# A New PUF Based Lock and Key Solution for Secure In-field Testing of Cryptographic Chips

Aijiao Cui, *Member, IEEE*, Chip-Hong Chang, *Fellow, IEEE, Wei Zhou, Yue Zheng*

**Abstract**—Scan-based side-channel attacks have become a new threat to cryptographic chips. Existing countermeasures require a secret test key to unlock the scan chain before in-field testing is allowed. However, test key disclosure poses tremendous risks to multiple crypto chips that share a common test key. We address this open problem of in-field testing by leveraging physical unclonable function (PUF) to make the derived test key unique to each chip. The PUF's response is invoked only once and hardened into a one-time programmable pad. The PUF response required by the designer to derive a test key of each crypto chip can only be recovered at the time of locking the scan chains without directly reading it out. The manufacturer can test the chip normally with no test time penalty before the passed chips are locked. The proposed solution is analyzed to be secure against all known scan-based side-channel attacks and the overhead incurred for the added security is negligibly small.

**Index Terms**— scan chain; scan-based side-channel attack; lock and key; golden key; PUF

## 1 Introduction

DUE to manufacturing process variability, faults are inevitable in integrated circuit (IC) fabrication [1]. The faults in IC can mix with the system intrinsic faults [2]-[4] to make fault detection more complicated and costly [5]-[7]. By adding testability features at design time, scan-based design makes test pattern generation, application and evaluation cost-effective to screen out defective ICs. Full-scan design provides total controllability and observability of the core under test (CUT), with high fault coverage of structural defects. Its test time can be reduced dramatically with modern scan architectures such as multiple scan chains [8].

Due to the unrestrictive and ease of access to the internal states of the CUT, scan chains have also become an exploitable side channel for attacker to steal sensitive information such as cipher key of cryptographic core [9] through in-field testing [10]. Scan-based attacks have been reported to crack on-chip implementation of private key algorithms like data encryption standard (DES) [10] and advanced encryption standard (AES) [11], public-key cryptosystems based on elliptic curve cryptography (ECC) [12] and Rivest-Shamir-Adleman (RSA) [13], and Trivium stream ciphers [14]. To thwart scan-based side-channel attacks, fuses are added to the scan architecture and blown off after manufacturing test to disable their scan testability. Separate built-in self-test (BIST) [15], [16] has also been used to limit the controllability and observability of CUT. These countermeasures either completely incapacitate in-field testing or severely limit its fault diagnosis, which may delay repair and replacement of faulty ICs in service, leading to equally undesirable consequence.

To secure in-field testing, modern countermeasures [9]-[14] prevent the exploitation of scan chain as a side-channel by inserting a logic 'lock' around the scan cells. Attacker who does not have a correct test key observes only erratic data from the scan dump. The "lock and key" schemes introduce several new and unaddressed security issues too. As the foundry is assumed to be entrusted to unlock the fabricated chips for production test, there is a high risk of test key leakage by merchant foundry or contracted test engineer. If the test key is shared among the same design in different chips, it becomes a very attractive high reward-effort ratio target for attack. An attacker needs only to succeed in attacking one chip to gain access to the scan chains of all other crypto chips. If the scan chain of each chip is to be unlocked by a different hardcoded key [18], [20], [22], then a different hardware 'lock and key' will have to be implemented for each design with the same functionality, which increases the mask cost. Programming a soft key into an on-chip nonvolatile memory (NVM) allows a different test key for each chip but the test key is vulnerable to recovery and corruption by memory dump, data remanence and bumping attacks.

Silicon physical unclonable function (PUF) [5] has been widely investigated as a security primitive for chip identity

- *Aijiao Cui is with the School of Electronic and Information Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen, 518055. E-mail: cuiaj@hit.edu.cn*
- *Chip-Hong Chang is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. E-mail: echchang@ntu.edu.sg*
- *Wei Zhou is with the School of Electronic and Information Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen, 518055. E-mail: zhou_9588@163.com*
- *Yue Zheng is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. E-mail: yzheng015@e.ntu.edu.sg*

and nonce generation to avoid storing the secret in persistent memory. PUFs transform the variations in device intrinsic attributes, such as length, width and oxide thickness of transistor and dopant concentration, arising from fundamental physical limitations of IC manufacturing process, into variations in circuit-level parameters such as delay. Such transformation can typically be used to construct a set of unique and random challenge-response pairs (CRPs) for chip identification and authentication. As the responses of PUF are generated based on random disorder of nanoscale device fabrication, it is prohibitively difficult to refabricate another chip to produce the same CRPs exactly even if the entire structure is known down to the transistor-level. Ideally, the response bits generated by each PUF instance to the same challenges should match perfectly to those measured during enrollment. In practice, some response bits may flip due to variations of environmental conditions or aging of transistors. Although majority voting or error correction code can be used to reconcile the unstable response, the area and latency overheads incurred are non-trivial and dependent on the PUF structure.

In this paper, we use a PUF [23] as a secure and tamper-resistant device ID to design a scan-lock that can only be activated by the designer. The PUF response is obtained from the random start-up value of D flip-flops [26]. Our proposed scan-lock scheme has high tolerance to response bit error as the PUF response is generated only once and programmed into a one-time pad. The scan lock is designed such that only its designer can decode the test key from an obfuscated CUT response to a selected test vector without the need to read out the PUF response directly. The uniqueness of PUF response guarantees that each test key is valid only for the scan lock of one chip even though every chip shares a common scan protection circuit. Hence the incentive to invest huge resource to break the test key applicable only to one chip is greatly reduced. Manufacturing test of crypto core can still be performed efficiently through the unlocked scan chain while in-field testing can only be performed by an authorized user with a valid test key after the scan chain has been locked by the designer. A wrong test key will shut down the controllability and observability backdoor for any forms of scan-based attack.

The rest of this paper is organized as follows. Section II reviews existing lock and key schemes for secure in-field testing. Section III presents an overview of the proposed scheme and introduces its key components for in-field test protection. The controller design to facilitate three different phases of operation is elaborated in Section IV. In Section V, security analysis and overheads are evaluated and compared with other existing lock and key schemes. Finally, this paper is concluded in Section VI.

## 2 Related Work

A locking mechanism was implemented in [17] with a secure boundary scan protocol. Two additional instructions are included into the IEEE Std 1149.1 boundary-scan instruction set and supporting hardware is added in the IEEE Std 1149.1 architecture. If the input test key does not match the programmed key, boundary scan is disabled and all the test instructions are bypassed. However, as the soft key is stored in on-chip memory, it is difficult to protect against unprivileged user access. A secure test wrapper (STW) was proposed in [18]. The STW is locked by default to prohibit any test access to the internal scan chains and primary inputs/outputs of the CUT. The STW is enabled by applying a valid test key. The golden key is generated by a linear feedback shift register (LFSR), which shares the same flip-flops with the wrapper boundary cells. As all chips of the same design share a common golden key, when the test key is leaked by a privileged user, all similarly protected chips, including those owned by other users, become vulnerable. Because of the common test key, it is also not possible to denounce the culprits of the break-in.

Existing scan-based attack scenarios assume that the attacker has no knowledge about the detailed scan chain structure. To deduce the cipher key from the scan dumps, the attacker needs to identify the correspondence between encryption registers and scan cells [10]-[12]. Countermeasures are hence proposed to barricade the successful identification of such correspondence by obfuscating the scan chain order [19], [22]. It has been proved, however, that signature attacks [24] do not require full knowledge of scan order but a complete state of the scan chain. Therefore, obfuscation of scan chain order [19] is still vulnerable. A test security controller and an LFSR were introduced in [21] to manipulate the order of multiple scan sub-chains. A valid test key that matches the key stored in a secure register must be entered to correctly seed the LFSR before testing in secure mode can be performed with a known order of sub-chains. This scheme [21] relies on the insecure mode to obscure the complete state of all sub-chains but the measure to safeguard the golden key was not discussed, making it a target of similar threat scenarios as those of [17].

