Trustworthy Reconfigurable Access to On-Chip Infrastructure

Kochte, Michael A.; Baranowski, Rafal; Wunderlich, Hans-Joachim

Proceedings of the 1st International Test Conference in Asia (ITC-Asia'17) Taipei, Taiwan, 13-15 September 2017

doi: http://dx.doi.org/10.1109/ITC-ASIA.2017.8097125

Abstract: The accessibility of on-chip embedded infrastructure for test, reconfiguration, or debug poses a serious security problem. Access mechanisms based on IEEE Std 1149.1 (JTAG), and especially reconfigurable scan networks (RSNs), as allowed by IEEE Std 1500, IEEE Std 1149.1-2013, and IEEE Std 1687 (IJTAG), require special care in the design and development. This work studies the threats to trustworthy data transmission in RSNs posed by untrusted components within the RSN and external interfaces. We propose a novel scan pattern generation method that finds trustworthy access sequences to prevent sniffing and spoofing of transmitted data in the RSN. For insecure RSNs, for which such accesses do not exist, we present an automated transformation that improves the security and trustworthiness while preserving the accessibility to attached instruments. The area overhead is reduced based on results from trustworthy access pattern generation. As a result, sensitive data is not exposed to untrusted components in the RSN, and compromised data cannot be injected during trustworthy accesses.

Preprint

Trustworthy Reconfigurable Access to On-Chip Infrastructure

Michael A. Kochte, Rafal Baranowski, Hans-Joachim Wunderlich
ITI, University of Stuttgart, Pfaffenwaldring 47, D-70569 Stuttgart, Germany

***Abstract*—The accessibility of on-chip embedded infrastructure for test, reconfiguration, or debug poses a serious security problem. Access mechanisms based on IEEE Std 1149.1 (JTAG), and especially reconfigurable scan networks (RSNs), as allowed by IEEE Std 1500, IEEE Std 1149.1-2013, and IEEE Std 1687 (IJTAG), require special care in the design and development.**

This work studies the threats to *trustworthy* data transmission in RSNs posed by untrusted components within the RSN and external interfaces. We propose a novel scan pattern generation method that finds trustworthy access sequences to prevent sniffing and spoofing of transmitted data in the RSN. For insecure RSNs, for which such accesses do not exist, we present an automated transformation that improves the security and trustworthiness while preserving the accessibility to attached instruments. The area overhead is reduced based on results from trustworthy access pattern generation. As a result, sensitive data is not exposed to untrusted components in the RSN, and compromised data cannot be injected during trustworthy accesses.

***Index Terms*—Hardware security, trustworthiness, IJTAG, IEEE Std 1687, secure DFT, secure pattern retargeting, reconfigurable scan network**

I. Introduction

On-chip infrastructure and instrumentation is used for manufacturing test, diagnosis, debug and post-silicon validation, as well as for maintenance, monitoring, and reconfiguration in the field. The access to this infrastructure is often based on standardized scan networks, for instance according to IEEE Std 1149.1 (JTAG) or IEEE Std 1500. Recently, *reconfigurable scan networks (RSNs)*, as standardized in IEEE Std 1149.1-2013 and IEEE Std 1687-2014 (IJTAG, [1]), have been proposed to handle the growing number and diversity of instruments in the infrastructure. Such scalable low-latency scan networks allow to change the path in the network through which data is shifted to minimize access latency to the embedded instruments. In addition to bypass-based hierarchical RSNs allowed by IEEE Std 1149.1-2013, IEEE Std 1687 also allows highly flexible architectures with distributed configuration. RSNs can be accessed through a JTAG-compliant test access port (TAP) and be viewed as a reconfigurable test data register (TDR) with variable length.

The access to on-chip scan infrastructure poses a serious security problem. An attacker may exploit the scan infrastructure as a side-channel to gain access to protected data (secret key or IP), or to alter the system state to perform illegal or unsafe operations [2, 3].

Defenses against attacks that exploit the external JTAG interface (test access port, TAP) include access authorization [4–7], scan data encryption [6], and scan chain obfuscation [4, 8, 9]. The goal of these approaches is to assure that only users who know a shared secret (e.g. encryption key, challenge-response pair, or obfuscation principle) can access the scan infrastructure. In RSNs, the scan security problem is further exacerbated due to the distributed control over the access to scan segments [10, 11]. Recently proposed secure RSN architectures control the access to sensitive infrastructure by extending the RSN [12, 13] or the TAP [14], employing obfuscation, challenge-response authentication, or a filter that restricts the allowed scan accesses. These approaches provide access control and data protection at TAP level, but do not sufficiently protect against attacks from within the chip, such as sniffing or spoofing of shifted data by components in the scan network. The relevance of such *internal* threats has recently also been recognized by EDA companies [15].

