the previous wrapper cell in the chain. The core test output (cto) is the test output of the wrapper cell. It can drive either a primary output (if the cell is the last cell in the wrapper chain) or the cto signal of the next cell in the wrapper chain. The core functional input (cfi) is fed from the user-defined logic for input wrapper cells. For output wrapper cells, this input is fed from the core. The core functional output (cfo) for input wrapper cells drives the core. For output wrapper cells, this output drives the user-defined logic [17].

The Test Wrappers have three modes of operation. In the INTEST mode, input vectors are applied to the core and the core response observed at the output. In the EXTEST mode, the user-defined logic surrounding the core is tested while the core itself is isolated. The NORMAL mode is the functional mode of operation.

IV. PROPOSED SECURE TEST DESIGN

The proposed secure test architecture, shown in Fig. 2, is based on a challenge - response based test protocol using the KATAN [10] light-weight block-cipher. The latter is a family of flexible hardware-oriented block ciphers suitable for constrained environments, as in cryptographic applications such as smart cards or RFID chips. Our choice of this block-cipher was driven by the fact that it has one of the smallest area footprint for such secure applications, and has many variants.

KATAN runs in software on a secure server. The fixed key should also be securely stored on the server. KATAN receives its plaintext input, via a serial interface, in the form of a random number nonce RN, which is generated by the on-chip True Random Number Generator (TRNG). This output is also transmitted to the on-chip KATAN hardware implementation, fed by the same key, which is loaded from a secure on-chip non-volatile memory location, by an external interrupt, at the start of the testing process.

The ciphertext outputs generated by the software, \(E_K(RN)\), returned through the serial interface, and hardware, \(E_K^*(RN)\), are then evaluated in an on-chip comparator. When they match, only then an ‘Unlock Wrapper’ signal unlocks the wrapper to enable normal testing of the crypto scan core using scan chains, through the wrapper. This Unlock signal is connected to the AND gates between the input and output wrapper boundary registers (IWBR and OWBR) and the AES core scan chains. The test protocol involved in the design is presented below:

<table>
<thead>
<tr>
<th>Software</th>
<th>Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>(E_K^*(RN))</td>
<td>Compute (E_K(RN))</td>
</tr>
<tr>
<td>Possession of key K</td>
<td>Generate Random number RN</td>
</tr>
<tr>
<td>Is (E_K(RN) = E_K^*(RN)) ?</td>
<td>Send Result to Test Wrapper</td>
</tr>
</tbody>
</table>

The above protocol can only be executed by the SoC Integrator in possession of a valid key, the scan chains are otherwise locked against attacks by disabling the scan-in and scan-out.

A brief description of the main building blocks used in the secure test design is now provided.

A. KATAN Lightweight Block Cipher

In KATAN (shown in Fig. 3), the plaintext is loaded in two registers L1 and L2. In each round, several bits are taken from the registers and entered in two nonlinear Boolean functions. The output of the Boolean functions is loaded to the least significant bits of the registers (after they are shifted). This is done in an invertible manner. To ensure sufficient mixing, 254 rounds of the cipher are executed.

An LFSR \(T\) is used instead of a counter, for counting the rounds to stop the encryption after 254 rounds, and to introduce more diffusion as well. As there are 254 rounds, an 8-bit LFSR with a sparse polynomial feedback can be used. The LFSR is initialized with some state, and the cipher has to stop running the moment the LFSR arrives to the predetermined state.

The key schedule of the KATAN cipher loads the 80-bit key into an LFSR \(K\) (the least significant bit of the key is loaded to position 0 of the LFSR). In each round, positions 79 and 78 of the LFSR are generated as the round’s subkey \(k_0\) and \(k_{25+1}\), and the LFSR is clocked twice. The feedback...
A polynomial that was chosen is a primitive polynomial, with minimal hamming weight of 5: \(x^{80} + x^{61} + x^{50} + x^{13} + 1\). In other words, let the key be \(K\), then the subkey of round \(i\) is
\[k_{2i}||k_{2i+1} = k_{i-80} \text{xor} k_{i-61} \text{xor} k_{i-50} \text{xor} k_{i-13}, \text{otherwise} [10].\]

B. True Random Number Generators (TRNGs) using Ring Oscillator

True Random Number Generators (TRNGs), based on the approach by Golic [11], provide a truly random sequence of bits, as compared to pseudo-randomness obtained from Linear Feedback Shift Registers (LFSRs). LFSRs generate a predictable sequence of bits due to their linearity and it is possible to deduce the next sequence once the seed and feedback polynomial are known.

TRNGs, as shown in Fig. 4, are composed of an odd-number of inverters with XOR gates in the forward or feedback paths, for the Fibonacci and Galois Ring Oscillators respectively. To enhance the randomness property, the outputs of the two types of ring oscillators are combined in an XOR gate and stored in registers. The frequency of the generated bits depends both on the number of inverters in the chain as well as the fabrication process technology employed. There should be a mechanism to take the output from the TRNG only after sufficient number of clock cycles have elapsed, as the output of TRNGs is not completely random in the initial period, and randomness increases over time. In the proposed implementation, there is a concurrent process which keeps tab on the count of clock pulses, and only after a predetermined number of clock cycles, the output of the TRNG is fed into an 32 or 64-bit register to be used as the plaintext random number nonce.