Using the historic state of CUT to obfuscate the scan data has also been proposed in [25]. The recovery of original output response from the obfuscated test data is a concern without a clear elaboration on how the obfuscation was controlled by the historic state. In [20], a shift register is introduced to control the working mode of selected scan cells. When an incorrect test key is input in the test mode, the scan chains output erratic data to deceive the attacker. Besides the same issue of hardcoded test key as [18], it has been demonstrated that specific signature attack [22] can successively identify the misconfigured scan cells to flip one erroneous bit a time until all bits of the test key are recovered. To defend against this powerful test-mode-only signature attack (TMOSA), a more robust dynamic obfuscation of scan data was also proposed in [22]. When an incorrect test key is detected, the test key itself also permutates cyclicly to vary the scan control temporally. In [30], a dynamically obfuscated scan (DOS) architecture is used to

resist noninvasive scan-based attacks by generating a protected obfuscation key to perturb test patterns/responses. The obfuscation key is originated from the output of an LFSR. The LSFR is seeded by a control vector through nonvolatile directly memory access (NDMA). As discussed in Section 1, NVM is vulnerable to recovery and corruption by memory dump, data remanence and bumping attacks.

On-chip decryption and encryption modules are introduced in [31] to protect the scan chain content. The key for data encryption is embedded in the circuit and assumed to be secure. Such scheme inflicts nontrivial test time penalty. In contrast, the authentication keys in [32] are generated by an off chip simulated LFSR. They are embedded using dummy flip flops in the scan chains or don't cares of the test patterns, and verified by an on chip LFSR with the same seed used in the previous key generation stage. The lock and key scheme in [33] uses the test stimulus itself to embed an $N$-bit key. The scan-out is enabled only when $M$ different $N$-bit keys are matched during the test initiation process. This scheme [33] also uses the same hardcoded key to unlock the scan design of all chips fabricated from the same mask set. None of the lock and key schemes addresses the repudiation problem of shared test key. It is near to impossible to testify any privileged users against the liability for the loss or leak of test key to unauthorized users.

## 3 Universal Lock with Unique Test Key Per Instance

A common test key is an alluring target for scan-based side-channel attacks since the same test key can be applied to all chips once it is broken. On the other hand, having a different test key for each chip makes production test by manufacturer costly and inefficient. To address this dilemma, a universal lock with unique test key per instance of a design is proposed to enable the designer to lock every good chip that has passed manufacturing tests distinctively before they are sold to the market. The authorized manufacturer can still perform mass production test of fabricated chips as usual and untrusted merchant foundry should not be able to retrieve the test key from an unactivated lock. As only the authorized buyers or users can unlock the scan chains of individual chips after mass production test, chips that are sold without their scan chains locked are hence unauthorized and vulnerable. It becomes therefore feasible to track risky deployment of unauthorized cryptographic chips.

### 3.1 Overview of the Proposed Scheme

The principle of operation and processes involved in this new lock and key scheme are illustrated in Fig. 1. The scan lock and its key controller are added after scan insertion. Then, the GDS II files and the test vectors of the design are passed to the authorized foundry for chip fabrication and test. The scan design is initially unlocked and can be tested normally after the dies have been packaged. Upon receiving the good chips after mass production test, the designer will activate the lightweight physical unclonable function (PUF) [26] built around some DFFs, which are chained to double as a key register (KR). A one-time programmable (OTP) is used to harden the PUF response. The obfuscation logic of selected scan cells is controlled by the permuted OTP output and the KR content. The scan data is initially unobfuscated until the default OTP states are altered by programming the PUF response bits into the OTP bit-cells with a specific mask activation vector selected by the designer. Once locked, the data scanned into and out of the scan chains will be dynamically obfuscated if a valid test key is not applied before the scan test. Such obfuscation confuses and disrupts the adversary, deceiving him into wasting time and cost on unsuccessful cracking of cipher key from the incorrect intermediate encryption results retrieved from the scan chain. The scan lock is designed in such a way that a valid chip-unique test key to unlock the scan chains can be computed from the PUF response. The PUF response cannot be read externally but derived once upon manufacturing by comparing the obfuscated output response with the known valid response when a specific test vector is applied to the scan design. Only the designer can deduce the PUF response as the obfuscation logic of scan cells is designed according to the specific test vector selected by the designer at design time. Thus, only the designer can deliver a valid test key of each chip to the legal user for in-field testing. The same test key cannot be used to unlock illegally obtained overproduced chips. As such, a non-repudiable binding can be easily established between the chip designer and the tester to deter leakage of test key to unauthorized users.

The locking mechanism is implemented as shown in Fig. 2. The main components include a controller, scan lock, mask generator $\Lambda$, an $n$-bit OTP, $n$ DFFs and a hash module.

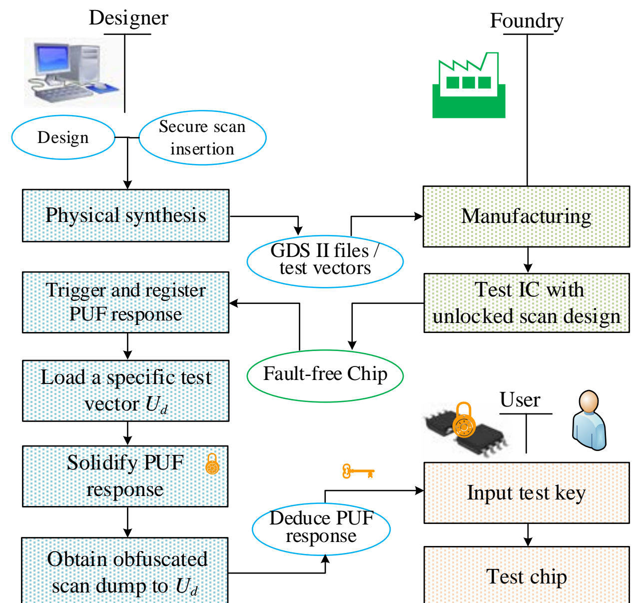



Fig. 1. Overview of proposed secure in-field scan test scheme.

### 3.2 Obfuscation of Scan Data

The scan data obfuscation logic is designed to achieve two goals. First, without a valid test key, the test data scanned into and out of the scan chain will be incorrect. Secondly,

only the chip designer is able to deduce the test key from the obfuscated test response upon activating the PUF to lock the chip.

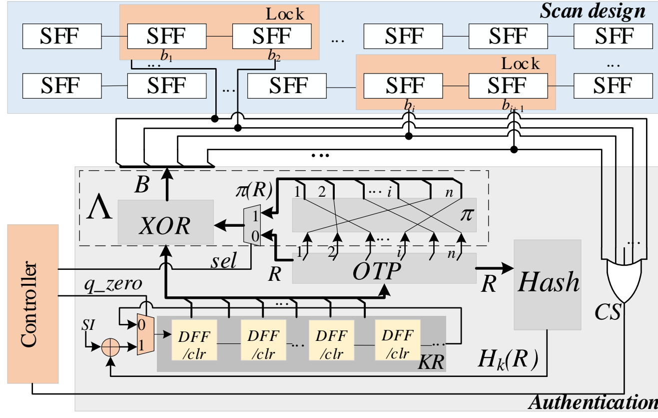
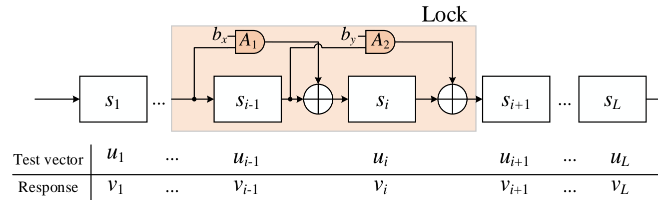
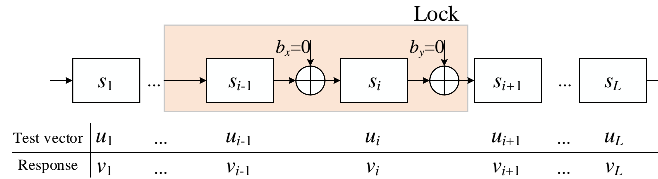



Fig. 2. Components of proposed universal lock with unique test key per chip for securing in-field testing of cryptographic chip.