Once the scan data is decrypted on chip and fed to the scan infrastructure, it is exposed to a potential attacker via components of the scan infrastructure itself or on-chip instruments accessible by the scan infrastructure, as shown in Fig. 1. For instance, if sensitive data can propagate to infrastructure components that can be observed by an attacker, such as scan segment $S2$ in the figure, sniffing attacks become possible. Likewise, if sensitive plain data is shifted to the destination register through components that can be controlled by an attacker, e.g., by fault injection, the data can be spoofed (cf. segment $S4$ and multiplexer $M1$). If there are data paths from such *untrusted* components to pins of the chip, e.g., to sensor or actor interfaces, such attacks may not even require access to the TAP (cf. scan segment $S3$).

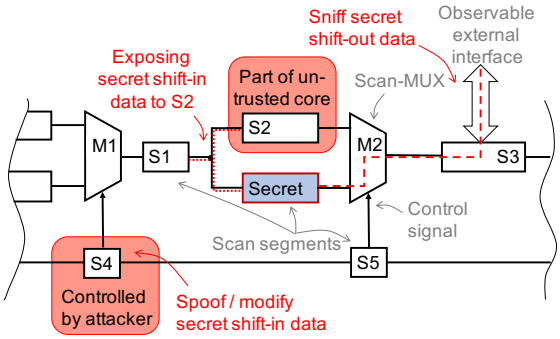

Fig. 1: Attack scenarios in a reconfigurable scan network

Permanently disabling the infrastructure access [2] is not desired since complex systems require at least limited access throughout the lifetime for in-field maintenance, monitoring, or configuration. Dedicated encryption for the communication to each attached data register, on the other hand, incurs high costs in area and power and is thus impractical.

This paper discusses the threats for trustworthy data transmission in general RSNs. We propose an algorithm to generate trustworthy accesses to existing RSN designs, such that during an access to sensitive infrastructure, secret scan data cannot be sniffed or spoofed by untrusted components or subnetworks. If this cannot be guaranteed for a given RSN, the RSN is transformed such that finally any sensitive component can be accessed in a trustworthy way, i.e., sniffing or spoofing of transmitted data by untrusted components in the RSN is not possible. This method is orthogonal to previously proposed secure RSN architectures [12–14]. Our approach defends against RSN attacks without any access time penalty, without limiting the flexibility and accessibility of the RSN, and with marginal

hardware overhead. It is applicable at the phase of system integration, when the scan infrastructure is stitched together.

The next section introduces the used terminology. Section III studies possible side-channel attacks on RSNs and defines trustworthy accesses. Section IV describes the algorithm to generate trustworthy accesses. Section V presents an RSN transformation for security improvement, followed by experimental results in Section VI.

II. RSN Terminology

The basic building blocks of RSNs comprise scan segments (registers), multiplexers, and control signals, as shown in Fig. 1. The logic state of the RSN determines which registers in the network are currently accessible, i.e. can be read or written by shifting data through the network.

A *scan segment* is a shift register of one or more bits length with a *select* control input. If *select* is active (segment *is selected*) during a *capture* operation, the shift register is loaded with data from outside of the RSN, e.g. with output of an on-chip instrument. If the segment is selected during a *shift* operation, data is shifted from the segment's scan-input, through its register bits, to the scan-output of the segment. A scan segment may include a *shadow* latch that is stable during the shift operation (as in JTAG test data registers). When the scan segment is selected during an *update* operation, the shadow latch is loaded from the shift register. A scan segment with a shadow latch can be used for bidirectional communication with an on-chip instrument or to drive internal control signals, such as *select* inputs of other scan segments.

Scan multiplexers control the path through which the data is shifted and allow to bypass scan segments. The *address* control of a scan multiplexer specifies the selected scan input.

The *select* and *address* control signals may be driven by arbitrary combinational logic that take their input from shadow latches of scan segments or external inputs. An RSN has a *primary scan-input*, a *primary scan-output*, and global control signals for the capture, shift, and update operations.

A *scan path* is a non-circular connection of scan segments and multiplexers from a primary scan-in to a primary scan-out port. A scan path is *active* if and only if (iff) the select signals of all on-path segments are asserted and all on-paths multiplexers address the input belonging to the active scan path. A *subnetwork* is a connected subset of RSN elements.

A *scan configuration* is the logic state of all sequential elements and external control signals. It is assumed that after reset or power-up all sequential elements are in a known state ('0' or '1'). A scan configuration is *valid* if and only if: (i) one active scan path exists and (ii) scan segments that do not belong to the active scan path are deselected. This ensures that the shift-in data is delivered to the target scan segments, the captured data is shifted toward the primary scan-out, and all scan segments that do not participate in the access (i.e., do not belong to the active scan path) are stable [16].