V. Security Analysis of the Test Protocol

Our analysis is based on some feasible security assumptions. It is assumed that the attacker does not have access to the Test Server or the key used to run KATAN in software. The attacker cannot probe the ‘Unlock Wrapper’ signal on the chip, otherwise he can just flip the bit (for instance, by pointing laser light) to unlock the wrapper, to enable the scan chains. This can be achieved by laying the signal line in the deep metal layers of the SoC during fabrication to prevent any kind of probing attacks on it. Attacker cannot observe the bit patterns of the on-
chip $E_K(RN)$ signal. Otherwise, he can just send $E_{K^*}(RN)$ matching with this sequence, to unlock the wrapper. This can also be achieved by a similar approach as in the last case. The KATAN Block Cipher is assumed to be secure. This can also be taken as a safe assumption, as there are no known successful attacks against this block cipher. The TRNG is also highly secure against fault attacks (through lowering of temperature, for instance) and cannot be forced to generate the same nonce (such as all zeroes).

Some attack scenarios are considered. Chosen plaintext attack is not feasible as the plaintext is a Random Number nonce. Replay attacks are not useful for an attacker, due to the same reason. The attacker can only eavesdrop on the serial communication taking place between the secure chip and the test server. Since the random number nonce is generated on-chip and sent serially to the server, while the ciphertext on this nonce is generated by the server, it makes no sense to replay it back, as each time, a new nonce is generated by the TRNG. Man-in-the-middle attack is feasible only if the attacker has physical access to the chip as well as the communication taking place between the secure test server and the chip.

The probability of mounting successful attacks on the proposed secure test structure is now stated briefly. The probability of collision in a Guessing attack (or Brute Force Attack) is $2^{-64}$, if 64 bits is the block length (KATAN 64) or $2^{-32}$, if 32 bits is the block length (KATAN 32). Attacks on the block cipher have the same probability. The complete security analysis of the KATAN block cipher can be found in [10], and an algebraic and side-channel analysis is presented in [25].

VI. THE KEY MANAGEMENT PROBLEM AND POSSIBLE SOLUTIONS

In order to be able to test, the SoC integrators require the key to unlock the wrapper. This key should be securely transported from the IP core vendors to the SoC integrators.

In a first scenario, the key can be transported over a secure channel that is established between the core vendor and an integrator. This secure channel can be established using off-the-shelf solutions such as VPN’s, SSL, IPSec, etc. SoC integrators can use this channel to send a list of part numbers and the core vendors can respond with the list of unlock keys.

A second possible solution is to embed the unlock key inside the core itself in such a way that only authorized SoC integrators can access it. Embedding the keys can be done using a supporting Public Key Infrastructure (PKI): the core vendor encrypts the unlock key with the public key of the SoC integrator that will be using this particular core. This encrypted unlock key is stored on the SoC. After shipping, the SoC integrator can extract the encrypted unlock key from the SoC and decrypt it using its private key. The protocol can be depicted as follows:

Test Server $\rightarrow$ Part_no $\parallel$ PKe(K) $\rightarrow$ Secure Chip

$\leftarrow$ RN

$E_{K^*}(RN)$

Here, PKe(K) refers to the key K encrypted with the public key of the SoC Integrator, stored on-chip by the IP core vendor. The key K is only accessible by the authorized SoC integrator using their corresponding private key. However, the chip must now store both K and PKe(K). Elliptic Curve Cryptography (ECC) can be used to reduce storage requirements. The chip does not require any of the PKI circuitry for this purpose. It simply stores K and PKe(K) in its non-volatile memory.

VII. COMPARISON WITH SECURE SELF-TEST OF CRYPTOGRAPHIC HARDWARE IMPLEMENTATION APPROACHES

Recent work on secure self-testing for AES hardware implementation focuses on altering the design of the AES cores using hardware redundancy. The approach taken in [21], for instance, which implements BIST on the AES, modifies the structure of the control unit, key generator and AES logic. Similarly, the approach adopted in [22], which provides secure test access and side-channel attack resistance, adds one S-Box for every four in the original AES design (using repetitions in 16 S-Boxes) and works only for parallel AES implementations. Thus, along with area overhead, caused by S-Boxes which consume the largest portion of the area of an AES hardware implementation, the original AES core is altered, and would not be suitable for a SoC integration scenario. Likewise, the low-area overhead approach taken in [23] for secure testing of public-key crypto-cores, only works for ECC cores containing the Digit Serial Multiplier as the basic building block. Any approach which works on self-test of the crypto-core would therefore modify the existing design or works with specific implementations, and is not suitable for an SoC environment.