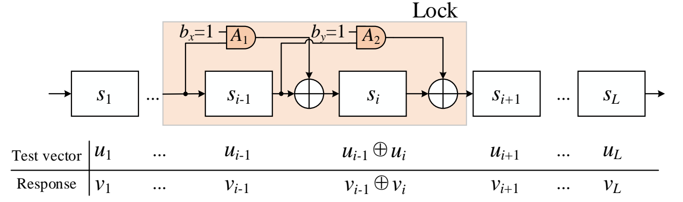
The obfuscation scheme is applicable to full, partial and parallel scan designs. For simplicity and without loss of generality, $m$ scan sub-chains are assumed, and each sub-chain $c_j$, $j = 1, 2, \dots, m$ has approximately equal length, i.e., $|c_j| \approx L \; \forall j$. Each scan cell in $c_j$ is denoted by $s_i$ for $i = 1, 2, \dots, L$. To obfuscate the data scanned into and out of a sub-chain $c_j$, two XOR gates and two AND gates are inserted into a pair of adjacent scan cells, $s_{i-1}$ and $s_i$, as shown in Fig. 3(a). One XOR gate is placed between $s_{i-1}$ and $s_i$ so that the output of $s_{i-1}$ that feeds an (horizontal) input of the XOR gate will be passed either directly to the input of $s_i$ or inverted before it is input to $s_i$, depending on the other (vertical) input of the XOR gate. A masked bit $b_x$ is used to gate the output of $s_{i-2}$ into the vertical input of the XOR gate to control if the data in $s_{i-1}$ is passed unchanged or obfuscated into $s_i$. Likewise, the other XOR gate is placed between $s_i$ and $s_{i+1}$ so that the output of $s_i$ is either passed directly or inverted into $s_{i+1}$ under the control of another masked bit $b_y$ and the output of $s_{i-1}$.



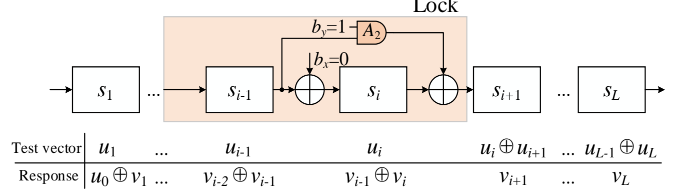

(a). A logic lock on scan path.



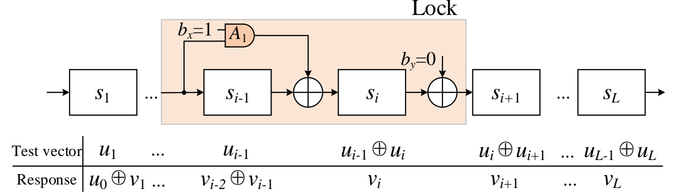

(b) Case 1: $b_xb_y$=00.




(c) Case 2: $b_xb_y$=11.




(d) Case 3: $b_xb_y$=01.




(e) Case 4: $b_xb_y$=10.

Fig. 3. Scan input and output data for different assignments of $b_xb_y$ for a logic lock on scan path.

Let $s_{i-1}$, $s_i$ and $s_{i+1}$ be the three adjacent cells that sandwiches the two XOR gates in $c_j$, and let $U = u_1u_2\dots u_L$ be an arbitrary test vector to be scanned into $c_j$. For an unmodified scan chain, $u_L$ will be scanned into $s_1$ at the first clock cycle. At the second clock cycle, $u_L$ will be shifted into $s_2$ and $u_{L-1}$ will be scanned into $s_1$. In general, $u_{i+L-t}$ will be registered into $s_i$ at clock cycle $t$, $\forall i, t \in [1, L]$. The outcomes of scanning the test vector into $c_j$ for the four scenarios with different combinations of $b_x$ and $b_y$ are analyzed as follows. When $b_xb_y$ = "00", as illustrated by Fig. 3(b), the scan chain functions normally as both XOR gates behave like a non-inverting buffer. When $b_xb_y$ = "11", as shown in Fig. 3(c), only the test data that moves through $s_i$ will be obfuscated. This is because the data scanned into $s_i$ at cycle $t \geq i$ is $u_{i-2+L-(t-1)} \oplus u_{i-1+L-(t-1)} = u_{i-1+L-t} \oplus u_{i+L-t} \neq u_{i+L-t}$ unless $u_{i-1+L-t} = 0$. Since the data present in $s_i$ at cycle $t$–1 is $u_{i-1+L-(t-1)} \oplus u_{i+L-(t-1)}$ instead of $u_{i+L-(t-1)}$, the data scanned into $s_{i+1}$ is $u_{i-1+L-(t-1)} \oplus (u_{i-1+L-(t-1)} \oplus u_{i+L-(t-1)}) = u_{i+1+L-t}$, which is the same as the test data that will be scanned into this cell if $c_j$ is unmodified. Hence, all subsequent scan cells will also receive the correct input test data $u_{i+L-t}$ for $t \geq i$. When $b_xb_y$ = "01", as shown in Fig. 3(d), all cells before $s_{i+1}$ will receive the correct data but the data scanned into $s_{i+1}$ for $t \geq i+1$ will be $u_{i-1+L-(t-1)} \oplus u_{i+L-(t-1)} = u_{i+L-t} \oplus u_{i+1+L-t} \neq u_{i+1+L-t}$ unless $u_{i+L-t} = 0$. It can be shown that the data scanned into all subsequent cells $s_k$ $\forall k \geq i+1$ and $t \geq i+1$ will also be obfuscated to $u_{k-1+L-t} \oplus u_{k+L-t}$. Similarly, when $b_xb_y$ = "10", as shown in Fig. 3(e), all cells before $s_i$ will receive the correct data but the data scanned into $s_i$ will be $u_{i-1+L-t} \oplus u_{i+L-t} \neq u_{i+L-t}$ for $t \geq i$ unless $u_{i-1+L-t} = 0$. Unlike the case of $b_xb_y$ = "11", the data received by all subsequent cells $s_k$ $\forall k \geq i$ will also be

obfuscated to $u_{k-1+L-t} \oplus u_{k+L-t}$. By setting $t = L$, the test data appears in each cell after the entire test vector $U$ has been scanned into $c_j$ is shown below each subfigure for each of the above cases. This possibly obfuscated vector $\hat{U}$ is then applied to the CUT and its response $\hat{V}$ is captured back into $s_1s_2\ldots s_L$ of $c_j$. As $v_L$ will be scanned out first followed by $v_{L-1}$, the response bits $v_{i+1}$ to $v_L$ that do not shift through $s_{i-1}$ and $s_i$ will be scanned out unchanged. The other response bits can be analyzed as above when they are shifted through the obfuscated cells. After $L$ clock cycles, the final response vector $\hat{v}_{out}$ that is fully scanned out of $c_j$ for each of the four scenarios is shown below the respective subfigures in Fig. 3, where the first bit scanned out of $c_j$ is listed as the least significant bit of $\hat{v}_{out}$, and $v_0 = u'_L$ refers to the first bit of the next test vector $U'$ that is scanning into $c_j$ as $\hat{V}$ is scanning out.

The same analysis applies if more than one pair of adjacent cells in $c_j$ are modified. For each pair of masked bits, the test and response vectors scanned through a sub-chain can be analyzed in fragment encompassing only the pair of obfuscated cells $s_i$ and $s_{i+1}$ in consideration. Let $p \in [2, i-2]$ denotes the index of the nearest obfuscated scan cell before $s_i$ and $q \in [i+2, L-2]$ denotes the index of the nearest obfuscated scan cell after $s_{i+1}$. Then the corresponding test vector scanned into the fragment and the scan output of the corresponding response captured into the fragment can be obtained by replacing the indexes 1 and $L$ of $c_j$, $\hat{U}$ and $\hat{V}$ in Fig. 3 by $p$ and $q$, respectively, and replacing $u_0$ by $v_{p-1}$.

It is evident from the above analysis that Case 1 and Case 2 in Fig. 3 can be distinguished if $v_i$ is different from $v_{i-1} \oplus v_i$ in $\hat{V}$, which implies that $v_{i-1}$ has to be '1'. If $v_{i-1}$ = '1', both Case 1 and Case 4 can also be distinguished from Case 2 and Case 3. To distinguish between Case 2 and Case 3 from the scan output, at least one of the bits from $v_{p-1}$ to $v_{i-2}$ should be '1'. Thus, all four cases can be discriminated to identify the input combination of $b_x$ and $b_y$ if the following two conditions are satisfied: (1) $v_{i-1}$ = '1' and (2) at least one of the bits from $v_{p-1}$ to $v_{i-2}$ is '1'.