A *scan access* in an RSN is a sequence of atomic *CSU* operations, each with three phases: *capture, shift, and update*. During capture, the scan segments on the active scan path may latch new data. Then, this data is shifted out while new scan data is shifted in. During the update phase, the shifted-in data is latched in the shadow registers on the active scan path. A read or write access to a scan segment requires the accessed segment to be part of the active scan path.

III. Threats and Defenses

The attacker aims to sniff or spoof secret data that is written to or read from the RSN by an authorized entity. In general, the attacker may achieve this via the following *channels*:

Channel 1: observation and/or manipulation of the external RSN interface, e.g., the TAP,

Channel 2: observation and/or manipulation of internal scan elements of the RSN, including shift registers of scan segments, scan multiplexers, etc.,

Channel 3: observation of logic elements to which sensitive scan data may propagate to, potentially via combinational or sequential functional logic,

Channel 4: manipulation of logic elements that impact how data is shifted through the RSN (e.g., shadow latches of scan segments driving control signals).

The attacker may also combine any of the above methods using power analysis, fault injection, or micro-probing.

A. Existing Defenses

To protect against unauthorized access at the external RSN interface, e.g., the JTAG TAP, the interface is secured by authorization and message authentication [4–7]. To prevent external sniffing and spoofing, stream ciphers are placed at the TAP to decrypt input shift data from the TDI and encrypt output shift data at the TDO of the TAP [6]. Combined with message authentication, such a protection scheme prevents sniffing and spoofing of secret data at the (external) TAP level, and hence limits the risk of an attack via channel 1. Internally, however, the shift data is transported through the RSN in plain text and is therefore prone to attacks via internal components.

The unencrypted shift data may be both exposed or modified within the RSN (channel 2). For instance, if the attacker has control over data in a scan segment or can abort CSU operations, the data in any downstream scan segment on the active scan path may be corrupted. A premature CSU abort may be caused by manipulating the JTAG control signals (TCK/TMS, if accessible) or by fault injection attacks. Scan data may also leak to functional logic or external pins through scan chains or instruments (channel 3). Moreover, the attacker may configure the scan network so as to propagate the shift data to external pins or instruments (channel 4). The architectures for access protection in RSNs [12, 13] do not target the protection of sensitive shift data against threats from within the RSN.

To protect sensitive data from such internal threats, encryption and authentication circuitry has to be placed locally [17] to locally decrypt shift-in and encrypt shift-out data of trusted subnetworks. However, if multiple scan segments or subnetworks require protection, this scheme becomes unwieldy and unaffordable due to high hardware overhead.

B. Threat Model

If a scan segment or subnetwork potentially exposes scan data, can be controlled by the attacker, or is unknown third-party IP, it is regarded as *untrusted*. Scan data may be exposed by untrusted components directly, e.g., by propagation to observable pins or instruments, possibly through combinational

and/or sequential logic, or it may leak indirectly, e.g., by causing a measurable effect in power consumption.

Our threat model assumes that the attacker has internal access to the *untrusted* RSN components, but has no TAP-level access to the RSN:

- Attacks via channel 1 are not possible (the TAP is protected, for instance by TAP-level scan data encryption and message authentication, as in [6]).
- Attacks via channels 2, 3, and 4 are possible.
- The system integration phase is secure and the system integrator can distinguish trusted components (scan segments and subnetworks) from untrusted ones.
- The manufacturing process is trustworthy, as is the case for secure or safety-critical products.
- Invasive attacks (chip dismantling/microprobing) are unattainable to the attacker or precluded by physical design.

C. Trustworthy Access

The goal of the proposed approach is to facilitate a *trustworthy access*. An access is considered trustworthy if the attacker is unable to sniff or spoof the scan data while an authorized entity performs the access. We do not intend to restrict the RSN to trustworthy accesses only (can be achieved using [14]).

Def. 3.1: A *trustworthy access* is a sequence of read/write accesses to trusted scan segments, such that:

- the scan data does not pass through untrusted subnetworks (e.g., untrusted scan segments), and
- the configuration of the active scan path is stable regardless of the state of untrusted control signals (external or controlled by untrusted subnetworks).

Def. 3.2: A scan segment or subnetwork is *trusted* if:

- the subnetwork does not distribute scan data that is written to it or passing through it to any components that may be observed by the attacker, except for propagating the data to its own scan output (which may be connected to the scan input of an untrusted subnetwork), and
- the subnetwork's state is not controllable by the attacker. (This prevents the attacker e.g. from reconfiguring the network and exposing secret scan data.)

IV. Trustworthy Access Generation

The proposed method is based on a formal model of the RSN (Sec. IV-A) to build a Boolean formula that is satisfied by trustworthy accesses only (Sec. IV-B). A SAT solver checks the satisfiability of that formula: If it is satisfiable, a trustworthy access exists and the required scan patterns are derived from the satisfying assignment. Otherwise, when the formula is unsatisfiable, no trustworthy access is found since the RSN structure is inherently insecure. The RSN is then transformed using bypass, masking, and isolation logic to improve its security and to enable trustworthy accesses (Sec. V). The final RSN includes only a small amount of additional logic, required to assure trustworthy access. This approach causes no access time penalty, does not constrain the the RSN topology, and incurs only marginal hardware overhead.