The method proposed in this paper is generic and works for all types of crypto hardware implementations. Additionally, it does not alter the crypto-core or any DFT present in the IP module. It is highly suitable for a SoC environment, where the entire cryptographic hardware implementation, along with any DFT is provided as a hard IP module for integration on the SoC. Our approach aims at protecting the crypto DFT infrastructure from scan-based side-channel attacks, and not from power analysis side-channel attacks. The KATAN hardware implementation is assumed to be secure, so side-channel attacks on it is outside the scope of this paper.

VIII. IMPLEMENTATION DETAILS AND RESULTS

Synopsys DFT Compiler Version C – 2009.06-SP3 with 130nm Faraday library, is used for standard scan chain insertion into the AES crypto-core(which is used as a test case), followed by standard IEEE 1500 Test Wrapper Insertion by the same tool. Then, the entire design with the other test modules - TRNG, KATAN hardware implementation, on-chip key memory, comparator and the RS-232 serial interfacing module, is compiled in Synopsys Design Compiler. Multi-Processor Scheduling (MPS) algorithms can be used for the partitioning of the scan chains to achieve reduced test times [15].

To reduce the test area overhead in standard IEEE 1500 Test Wrapper insertion on large cryptographic circuits, the wrapper boundary registers are reused for the implementation of the KATAN Block cipher, the TRNG storage and the other secure test modules. Scan flip flops immediately after (before) a core input have been reused as wrapper input (output) cells, making additional dedicated test wrapper cells unnecessary [15].

FPGA Prototyping has been done to verify the correct operation of the Design. The TRNG has been implemented using Look-Up Tables (LUTs) and Controlled Place constraints in Xilinx ISE and mapped onto a Spartan 3E FPGA (xc3s500e). The randomness property of the sequences generated by the TRNG has been tested using standard NIST and Diehard Randomness Tests.
In our test approach, we have used a RS-232 interface for serial communication between the secure chip and the test server to reduce the test area overhead. However, a JTAG interface with an IEEE 1500 standard Test Access Mechanism (TAM) may also be employed for other suitable applications.

The area comparison for the proposed secure test design, including the serial interface, in Synopsys Design Compiler synthesis is shown in Table I. Implementation of the AES core requires 9317 gates and 263 flip-flops (FFs). The AES core, with 10 and 20 scan chains, requires similar additions in the number of gates. The secure test wrapper on the 10 scan chains results in an area overhead of 9.72% over the Standard IEEE Test Wrapper insertion. The design with 20 scan chains, on the other hand, requires only slightly less gates than the one with 10 scan chains, incurring an area overhead of 8.48% over the standard test wrapper. A custom Perl script can be used for optimized wrapper insertion using the method proposed in [15], instead of the standard insertion by the Synopsys DFT Compiler to further reduce test area overhead.

The scan registers are reused for the construction of the KATAN block cipher, the storage of random number generated by the TRNG, and other modules in the proposed design, to reduce area overhead. To implement KATAN (Fig. 3), we require 120 FFs: 8 FFs for the round count register T, 13 FFs for the shift register L1, 19 FFs for the shift register L2, and 80 FFs for the key register K. Reuse of part of the 263 Scan FFs is possible due to the results described above. The fact that the random number generation and scan test does not occur simultaneously, permits this reuse. Additional multiplexers are required to switch between the normal scan and the random number generation modes.

<table>
<thead>
<tr>
<th>Number of Gates Required</th>
<th>20 Scan Chains</th>
<th>20 Scan Chains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>10 Scan Chains</td>
<td>Original AES Implementation</td>
</tr>
<tr>
<td>AES with Normal Scan</td>
<td>9725</td>
<td>9666</td>
</tr>
<tr>
<td>AES with Scan and Std. Test Wrapper</td>
<td>11415</td>
<td>11309</td>
</tr>
<tr>
<td>AES with Scan &amp; Secure Test Wrapper</td>
<td>12525</td>
<td>12268</td>
</tr>
<tr>
<td>Test Area overhead of Secure Test Wrapper over Std. Test Wrapper</td>
<td>9.72%</td>
<td>8.48%</td>
</tr>
</tbody>
</table>

As seen in the table above, most of the area overhead in the proposed secure test design comes from the test wrapper architecture which is emerging as the leading test standard for future SoC core testing [16]. This test approach is the first of its kind to use a cryptographic primitive to provide both security and testability, preserving the complete scan chain structures.

IX. CONCLUSION

Various secure test schemes have been proposed in past work in this area. However, these methods have a trade-off between testability, security and area overhead, and are not suitable for a SoC environment. In this paper, we have proposed a secure challenge-response based test method, where scan chains have been combined with a custom Test Wrapper structure to achieve high testability and security in a SoC integration scenario. Future challenges are to replace KATAN with KTANTAN [10], for wrapper designs with less reusable flip-flops, to reduce area overhead further and to solve the problem of using non-volatile memory. A functional test will also be designed to test the KATAN implementation itself. Comparison of our secure test method with other similar approaches, such as in [24], is planned as part of future work.

REFERENCES

[18] Ryuta Nara, Nozomu Togawa, Masao Yanagisawa, Tatsuo Ohtsuki, “A Scan-Based Attack Based on Discriminators for AES Cryptosystems”, IEICE 2009.