The designer can select a specific test vector $U_d$. From its output response $V_d$, he can then identify $n/2$ pairs of adjacent scan cells among the sub-chains to insert the obfuscation logic $\psi$ to satisfy the above two criteria. This way he can first apply $U_d$ with $B = 0$ to capture the CUT response $V_d$ into the scan chains. Then, the scan chains are obfuscated by applying a nonzero random mask $B = \hat{B}$ to scan out $V_d$. Only the designer who knows the placement of all obfuscated cells can easily recover the unknown mask $\hat{B}$ by comparing the obfuscated response $\hat{V}_d$ collected at $SO$ with the known $V_d$. The probability of finding $\hat{B}$ from the obfuscated output response of the CUT after passing through all the $n/2$ obfuscated cells to any other test vectors that satisfy the abovementioned criteria by coincidence is $2^{-n/2}$.

### 3.3 Mask Generation and Test Key Derivation

To ensure that the scan chains of each chip is uniquely "locked" by having different random obfuscated responses to the same test vector, the $n$ masked bits ($B = b_1b_2\ldots b_n$) for the aforementioned obfuscation logic are determined indirectly by the response bits ($R = r_1r_2\ldots r_n$) of a PUF.

In principle, any PUF with reasonably good uniqueness can be considered for the response generation. To save hardware cost, the DFFs of the KR that are used to hold the input test key can be reused for the implementation of PUF. The simple PUF design in [26] makes an ideal choice for this purpose, where the power-up characteristic of uninitialized D flip-flops is leveraged to generate a random chip-unique response. It has been shown in [26] that enough randomness exists in such DFF elements when they are implemented on an Application-Specific Integrated Circuit (ASIC). As a weak PUF [26], it is immune to machine learning attacks. Alternatively, the lightweight SCAN-PUF [27] that is designed based on random power-up state of scannable flip-flop can also be used to generate the mask $B$.

As no native silicon PUF can extricate from occasional flipping of one or more response bits due to change in operating environment or device aging, expensive fuzzy extractor is usually required to reconcile the unstable response. To completely evade response reproducibility problem and keep the PUF response confidential, the PUF response is generated only once under the control of the designer and refrained from reading out externally. Highly secure OTP NVM [28] can be used to harden the PUF response without the need for expensive error correction hardware.

A mapping $\Lambda$ is used to permutate the programmed state $Y$ of the OTP and selectively invert the bits at the nonzero bit positions of the KR output $Z$ such that the scan mask $B = \Lambda(Y, Z)$. Before the OTP is programmed, $Y = 0$ and $B = Z$. Once the OTP has been programmed, $B = \Lambda(R, Z) \neq 0$. To unlock the scan chain, a nonzero integer $Z_u$ must be loaded into the KR to make $B = \Lambda(R, Z_u) = 0$. Our design ensures that $Z_u$ can only be derived by the designer without having to read out the OTP state or the PUF response, but not the attacker who has no knowledge about the correct output of the CUT to the test vector $U_d$ selected by the designer and the obfuscation logic $\psi$ that satisfies the two criteria imposed by $U_d$. To avoid fully relying on the obscurity of OTP bit-cell states [28] for security, the hardened PUF response is post-processed by a low-cost keyed hash module [29]. A keyed hash $H_k: X \rightarrow Y$ maps a finite set $X$ to another finite set $Y$ such that it is difficult to find a collision pair $(x_1, x_2)$ for $H_k$, i.e., $x_1 \neq x_2 \in X$ satisfying $H_k(x_1) = H_k(x_2)$ with high probability for an arbitrary key $k$. $Z_u = F\left(H_{K_h}(R), K_t\right)$ becomes a function of the randomized PUF response $R$ and the test key $K_t$, where $F$ is an invertible function. The authorized user is assigned a key pair $(K_h, K_t)$ by the designer, where $K_h$ and $K_t$ are to be input to the hash module and $F$, respectively to make $Z = Z_u$ for unlocking

the scan chain. With the knowledge of $\psi$, the designer is able to recover $R$ by comparing $\hat{V}_d$ with the known $V_d$ to his selected $U_d$ when the OTP is programmed according to the procedure described in Section III.B. Knowing $K_h$, he can easily compute $H_{K_h}(R)$. With the knowledge of $\Lambda$, he will be able to recover the nonzero integer $Z_u$ to make $B$ = 0 and derive a test key $K_t = F^{-1}\left(H_{K_h}(R), Z_u\right)$. Due to the collision resistance of keyed hash function, the attacker who has no knowledge of $K_h$ is unable to compute $H_{K_h}(R)$ and hence $K_t$ to unmask $B$, even if he could successfully hack the OTP to retrieve $R$. Since $K_t$ is input serially through *SI*, *F* can be realized as a modulo-two sum with an XOR gate.

## 4 Controller Design

The OTP remains unprogrammed upon chip fabrication. Hence, the merchant foundry can carry out manufacturing test on the unlocked chips as usual. Since the test key is not even born at this stage, no secret can be leaked. The OTP of each good chip after production test will then be programmed by the designer following the procedure presented in Section 3. It is at this stage that the scan chain is locked and a valid test key $K_t$ is derived by the designer to pair with a selected $K_h$. The scan-locked chip is then sold to the customer who is authorized to test the security crypto core with the key pair ($K_h$, $K_t$) assigned by the designer. The OTP should be unprogrammed before production test and the chip sold in the market or used in the field should have the scan chain locked by a programmed OTP. Defective, compromised and gray market chips can be easily identified if these conditions are not met. The legit stakeholder who has the privilege to perform the right operation in each phase can also be identified by the state of the OTP and the key input through *SI*. A controller can thus be designed to facilitate (prohibit) these operations at different phases by the privileged (unauthorized) stakeholders.

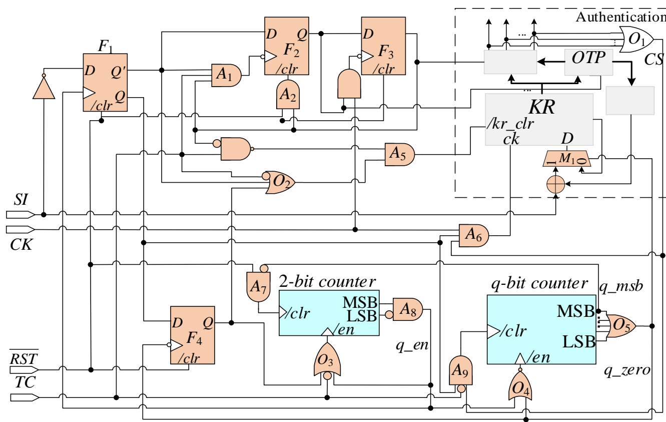



Fig. 5. The test controller for the proposed scheme.

The controller is shown in Fig. 5. It consists of four DFFs, $F_1$ to $F_4$, two counters with synchronous clear input and some logic gates. The output *LS* of $F_1$ is used to disable the clock to KR by the designer while its complementary output *CE* is used to clear the PUF response in the KR, which is generated upon power up when the chip is operated by the manufacturer or the end user other than the designer. $F_2$ and $F_3$ are used by the designer to generate respectively an *EN* signal to enable the programming of OTP and a *sel* signal to temporarily unlock the scan chain for scanning the test vector $U_d$ into the CUT before the PUF response in KR is cleared. $F_4$ disables the 2-bit counter to make multiple counting during in-field testing by the user. The 2-bit counter detects if the chip is working in normal or test mode by counting the number of clock cycles when *TC* = '1'. The *q*-bit counter is used to track the number of test key bits that have been scanned through *SI* by the user. The cyclic probe *CS* is used to configure the KR as a circular shifter when $B \neq 0$.

The chip undergoes a chronological order of operations, namely production test by the chip manufacturer, scan lock activation by the designer and in-field test by the chip user or attacker. In what follows, the operations of the controller and the timing diagrams of affected signals for post-manufacturing production test (by chip manufacturer), locking of good chips (by the designer), and testing of locked chip (by authorized and unauthorized users) will be elaborated and discussed.