A. Formal CSU-Accurate RSN Model (CAM)

We represent RSNs with the CSU-accurate model (CAM) of [16] that is understood as an abstract FSM. A transition in the CAM corresponds to a complete CSU operation and covers multiple clock cycles of actual operation. The CAM is constructed from a register-transfer level (RT-level) RSN description in Instrument Connectivity Language (ICL; cf. IEEE Std 1687). Details of the construction are given in [16].

Def. 4.1: The *CSU-accurate RSN model* (CAM) $\mathcal{M} = \{S, I, C, c_0, T\}$ consists of a set of state elements S, a set of external control signals I, a set of scan configurations $C \subseteq \{0, 1, X\}^{|S \cup I|}$, the initial scan configuration $c_0 \in C$, and a transition relation $T \subseteq C \times C$. Each state element $s \in S$ corresponds to a 1-bit shadow register of a scan segment in the RSN. A scan configuration $c \in C$ specifies the state of all elements in S and external inputs in I. It is also interpreted as a function $c : S \cup I \to \{0, 1, X\}$ that maps each element $e \in S \cup I$ to state $c(e)$. The transition relation T includes all pairs of scan configurations $(c_1 \in C, c_2 \in C)$ such that c_2 is reachable from c_1 within one CSU operation.

Def. 4.2: The *characteristic function* of a transition relation T of $\mathcal{M} = \{S, I, C, c_0, T\}$ is defined as:

$$T(c_1, c_2) := \bigwedge_{s \in S} \big[[(\texttt{Active}(c_1, s) = 0) \Rightarrow (c_2(s) = c_1(s))] \wedge [(\texttt{Active}(c_1, s) = X) \Rightarrow (c_2(s) = X)]\big],$$

where the predicate $\texttt{Active} : C \times S \to \{0, 1, X\}$ assigns each element $s \in S$ and configuration $c \in C$ a value which is true ($\texttt{Active}(c, s) = 1$) exactly when s is selected in c and c is a *valid* scan configuration, i.e., when s belongs to the active scan path in c.

The transition relation $T(c_1, c_2)$ defines the conditions for state changes in the RSN: if a state element $s \in S$ does not belong to the active scan path in c_1, the state of s does not change between c_1 and c_2. The state of s can only change in a deterministic way if s belongs to the active scan path in c_1 and c_1 is a valid scan configuration. We also assume that the state of s in c_2 becomes unknown (X) when it is unsure whether s belongs to the active scan path in c_1, i.e., when $\texttt{Active}(c_1, s) = X$. Note that for invalid scan configurations, all predicates `Active` evaluate to X.

The CAM is a sound abstraction: Properties that hold in the CAM also hold in the RT-level RSN model, assuming that the internal control signals (e.g. multiplexer addresses) are stable during the capture and shift phases [16]. This holds trivially for control signals driven by shadow registers of scan segments, as they may only change during the update operation. The stability of external signals must be guaranteed by the system logic external to the RSN, otherwise the active scan path may change during shifting and cause security issues. The stability of external signals should be formally verified in the RT-level RSN design prior to CAM extraction.

B. Trustworthy Access Generation

Trustworthy access generation is the search for scan patterns that must be shifted into the RSN during one or multiple CSU operations to read or write trusted scan segments, so that no data is shifted through untrusted segments, and no untrusted segment may affect the read or write access. In general, access generation for RSNs requires sequential logic justification and is an NP-hard problem.

Let $\mathcal{M} = \{S, I, C, c_0, T\}$ be the CAM of an RSN with a set of trusted scan segments S_{tr}, untrusted scan segments S_{untr},

trusted control inputs I_{tr}, and untrusted control inputs I_{untr}, such that $S_{\text{tr}} \cup S_{\text{untr}} = S$ and $I_{\text{tr}} \cup I_{\text{untr}} = I$. For the sake of brevity, we specify a scan access by its initial ($c_0 \in C$) and target ($c_t \in C$) scan configurations as (c_0, c_t). This is adequate to model concurrent access to multiple scan segments (access merging), both for read and write operations.