**Production Test**: Upon powered up, the scan input pin *SI* is kept constant at '0' with *TC* = '1'. After *TC* has stayed high for two clocks, *q_en* = '1'. This makes *CE* = '0' and hence $\overline{kr_clr}$ turns low to clear the power-up state in KR. *CE* maintains low throughout the process. As the OTP is not programmed, the zero state in KR makes $B$ = 0 and *CS* = '0'. Thus, the clock input to KR, *kr_ck* stays at zero state throughout to keep the output *Z* of KR zero at all time. Hence, the scan chains are not locked. To commence scan test, *TC* is pulled down to '0'. The test vector input at *SI* will be serially scanned into the scan chains for normal scan test with the low *q_zero* signal applied to the control input of multiplexer $M_1$. The timing diagrams of various signals involved in this phase are shown in Fig. 6.

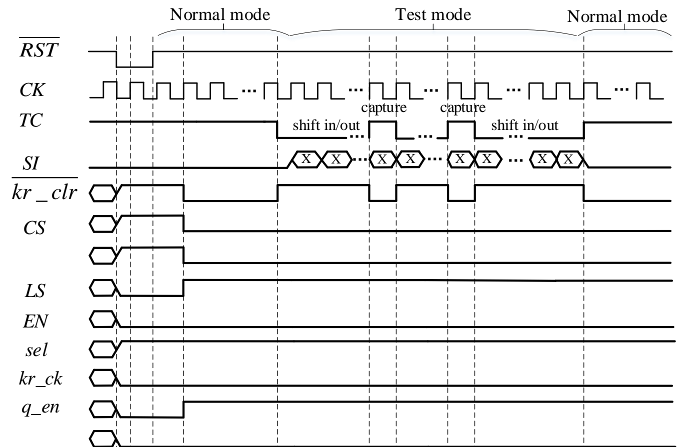



Fig. 6. Timing diagrams of test controller signals during manufacturing test.

**Locking the Scan Chains**: Upon receiving the good chips from the manufacturer, the designer can power up the chip to activate the PUF, program the OTP and retrieve its mask $B$ to work out the test key $K_t$ for each locked chip. The timing diagrams of the operations in this phase for $n$ = 128 are shown in Fig. 7.

Upon reset, the PUF response is generated in KR. Meantime, the 2-bit *Counter* is cleared. *TC* is kept high for two cycles before *q_en* changes from '0' to '1'. *SI* should be asserted high at the rising edge of *q_en* to maintain *CE* = '1' and keep $\overline{kr_clr}$ high. As *LS* = '0', the clock to KR, *kr_ck* is disabled to prevent the PUF response in KR from shifting.

*TC* is then pulled low to put the chip in test mode. $F_3$ generates a high *EN* signal at the falling edge of *TC* to program the PUF response into the OTP. At the falling edge of next clock cycle, the output *sel* of $F_3$ changes from '1' to '0', which clears $F_2$ on the falling edge one cycle later to disable the programming of OTP. The low *sel* selects the OTP output *R* from the multiplier input to XOR with the current content *R* of KR in $\Lambda$. Hence, *B* = 0 and the scan design is unlocked to allow the test vector $U_d$ selected by the designer to be scanned through *SI* into the CUT. The response after $U_d$ has been scanned into the CUT is then captured into the scan chains by setting *TC* = '1'. When *TC* = '1', $\overline{kr_clr}$ = '0', which clears the KR and causes $B = R \neq 0$. The scan design is hence locked. *TC* is then switched back to low so that the obfuscated response $\hat{V}_d$ can be shifted out from the scan design. The designer who knows the actual circuit response $V_d$ to $U_d$ and the positions of masking gates in the scan chains can retrieve the mask *B* and hence the PUF response *R* from $\hat{V}_d$. Then, designer can compute the test key $K_t$ according to the analysis in Section 3.3.

This phase of operation is different from the previous production test in that it is initiated by the designer by inputting bit '1' at *SI* to preserve the PUF response in KR after reset. Others who follow the same procedure to program the OTP will not be able to unlock the scan chains since they do not know the locations of obfuscated scan cells to recover *R* from the obfuscated $\hat{V}_d$ to any test vector $U_d$ they applied in this process.

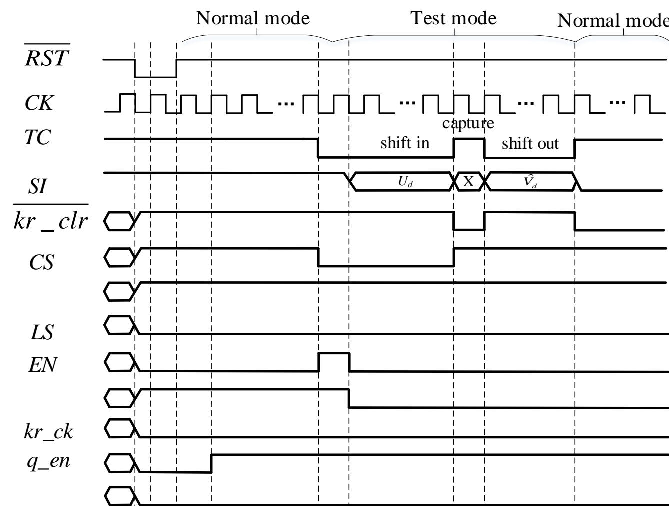



Fig. 7. Timing diagrams of test controller signals during programming of OTP and test key derivation.

**Field Test:** When the scan-locked chip is powered up, as the PUF response generated in KR is different from the permuted OTP output $\pi(R)$, $B \neq 0$. Hence *CS* = '1'. The test key can be applied after *TC* stays high in normal working mode for two cycles, i.e., until *q_en* = '1'. Before applying the test key, *SI* should be fed with '0' to keep *LS* high in order to gate the clock signal into KR. To commence testing after the keyed hash, *TC* is pulled low to scan in $n = 2^q$ bits of test key serially through *SI* and start the *q*_bit counter at the same time. When *q_zero* = '0', the test key has been fully scanned through *SI*. If the test key is correct, the KR will output $\pi(R)$ and *B* = 0. *CS* changes to '0' to disable *kr_ck* in order to hold $\pi(R)$ in the KR. Therefore, scan test can be performed normally through the unlocked scan chains. If the test key is incorrect, the content of KR is different from $\pi(R)$. Hence, $B \neq 0$ and *CS* remains high. As the control input *q_zero* to $M_1$ stays at '0', the active clock *kr_ck* continues to shift the content of KR circularly, causing *B* to change every clock cycle and the scan data to be dynamically obfuscated [22]. The timing diagrams of the related signals are shown in Fig. 8.

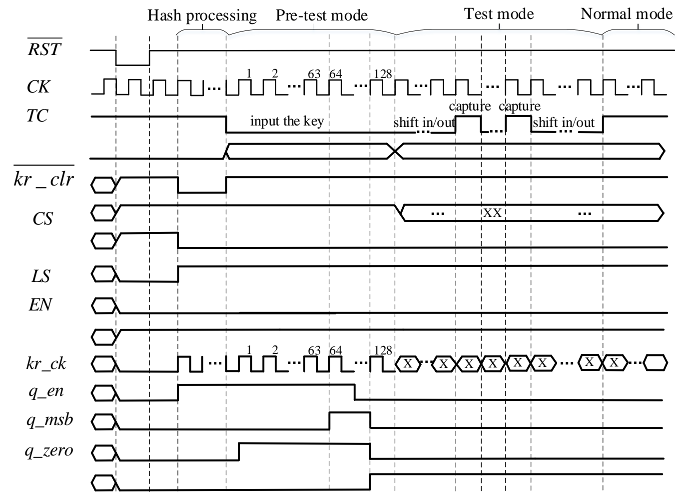



Fig. 8. Timing diagrams of test controller signals during in-field testing.

## 5 Experimental Results and Security Analysis

The testability, security and overhead of the proposed scheme are evaluated in this section.

### 5.1 Testability

Upon fabrication, before the OTP has been programmed, the scan chains can be kept in unlocked state, i.e., *B* = 0 simply by clearing the KR on reset. During production test, the manufacturer only needs to keep *SI* = '0' before *TC* is pulled low to commence the scan test as usual. Although the test vectors of the CUT were generated after scan insertion without the obfuscation logic, they can be applied transparently after the insertion of test protection circuit without compromising the fault coverage or testability achievable by the original test set.