Trustworthy scan pattern generation for access (c_0, c_t) is then the computation of a sequence of $n \in \mathbb{N}^+$ scan configurations $c_1, c_2, \ldots, c_n$ assuming that:

$$\forall_{i=0\ldots n} \forall_{s \in (S_{\text{untr}} \cup I_{\text{untr}})} c_i(s) = X \tag{1}$$

such that the following conditions hold:

$$c_n = c_t \wedge \tag{2}$$

$$\forall_{i=1\ldots n} \left((c_{i-1}, c_i) \in T\right) \wedge \tag{3}$$

$$\forall_{i=0\ldots n} \forall_{s \in S_{\text{untr}}} \texttt{Active}(c_i, s) = 0 \tag{4}$$

Assumption (1) states that the access must be performed regardless of the state of untrusted elements $S_{\text{untr}}, I_{\text{untr}}$. The scan patterns are hence generated with the pessimistic assumption that all untrusted elements have unknown state (X).

Conditions (2) and (3) are satisfied iff $(c_0, c_1, \ldots, c_n)$ is a valid sequence of consecutive scan configurations, such that the last configuration equals the target configuration. To prevent sniffing and spoofing, condition (4) requires that untrusted scan segments are never part of the active scan path.

The conjunction of conditions $(1) \wedge (2) \wedge (3) \wedge (4)$ is transformed into conjunctive normal form (CNF) with 3-valued variables. The formula forms a SAT problem instance with $c_i(s)$ being free variables for $s \in (S_{\text{tr}} \cup I_{\text{tr}})$ and $i = 1 \ldots n-1$.

If the SAT instance is satisfiable, there exists a valid and trustworthy scan access, such that $c_n = c_t$ and untrusted elements can neither affect the intermediate scan configurations, nor sniff or spoof the scan data. The *satisfying assignment* provides the state of all scan segments in the scan configurations and is easily translated to scan patterns [16].

If the SAT instance is unsatisfiable, a trustworthy scan access (c_0, c_t) with n CSU operations does not exist. In this case the SAT instance is extended with additional clauses to reflect $n+1$ CSU operations, and its satisfiability is examined again. This procedure is started with $n = 1$ and iterated until the instance is satisfiable, or until a user defined bound is reached (without a solution). In the latter case, the network is subject to a transformation, as described below.

V. RSN Transformation

If no access is found by the method of Section IV-B, the RSN is modified to facilitate trustworthy accesses. To this end, it is extended with masking logic, bypasses, and isolation cells for untrusted subnetworks. The transformation enables a trustworthy access to the targeted scan segments without limiting the access to any other (trusted or untrusted) segments. The masking, bypass, and isolation logic is activated with a global control signal *TrustEnable*, which indicates that a trustworthy access takes place. This signal must be reliably set by a trusted component, e.g., the TAP controller, and must not be routed via any logic of untrusted subnetworks.

The transformation is performed as follows: Initially, all untrusted subnetworks are enclosed with bypasses, and subnetworks driving internal control signals are equipped with masking logic. Next, we use an iterative SAT-based algorithm to identify a small subset of *essential* transformations such that trustworthy accesses to all trusted segments are guaranteed. Finally, isolation cells are inserted where required.

A. Bypass and Masking Logic

A trustworthy access requires that scan data is not shifted through untrusted scan segments. To guarantee that a trustworthy access exists, each untrusted scan segment or subnetwork is enclosed with a scan multiplexer that allows to bypass it. The control signal of this multiplexer is driven by the global signal *TrustEnable*.

To prevent untrusted subnetworks from controlling the active scan path, each control signal driven by an untrusted subnetwork is cut and connected to an additional 1-bit scan segment placed in the bypass path of the untrusted subnetwork. This transformation is shown in Fig. 2. The bypass of untrusted segment or subnetwork $S4$ (dotted) allows to reroute secret data so that they are not shifted through $S4$ during a trustworthy access. The additional scan segment $By4$ in the bypass path of $S4$ provides a way to reliably drive the control signals during a trustworthy access. Both the "masking control" and "bypass control" signals are driven by *TrustEnable*.

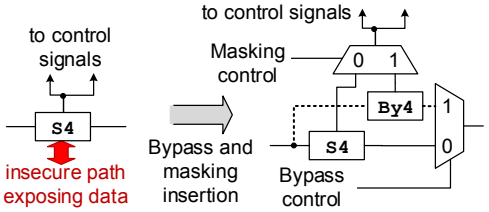

Fig. 2: Example transformation for untrusted segment $S4$

Note that $S4$ in Fig. 2 may represent an untrusted subnetwork with many untrusted scan segments. Just a single scan multiplexer is sufficient to bypass an entire untrusted subnetwork with one scan input and output, and just one n-bit scan segment is required for an untrusted subnetwork driving n control signals. This transformation maintains accessibility of the original RSN, i.e., there exists a trustworthy access to every trusted segment that was accessible in the original RSN. Access to untrusted segments is not limited.

B. Identification of Essential Transformations

We use the heuristic method described below to identify the set of *essential* transformations which are at least required for trustworthy accesses. The remaining non-essential transformations are redundant and removed from the RSN without affecting trustworthy accesses.