After the good chip has been uniquely 'scan-locked' by the designer, the scan chains can be unlocked once a correct test key has been fully scanned through the *SI* to make the KR output equal to $\pi(R)$. The test patterns can then be correctly scanned into the scan chains and the responses

from the CUT can be captured and scanned out unobfuscated for validation. The specified fault coverage and the original testability are also not affected by the added scan protection circuit.

For most lock and key schemes, faults in the scan protection circuit cannot be tested by the original scan chains. Allowing controllability and observability to the internal nodes of the scan protection circuit will expose the test key and undermine their security. These scan protection circuits are usually tested by a BIST with high fault coverage. This is, however, unnecessary for our scheme. Since the crypto core has already been tested independent of the scan protection circuit upon manufacturing, if there is any fault in the scan protection circuit, the output responses will be deviated from the correct ones when the same scan tests are performed with the correct test key derived by the designer after the OTP has been programmed. This is as good as the simple pass-fail test result indicated by the BIST. Thus, the correctness of the scan protection circuit can also be validated by the designer after he has programmed the OTP to lock the scan chains.

## 5.2 Uniqueness of test key

The uniqueness of a PUF characterizes how distinguishable is the response to the same challenge generated by the circuit on one die from that on another die. Since PUF is used as a main feature to provide a different test key to each identically designed scan chain protection circuit, this is important to minimize the probability of test key collision. The inter-class uniqueness test conducted on the hardware-efficient PUF in [26] reported an inter-class Hamming distance of at least 26% between PUF responses of different devices. Although the inter-class distribution is not ideal due to the one-skew response, this amount of bit differences is sufficient to trigger an avalanche effect from the hash function to derive a highly discriminable test key for each chip. The measured Hamming distance of 177 comparisons between two 16-bit PUF responses in [27] approximated a binomial distribution with a mean of 8.2, which is very close to an ideal mean Hamming distance of 8. Both PUF designs [26], [27] were reported to have low correlation between PUF responses from different chips. It is important to note that the PUF response is used as a nonce in our proposed scheme. Hence its response reliability is inconsequential and modeling attacks are irrelevant.

## 5.3 Security against typical attacks

1) *Brute force attack*

For an attacker without the knowledge of the obfuscation logic and the hashed key, the probability of correctly figuring out the $n$-bit test key to unlock the scan chains by coincidence is $1/2^n$. For $n = 64$ and 128, this probability is as low as 5.42E–20 and 2.94E–39, respectively. Brute force attack by exhaustive enumeration of test key is therefore computationally infeasible. Unlike other lock and key schemes, such effort will have to be repeated for every chip as a successfully cracked test key on one chip will not work on another scan-locked chip. Because each crypto core requires a different test key to unlock its scan chains, our scheme significantly enlarges the gap between (high) efforts and (low) rewards to effectively deter brute force attack.

2) *On-chip data dumping attack*

An unauthorized user who has access to a scan-locked crypto chip may use the scan chains to apply as many possible test vectors to capture the crypto core's responses as an attempt to recover the PUF response $R$ according to Section III.B. As each scan-locked chip has only one valid test key, which can only be decoded at the time of programming the OTP from the obfuscated CUT response to an input vector $U_d$ selected by the designer with the knowledge of the locations of obfuscated gates. Once the OTP has been programmed, it is computationally intractable for the designer to decode $R$ (if he has lost it) by re-entering $U_d$, let alone the attacker who has no knowledge about the obfuscated scan cell locations. In other words, the leak of $U_d$ after the OTP has been programmed poses no threat, as the effect of applying $U_d$ in scan mode is as good as commencing scan test with any other invalid test keys. Hence, the attacker cannot use the scan dumps to retrieve $R$, $\pi(R)$ and the hash key $K_h$ to compute $H_{K_h}(R)$ , which are required for the recovery of the test key.

3) *Attack on illegally obtained unlocked chips*

If any crypto chips are hijacked by the attacker before their OTPs have been programmed, the attacker may try to lock these chips using the same OTP programming process. As discussed above, the attacker can only have one chance to input a mask activation vector $U_d$ of his choice to retrieve $R$ from the obfuscated output response of the CUT. Since the attacker does not know the locations of the masking gates, we conservatively assume that he can find a CUT response to satisfy the second criterion stipulated in Section 3.2. However, without the knowledge of the design, it is almost impossible for him to find a test vector in only one trial to produce an output response that has '1' appearing in all the bit positions corresponding to those $n/2$ specific obfuscated scan cell locations. It is reasonable to assume that the probability of the CUT output response to an arbitrary test vector that satisfies the first criterion of Section 3.2 is $1/2^{n/2}$. When $n = 128$, this probability is as low as 5.42E–20. Assume that the attacker is fortunate enough to find a test vector that will capture a response from the crypto core with as many as $m/2$ ($m < n$) '1's out of these $n/2$ specific positions. If there is no obfuscated scan cell situated between these $m/2$ correctly predicted obfuscated scan cells and the *SO* pin, then he will be able to deduce the values of these masked bits according to the obfuscated output response. For the remaining $(n - m)/2$ bits of the captured response vector that should be '1's but are '0's, the scan data through these obfuscated scan cell positions remain undetermined. Hence, his derived $R$ will not match the output of the programmed OTP. With the permutation $\pi$ and the keyed-hash function $H_k(R)$, the intractability to deduce test key is significantly elevated. As long as there is one mismatched bit in the test key, the scan data will be dynamically obfuscated. Hence, the attacker cannot make an illegally obtained unlocked chip legit by programming its OTP using a mask activation vector of his choice.

4) *Test-mode-only signature attack (TMOSA)*

A TMOSA [22] targets static lock of scan chains by applying a signature attack under the test mode to successively identify the nearest locked scan cell position and retrieve the correct key bit one by one. This domino effect on consecutively resolved obfuscated scan data is prevented by dynamic obfuscation in our scheme. The *CS* bit produced by $O_1$ in Fig. 5 detects any bit difference between the input test key and the correct test key. If there is any mismatch, the input test key will be cyclically shifted to change the masking bits in each cycle during the test mode when the test vectors and the captured responses are scanned into and out of the scan chains. As the valid test key for each chip is uniquely determined by a random PUF response, the test key bits are uncorrelated, and shift invariant test keys like all zeros or all ones will be avoided. Hence, it is virtually impossible to observe the correct states of most, if not all, of the scan cells at *SO* with a partially matched test key.

### 5.4 Overhead analysis

The proposed field test scan lock (FTSL) is implemented on scan designs for pipelined and iterative AES circuits with key scheduling (KS). The suffix *L* of FTSL-*L* is used to denote the bit length of PUF response. The original designs are synthesized by Synopsys Design Compiler before the scan chains are added into each netlist by Synopsys DFT Compiler. The proposed lock and key components are then inserted into the netlists of the scan designs and synthesized by Synopsys Design Compiler. To facilitate comparison, technology independent synthesis results are shown in Table I. The areas of the original circuit before and after scan insertion (abbreviated as '*original*' and '*scan*', respectively) and the scan design protected by FTSL are expressed in terms of the number of equivalent two-input NAND gates. The lengths *L* of the scan chains are 6720 for the pipelined AES and 522 for the iterative AES. The keyed hash function is implemented by $SHA3_{low}$, which is the "Low_Area_Copro" from [29]. It has an area of 4965. The area overheads $\Delta A$ in gate count and percentage area overhead are shown in the last two columns of Table I. $SHA3_{low}$ has a latency of 48 clock cycles. The area and latency can be substantially reduced by using non-keyed hash considering the very low reward/effort incentive for high class attack on FTSL.

TABLE I. SYNTHESIS RESULTS OF PIPELINED AND ITERATIVE AES SCAN DESIGNS WITH AND WITHOUT THE PROPOSED SCAN PROTECTION SCHEME.

| AES | Scheme | Area, *A* (number of gates) | | | $\Delta A$ | $\Delta A$ (%) |
|---|---|---|---|---|---|---|
| | | *original* | *scan* | *secure* | | |
| Pipelined | FTSL-64 | 205934 | 212280 | 218842 | 6562 | 3.09 |
| | FTSL-128 | | | 220356 | 8076 | 3.80 |
| Iterative | FTSL-64 | 25052 | 25512 | 32074 | 6562 | 25.72 |
| | FTSL-128 | | | 33588 | 8076 | 31.66 |

FTSL-128 incurs only 3.8% hardware overhead over the originally synthesized pipelined AES with scan design. If the cryptographic chip has already a built-in keyed hash function core, the overhead can be reduced to 1.46%. Usually the crypto core is monolithically integrated into a larger SoC, the overhead due to FTSL components is negligibly small.