Only for the identification of essential transformations, we assume that each bypass and masking multiplexer is controlled with a separate control input $b \in B$, where $B \subseteq I$. In the actual RSN design, all these inputs are driven by *TrustEnable*. Trustworthy scan pattern generation, as described in Section IV-B, is performed on the CAM of the transformed RSN, i.e., with all bypass and masking logic. At first, however, we assume that bypasses and masking logic are inactive during all CSU operations, i.e., the transformed RSN is functionally equivalent with the original RSN. Under this assumption, a

trustworthy access (c_0, c_t) with n CSU operations exists if the following SAT instance is satisfiable:

$$\underbrace{(1) \wedge (2) \wedge (3) \wedge (4)}_{\text{trustworthy access (cf. Sec. IV-B)}} \quad \wedge \quad \underbrace{\forall_{i=0\ldots n} \forall_{b \in B} \, (c_i(b) = 0)}_{\substack{\text{assumptions to deactivate} \\ \text{bypass/masking logic}}}. \qquad (5)$$

To find the set of essential transformations, we follow an iterative algorithm based on incremental SAT solving. We allow the activation of a transformation by removing the assumption for its corresponding control signal from formula (5). The assumptions on transformations are revoked one by one until the formula becomes satisfiable and a trustworthy access is found. The selection of assumptions to revoke is guided by the incremental SAT solver, as explained below.

An incremental SAT solver checks the satisfiability of a formula of the form $F \wedge A$, where F is a formula in CNF form, and A is a conjunction of *assumptions* $A = \{a_0, a_1, a_2, \ldots\}$, where a_i are unit clauses (literals). If the formula $F \wedge A$ is unsatisfiable, a small set of *relevant assumptions* $A' \subseteq A$ can be extracted, such that the formula $F \wedge A'$ is still unsatisfiable. The search for relevant assumptions is computationally inexpensive and supported by state-of-the-art SAT solvers [18].

Here, the SAT instance is split into a CNF formula satisfiable for trustworthy accesses, and a set of assumptions activating or deactivating the bypass and masking logic in each CSU operation, as marked below formula (5). Intuitively, if the instance is unsatisfiable, the computed set of relevant assumptions includes the literals corresponding to candidates for essential transformations.

Let us assume that just one trustworthy access needs to be supported. We check the satisfiability of formula (5) and compute the set of relevant assumptions A'. We then pick one transformation, such that its corresponding negated literals occur most often in relevant assumptions (i.e. for the largest number of CSU operations). We mark it as an essential transformation and negate the corresponding literals in the set of assumptions A (effectively activating this transformation). We repeat this procedure until the SAT instance is satisfied and all essential transformations are found. The remaining transformations are not required to perform the targeted trustworthy access and hence do not need to be implemented in the RSN.

This algorithm is extended to support multiple trusted accesses. To select one essential transformation, the satisfiability of all trusted accesses is checked. A transformation is essential if the negations of its corresponding literals occur most often among relevant assumptions computed for all accesses.

This algorithm is a heuristic that does not guarantee the optimal solution. The search for the minimal set of essential transformations can be formulated as a pseudo-Boolean optimization problem. For a large number of trustworthy accesses, however, this problem becomes very hard or impossible to solve due to the required computational effort. Since our heuristic provides very good results at low computational costs, we omit the discussion of the optimal solution.

C. Isolation Cells

Secret scan data may be sniffed at the scan inputs of untrusted subnetworks even if the data is not shifted through them. We prevent this by isolation cells (AND gates), placed in front of each scan input of untrusted subnetworks if there is a combinational path from a trusted scan output to that input. The isolation cells are activated by the *TrustEnable* signal.

VI. Evaluation

We evaluate the proposed method on multiple reconfigurable scan networks based on ITC'02 SOC benchmarks. A randomized subset of scan segments is considered untrusted, i.e. possibly misused by the attacker to either expose or modify shift data. We generate trustworthy accesses to another randomized subset of trusted elements, such that untrusted segments do not take part nor interfere in the accesses. If a trustworthy access is impossible in the original RSN, our approach finds the required essential transformations (bypass and masking logic) that render a trustworthy access possible. All experiments are executed on a single core of an Intel Xeon CPU operating at 3.33 GHz.

A. Benchmark Circuits

Our benchmark RSNs are based on the ITC'02 SOC benchmark circuits and provide configurable access to boundary and internal scan chains, as proposed in [16]. We distinguish two access modes: configuration access and data access. Configuration access mode allows to reconfigure the scan chain of a core by attaching or detaching its scan segments for inputs, outputs and internal scan chains, as well as subnetworks of constituent submodules. Fig. 3 shows the hierarchical architecture for the top-level part of the p34392 benchmark RSN. The scan chain of each module starts with a 1-bit configuration register AM that distinguishes between configuration ($AM = 0$) mode, in which only the configuration registers (C) can be accessed, and data access mode ($AM = 1$).