TABLE II. COMPARISON OF LOCK AND KEY SCHEMES

| Design | ΔA (%) | Attacks | Test key | Weaknesses | Testability impact |
|---|---|---|---|---|---|
| SOSD-128 | 0.34 | TMOSA | Shared; Predetermined; Hardcoded; Validate by direct comparison. | Repudiable liability for test key leakage; Test key is made known to manufacturer. | Nil |
| DOSD-128 | 0.47 | None | | | Nil |
| ROS-16c | 0.15 | Signature attack | | | Nil |
| SDSFF100 | 0.25 | None | | | Yes |
| Vim-Scan | 1.54 | Bit-role identification attack[1] | | | Nil |
| SSTKR | 2.61[2] | Bit-role identification attack; Memory attack. | Shared; Predetermined; NVM; Validate by direct comparison. | | Nil |
| TSC-12bits | 2.74 | Signature attack; Memory attack. | | | Nil |
| DOS | 2.01 | Memory attack | | | |
| SIE | 2.92 | Memory attack | | | |
| FTSL-128 | 3.80 | None | Chip-unique; Derived from PUF; OTP and hashed; Compared with permuted OTP o/p. | None | Nil |

The area overhead, security and impact on testability of the proposed secure design are also compared with other lock and key schemes in Table II. The compared scan protection schemes include SOSD-128 [20], DOSD-128 [22], ROS [19], SDSFF [25], Vim-Scan [33], SSTKR [32], TSC [21], DOS [30] and SIE [31]. ROS-16c refers to the ROS design [19] with 16 sub-chains. SDSFF-100 refers to the SDSFF design [25] with 100 state-dependent SFFs, 1054 SFFs and a 5-bit test key. Vim-Scan refers to the Virtually Impervious Scan design [33] with 128-bit matched keys. SSTKR refers to the secure design [32] with area overhead proportional to the size of the key but its area overhead to AES core is not reported in [32]. DOS refers to the dynamically obfuscated scan scheme [30]. SIE refers to the scan interface encryption scheme [31]. TSC-12bits refer to the TSC design [21] with 12-bit LFSRs and a 64-bit test key. The denominator of area overhead is the gate counts reported in Table I for the implementation of the scan design of pipelined AES with KS. The extra gate counts (i.e., $\Delta A$) required for the implementation of other schemes are excerpted directly from the literature. Possible attacks in Table II refer to the known scan-based side-channel attacks that the schemes in comparison are unable to defend against. Different schemes are compared by means of their differences in the

[1] Flipping of the authentication key bit causes change in scan out vectors but not the nonkey bit. This will significantly reduce the difficulty of identifying the key bits.

[2] Design without dummy flip flops. Benchmark: ISCAS'89 (S15850), key size = 10.

generation and safekeeping of test keys, as well as the weaknesses in the way the test keys are preserved. Ideally, the protection scheme should not degrade the test quality or the reliability of the original scan design. Among the schemes in comparison, the testability of the CUT of [25] will be compromised when any fault exists in its protection circuit. As a whole, the proposed scheme can resist all existing known scan-based attacks while incurring low overhead. It is the only scheme that does not use a shared test key and has the least security impact on the loss or leakage of test key, yet normal production test can be performed more efficiently without fear of test key leakage. The simplicity of preserving a common test key for all chips in hardcoded form or in multi-programmable NVM makes the area overhead of other countermeasures lower. However, preserving shared test key in programmable memory or hardcoded permanently on chip is evidently far less secure and lacks non-repudiation than a chip-unique tamper-proof test key. As analyzed in Section 3.2, no one other than the designer who places the obfuscated gates can safely lock the scan chains to deduce its test key for each chip. Even with these added security features, the overhead is kept practically small, though not the minimum. More importantly, only FTSL allows tracking of individual chips. This has a positive side effect of increasing the barrier for counterfeit and malicious ICs to successfully infiltrate into mission-critical systems.

## 6 Conclusion

This paper presents a new solution to the unresolved security issues arising from the use of a common test key in existing lock and key schemes for protecting cryptographic chips against scan-based side-channel attacks. The risk of test key leakage of any protected chip by privileged user is minimized by having a unique key for each cryptographic chip protected by the same scan lock. Test key uniqueness and tamper-resistance are achieved by reusing the KR as PUF. To make PUF response reliability a nonissue, the PUF response is generated once and programmed into an OTP. The PUF response is made inaccessible so that physical measurements cannot be used to attack the PUF. The scheme ensures that the internally generated response can be deduced from the obfuscated scan response only by the designer with a dedicated mask activation vector. The loss of mask activation vector after the hardening of PUF response poses no security risk. As no test key is generated nor stored on chip upon fabrication, production test can be performed efficiently by untrusted merchant manufacturer with no modification to existing test fixture and procedure. Fault coverage is also preserved for in-field testing without compromising resistance against scan-based side-channel attacks. This is more efficient and secure than the work around solution of disabling and substituting scan architecture after manufacturing test by BIST for in-field testing.

## Acknowledgment

This research was supported in part by the National Natural Science Foundation of China under Grant 61672182, the Guangdong Natural Science Foundation under Grant 2016A030313662 and the Shenzhen Overseas High-Level Talent Innovation Foundation under Grant KQJSCX20160226202510, and in part by the Singapore Ministry of Education Tier 1 Grant MOE2018-T1-001-131 (RG87/18). Aijiao Cui is the corresponding author.

## References

[1] L. Chen, A. Cui and C.-H. Chang, "Design of Optimal Scan Tree Based on Compact Test Patterns for Test Time Reduction," *IEEE Trans. Comput.*, vol. 64, no. 12, Feb. 2015, pp. 3417-3492.

[2] X. He, Z. Wang, L. Qin, and D. Zhou, "Active fault-tolerant control for an Internet-based networked three-tank system," *IEEE Trans. Control Systems Technology*, vol. 24, no. 6, Jun. 2016, pp. 2150-2157.

[3] Y. Y. Wang, Y. Sun, C.-F. Chang, and Y. Hu, "Model-based fault detection and fault-tolerant control of SCR urea injection systems," *IEEE Trans. Vehicular Technology*, vol. 65, no. 6, Jun. 2016, pp. 4645-4654.

[4] H. Badihi, Y. Zhang, and H. Hong, "Wind turbine fault diagnosis and fault-tolerant torque load control against actuator faults," *IEEE Trans. Control Systems Technology*, vol. 23, no. 4, Apr. 2015, pp. 1351-1372.

[5] I. Pomeranz, "On the computation of common test data for broadside and skewed-load tests," *IEEE Trans. Comput*, vol. 61, no. 4, 2012, pp. 578-583.

[6] I. Pomeranz, "Multicycle tests with constant primary input vectors for increased fault coverage," *IEEE Trans. CAD Integr. Cir. Syst.*, vol. 31, no. 9, Sept. 2018, pp. 1428-1438.

[7] S. Zhang, K. R. Pattipati, Z. Hu, and X. Wen, "Optimal selection of imperfect tests for fault detection and isolation," *IEEE Trans. Systems, Man, and Cybernetics: Systems*, vol. 43, no. 6, Jun. 2013, pp. 1370-1384.

[8] T. Yu, A. Cui, M. Li, and A. Ivanov, "A new decompressor with ordered parallel scan design for reduction of test data and test time," in *Proc. IEEE Int. Symp. Cir. Syst. (ISCAS)*, Lisbon, Portugal, May 2015, pp. 641-644.

[9] Y. Liu, Y. Jin, A. Nosratinia, and Y. Makris, "Silicon demonstration of hardware trojan design and detection in wireless cryptographic ICs," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 4, 2017, pp. 1506-1519.

[10] Y. Bo, K. Wu, and R. Karri, "Scan-based side-channel attack on dedicated hardware implementations of data encryption standard," in *Proc. Int.Test Conf.*, Washington DC, USA, Oct. 2004, pp. 339-344.