Table I shows the number of scan segments of these RSNs. The RSNs are synthesized using a 45nm standard cell library.

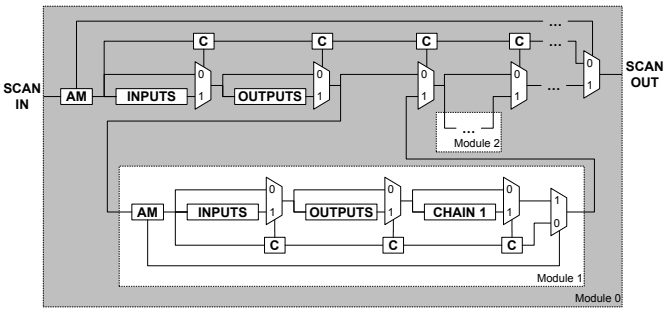

Fig. 3: RSN architecture for an ITC'02 benchmark circuit

B. Results

For each RSN, we perform a series of randomized experiments, in which (a) 1, 2, 5, 10, 20, 50, 80, 90, or 95% of scan segments are untrusted, and (b) 1, 2, 5, 10, 20, or 50 sensitive scan segments require a trustworthy access. For each combination of these two parameters, we evaluate the avg. amount of essential transformations required for the trustworthy accesses and the runtime. Each experiment is run 10 times with different random samples of untrusted segments and trustworthy accesses. Random samples result in hard instances since clustering of untrusted segments is less likely.

Fig. 4 shows the ratio of the average number of essential bypasses relative to the number of untrusted scan segments

in the largest benchmark p93791. This ratio depends on the number of sensitive scan segments that require a trustworthy access: For instance, in a network with 90% of untrusted scan segments, about 5% of those elements require a bypass to enable the trustworthy access to 1 random scan segment. For the trustworthy access to 50 scan segments, already 45% of untrusted scan segments must be bypassed. The proportion of essential bypasses is nearly linear with the proportion of untrusted scan segments in the RSN. Similar results are obtained for the proportion of essential masking logic. The worst case runtime for the proposed method for the largest benchmark p93791 is 25 seconds.

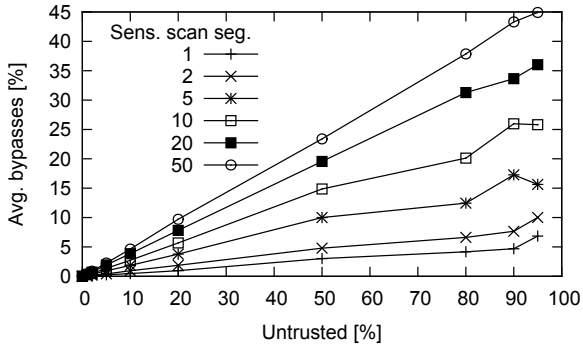

Fig. 4: Average amount of bypasses required for p93791

Table I shows the area overhead of the RSN transformation for 10 and 50 sensitive segments that require trustworthy accesses and different percentages of untrusted segments. The overhead (as percentage of the RSN area) ranges from 0.01% up to only 4.74%. The table also shows the maximum runtime (column t) observed during the iterations of the experiments. In most cases, the runtime is much less than one second.

TABLE I: Area overhead w.r.t. RSN area for different numbers of trustworthy accesses and percentages of untrusted segments

Design	Scan segm.	Num. acc.	Overhead [%] for untrusted segm. 5%	10%	20%	50%	95%	Max. t [s]
u226	99	10	0.27	0.52	1.08	2.64	4.60	0.1
		50	0.50	0.85	1.51	3.68	4.74	0.1
d281	117	10	0.11	0.24	0.51	1.23	1.96	0.1
		50	0.20	0.35	0.71	1.65	2.02	0.1
d695	335	10	0.13	0.28	0.56	1.53	2.83	1.0
		50	0.20	0.38	0.77	1.86	3.27	1.1
h953	109	10	0.11	0.17	0.35	0.78	1.18	0.1
		50	0.16	0.25	0.47	1.08	1.22	0.1
g1023	159	10	0.09	0.17	0.36	0.92	1.54	0.2
		50	0.19	0.31	0.61	1.53	1.58	0.2
f2126	81	10	0.03	0.05	0.09	0.24	0.38	0.1
		50	0.04	0.07	0.12	0.28	0.39	0.1
q12710	51	10	0.01	0.02	0.04	0.10	0.11	0.0
		50	0.02	0.03	0.05	0.11	0.12	0.0
p22810	565	10	0.03	0.08	0.16	0.43	0.81	2.5
		50	0.08	0.15	0.30	0.78	1.38	4.3
p34392	245	10	0.04	0.07	0.14	0.36	0.69	0.7
		50	0.06	0.09	0.19	0.47	0.73	0.7
p93791	1241	10	0.03	0.06	0.13	0.33	0.58	16.6
		50	0.05	0.11	0.23	0.54	1.03	24.8
t512505	319	10	0.01	0.02	0.04	0.11	0.20	0.6
		50	0.02	0.04	0.08	0.18	0.23	0.7
a586710	79	10	0.01	0.02	0.03	0.08	0.13	0.1
		50	0.02	0.02	0.05	0.11	0.14	0.1