[11] Y. Bo, K. Wu, and R. Karri, "Secure scan: A design-for-test architecture for crypto chips," *IEEE Trans. CAD Integr. Cir. Syst.*, vol. 25, no. 10, Oct. 2006, pp. 2287- 2293.

[12] R. Nara, N. Togawa, M. Yanagisawa, et.al, "Scan-based attack against elliptic curve cryptosystems," in *Proc. Asia South Pacific Des. Autom. Conf. (ASP-DAC)*, Taipei, Taiwan, Jan. 2010, pp. 407- 412.

[13] R. Nara, K. Satoh, M. Yanagisawa, et.al, "Scan-based side-channel attack against RSA cryptosystems using scan signatures," *IEICE Trans. Fundamentals Electron., Comm. Comput. Sci.*, vol. E93-A, no. 12, Dec. 2010, pp. 2481-2489.

[14] M. Fujishiro, M. Yanagisawa, and N. Togawa, "Scan-based attack against Trivium stream cipher independent of scan structure," in *Proc. IEEE 10th Int. Conf. ASIC (ASICON)*, Shenzhen, China, Oct. 2013, pp. 1-4.

[15] F. Opritoiu, A. Bozesan, and M. Vladutiu, "Pseudo random self-test architecture for advanced encryption standard," in *Proc. IEEE 19th Int. Symp. Design Techno. Electronic Packaging*, Galati, Romania pp. 271–276, Oct. 2013.

[16] S. H. Namin, A. Mehta, P. H. Namin, R. Rashidzadeh, and M. Ahmadi, "A secure test solution for sensor nodes containing crypto-cores," in *Proc. 2017 IEEE Int. Symp. Cir. Syst.*, Baltimore, MD, USA, pp. 1-4, May 2017.

[17] F. Novak and A. Biasizzo, "Security extension for IEEE Std 1149.1," *Journal of Electron. Testing*, vol. 22, no. 3, Jun. 2006, pp. 301-303.
[18] G.-M. Chiu and J. C.-M. Li, "A secure test wrapper design against internal and boundary scan attacks for embedded cores," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 1, Jan. 2012, pp. 126-134.
[19] Y. Atobe, S. Youhua, M. Yanagisawa, and N. Togawa, "Secure scan design with dynamically configurable connection," in *Proc. 19th Pacific Rim Inter. Symp. on Dependable Computing (PRDC),* Vancouver, British Columbia, Canada, Dec. 2013, pp. 256-262.
[20] Y. Luo, A. Cui, G. Qu and H. Li, "A new countermeasure against scan-based side-channel attacks," in *Proc. 2016 IEEE Int. Symp. Cir. Syst.*, Montreal, Canada, May 2016, pp. 1722-1725.
[21] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic, "Securing designs against scan-based side-channel attacks," *IEEE Trans. Dependable and Secure Computing*, vol. 4, Dec. 2007, pp. 325-336.
[22] A. Cui, Y. Luo, and C.-H. Chang, "Static and dynamic obfuscation of scan data against scan-based side-channel attacks," *IEEE Trans. Inform. Forensics Security*, vol. 12, no. 2, Feb. 2017, pp. 363-376.
[23] C.-H. Chang, Y. Zheng, and L. Zhang, "A retrospective and a look forward: fifteen years of physical unclonable function advancement," *IEEE Cir. Syst. Magazine*, vol. 17, no. 3, Aug. 2017, pp. 32-62.
[24] A. Cui, Y. Luo, H. Li, and G. Qu, "Why current secure scan design fail and how to fix them? ," *Integration, the VLSI Journal,* vol 56, Jan. 2017, pp. 105-114.
[25] Y. Atobe, S. Youhua, M. Yanagisawa, et.al, "State dependent scan flip-flop with key-based configuration against scan-based side-channel attack on RSA circuit," in *Proc. Asia Pacific Conf. Cir. Syst.*, Dec. 2012, pp. 607 – 610.
[26] V. Leest, G. J. Schrijen, H. Handschuh, and P. Tuyls, "Hardware intrinsic security from D flip-flops," in *Proc. 5th ACM Workshop Scalable Trusted Computing,* Chicago, IL, USA, Oct. 2010, pp. 53-62.
[27] B. Niewenhuis, R. D. Blanton, M. Bhargava, and K. Mai, "SCAN-PUF: A low overhead physically unclonable function from scan chain power-up states," in *Proc. IEEE Int. Test Conf.*, Anaheim, California, USA, Sept. 2013, pp. 1-8.
[28] DesignWare OTP NVM for Secure Encryption Keys, Synopsys https://www.synopsys.com/designware-ip/memories-logic-libraries/non-volatile-memory/one-time-programmable/encryption-keys.html.
[29] Keccak in VHDL, Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche and Ronny Van Keer, https://keccak.team/hardware.html.
[30] X. Wang, D. Zhang, M. He, D. Su and M. Tehranipoor, "Secure Scan and Test Using Obfuscation Throughout Supply Chain," *IEEE Trans. Computer-aided Design of Integrated Circuits and Systems,* vol. 37, no. 9, 2018, pp. 1867-1880.
[31] M. Da Silva et al., "Scan chain encryption for the test, diagnosis and debug of secure circuits," *Proc. 22nd IEEE Test Symp. (ETS),* Limassol, Cyprus, 2017, pp. 1–6.
[32] M. A. Razzaq, V. Singh, and A. Singh, "SSTKR: Secure and testable scan design through test key randomization," in *Proc. IEEE Test Symposium*, 2011, pp. 60–65.
[33] S. Paul, R. S. Chakraborty, and S. Bhunia, "VIm-Scan: A low overhead scan design approach for protection of secret key in scan-based secure chips," VTS, 2007, pp. 455–460.

**Aijiao Cui** (S'05–M'10) received the B.Eng. and M.Eng. degrees in electronics from Beijing Normal University, Beijing, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2009. From July 2003 to December 2004, she was a Lecturer with Beijing Jiaotong University, Beijing. She was a Research Fellow with Peking University Shenzhen SoC Laboratory, Shenzhen, from 2009 to 2010 prior to joining the School of Electronic and Information Engineering of Harbin Institute of Technology (Shenzhen) in 2010, where she is currently an Associate Professor. Her current research interests include hardware security and IC testing techniques.

**Chip-Hong Chang** (S'92-M'98-SM'03-F'18) received the B.Eng. (Hons.) degree from the National University of Singapore in 1989, and the M. Eng. and Ph.D. degrees from the Nanyang Technological University (NTU) of Singapore in 1993 and 1998, respectively. He is an Associate Professor of the School of Electrical and Electronic Engineering (EEE) of NTU. He held joint appointments with the university as Assistant Chair of Alumni from 2008 to 2014, Deputy Director of the Center for High Performance Embedded Systems from 2000 to 2011, and Program Director of the Center for Integrated Circuits and Systems from 2003 to 2009. He has coedited four books, published ten book chapters, around 100 international journal papers (more than two-thirds are IEEE) and more than 170 refereed international conference papers (mostly in IEEE), and delivered over 30 colloquia. His current research interests include hardware security, unconventional number systems, and low-power and fault-tolerant digital signal processing algorithms and architectures. Dr. Chang serves as the Associate Editor of IEEE Transactions on Very Large Scale Integration (VLSI) Systems since 2011, IEEE Access since 2013, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems and IEEE Transactions on Information Forensic and Security since January 2016, IEEE Transactions on Circuits and Systems-I from 2010-2013, Integration, the VLSI Journal from 2013-2015, Springer Journal of Hardware and System Security since June 2016 and Microelectronics Journal since May 2014. He guest edited several journal special issues and served in the organizing and technical program committee for more than 60 international conferences. He is also an IET Fellow and 2018-2019 Distinguished Lecturer of IEEE Circuits and Systems Society.

**Wei Zhou** received the B.Eng. degree from the Huaqiao University, Xiamen, China, in 2016, and the M.Eng. degree from the Harbin Institute of Technology (ShenZhen) in 2019. She is currently working in Huawei Company. Her research interests include physical design of VLSI and hardware security.

**Yue Zheng** (S'15) received the B.Eng. degree in Communication Engineering from Shanghai University, Shanghai, China, in 2015. She is currently working toward the Ph.D. degree in the School of Electrical and Electronic Engineering of Nanyang Technological University, Singapore. Her research interests include physical unclonable functions, device fingerprinting, and hardware security.