VII. Conclusion

While reconfigurable scan networks (RSNs) provide flexible access to the growing number of on-chip instrumentation, they also pose a security issue. This work presented a novel algorithm to generate trustworthy accesses to sensitive scan registers in RSNs to protect against attacks from within the RSN. Trustworthy accesses ensure that untrusted components within the RSN cannot sniff or spoof the transmitted data, or perform a reconfiguration-based attack that could potentially corrupt the access. If the structure of the original RSN does not allow for a trustworthy access, we efficiently transform the RSN to bypass, mask, and isolate untrusted components. The runtime of the method is very low. The area overhead of the transformation for RSN trustworthiness ranges from 0.01% up to only 4.74% w.r.t. the original RSN area.

Acknowledgments

This work was supported by the Baden-Württemberg Stiftung (IKT-Sicherheit, SHIVA).

References

[1] "IEEE Std 1687-2014 – IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device," Dec. 2014, IEEE Comp. Society.

[2] M. Tehranipoor and C. Wang, *Introduction to Hardware Security and Trust*. Springer, 2011.

[3] B. Yang, K. Wu, and R. Karri, "Scan Based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard," in *Proc. Int'l Test Conf. (ITC)*, 2004, pp. 339–344.

[4] J. Lee, M. Tehranipoor *et al.*, "Securing Designs against Scan-Based Side-Channel Attacks," *IEEE Trans. Dependable and Secure Computing*, vol. 4, no. 4, pp. 325–336, Oct.-Dec. 2007.

[5] K. Park, S. Yoo *et al.*, "JTAG Security System Based on Credentials," *Journal of Electronic Testing (JETTA)*, vol. 26, pp. 549–557, 2010.

[6] K. Rosenfeld and R. Karri, "Attacks and Defenses for JTAG," *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 36–47, Jan.-Feb. 2010.

[7] G.-M. Chiu and J.-M. Li, "A Secure Test Wrapper Design Against Internal and Boundary Scan Attacks for Embedded Cores," *IEEE Trans. VLSI Syst.*, vol. 20, no. 1, pp. 126–134, Jan. 2012.

[8] H. Fujiwara and M. Obien, "Secure and Testable Scan Design Using Extended de Bruijn Graphs," in *Proc. Asia and South Pacific Design Automation Conf. (ASP-DAC)*, 2010, pp. 413–418.

[9] Y. Shi, N. Togawa *et al.*, "Robust Secure Scan Design Against Scan-Based Differential Cryptanalysis," *IEEE Trans. VLSI Syst.*, vol. 20, no. 1, pp. 176–181, Jan. 2012.

[10] J. Dworak and A. Crouch, "A Call to Action: Securing IEEE 1687 and the Need for an IEEE Test Security Standard," in *Proc. IEEE VLSI Test Symp. (VTS)*, April 2015, pp. 1–4.

[11] M. A. Kochte, M. Sauer *et al.*, "Specification and verification of security in reconfigurable scan networks," in *Proc. IEEE European Test Symposium (ETS)*, May 22-26 2017.

[12] A. Zygmontowicz, J. Dworak *et al.*, "Making It Harder to Unlock an LSIB: Honeytraps and Misdirection in a P1687 Network," in *Proc. Conf. Design, Automation & Test in Europe (DATE)*, 2014, pp. 195:1–195:6.

[13] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Fine-Grained Access Management in Reconfigurable Scan Networks," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 34, no. 6, pp. 937–946, 2015.

[14] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Securing Access to Reconfigurable Scan Networks," in *Proc. IEEE Asian Test Symp. (ATS)*, 2013.

[15] G. Tshagharyan, G. Harutyunyan *et al.*, "Securing test infrastructure of system-on-chips," in *Proc. IEEE East-West Design Test Symposium (EWDTS)*, Oct 2016, pp. 1–4.

[16] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Reconfigurable Scan Networks: Modeling, Verification, and Optimal Pattern Generation," *ACM Trans. Design Automation of Electronic Systems (TODAES)*, vol. 20, no. 2, pp. 30:1–30:27, 2015.

[17] K. Rosenfeld and R. Karri, "Security-Aware SoC Test Access Mechanisms," in *Proc. IEEE VLSI Test Symp. (VTS)*, 2011, pp. 100–104.

[18] A. Nadel, "Boosting Minimal Unsatisfiable Core Extraction," in *Proc. Formal Methods in Comp.-Aided Design (FMCAD)*, 2010, pp. 221–229.