Provably Secure Sequential Obfuscation for IC Metering and Piracy Avoidance

Farinaz Koushanfar
Electrical and Computer Engineering Department,
University of California at San Diego,
La Jolla, CA 92093 USA

Editor's notes:
This article introduces formal security proofs in the context of integrated circuits obfuscability.

—Rosario Cammarota, Intel Labs
—Francesco Regazzoni, University of Amsterdam and Università della Svizzera Italiana

The escalating cost of updating and maintaining silicon foundries has caused a major paradigm shift in the semiconductor business model. Many of the key design houses are entirely fabless (i.e., without a fabrication plant), outsourcing their fabrication to third-party providers. Several design companies that have traditionally fabricated their designs in-house have either formed alliances to share the cost or have moved parts of their fabrication offshore to third-party providers.

In the earlier vertical market model, in-house fabrication together with the clandestine nature of the packaged chips was enough for IP protection. In the new model, however, fabrication outsourcing requires revealing the IP to external entities, creating a risk of infringements. The Alliance for Gray Market and Counterfeit Abatement has estimated that about 10% of the leading edge technology products available on the market are counterfeits. Several government and industry task forces are actively working to address these issues.

To enable security of the designer IPs and ICs, a suite of security mechanisms and protocols based on obfuscating and/or hiding information within the design has been introduced. The earliest known method of this family was called *hardware metering* [1], a term originally coined by Koushanfar, Qu, and Potkonjak. Metering in [1] suggests unique (passive) functional identification of ICs made by the same mask. The metering methodology was the first that could be used for specific functional tagging of ICs, or monitoring and estimating the number of fake components in the case of piracy detection. Active metering additionally enables the designers to have post-fabrication control over their designed IPs by actively controlling the number of used ICs, monitoring their properties and usages, and/or by remote runtime enabling/disabling [2].

In a typical hardware metering scenario, each chip is uniquely and unclonably identified, for example, by using a physical unclonable function (PUF) [3]. A PUF extracts the unique delay or current variations on each chip to assign a set of unclonable

Digital Object Identifier 10.1109/MDAT.2021.3065324
Date of publication: 12 March 2021; date of current version: 20 May 2021.

identifiers (IDs). To meter, the IDs are linked to parts of the IC's functional components. This way, a part of the design functionality is uniquely tailored to the unclonable properties (fingerprint) of the IC and is used to form a unique (obfuscated) lock for each IC's functionality. Only the designer who has the knowledge of the high-level design would be able to provide the specific key to unlock each of the ICs. Note that methods for the unique identification of chips based on process variations became available about two decades ago, but hardware metering was the first to hide the unique chip identifiers into the IC's functionality. The hiding mechanism was based on the *IC obfuscatability*.

The nominated paper presents a detailed description, comprehensive analysis, and new results including the first formal proof for obfuscating the active locking mechanism within the sequential description of the circuit. We emphasize that the sequential description of the control part of the circuitry is the core of most contemporary ICs. For example, cryptographic functions are most often sequentially implemented.

The novel aspect of this top pick paper compared to all previously published obfuscation and active hardware metering methods is introducing a secure construction of the active metering locks by extending the chip's behavioral specification in the finite states domain. The pioneering work in [2] creates a novel mechanism for actively metering by obfuscating the locks within the finite state machine (FSM) but does not provide formal security proof. The nominated paper [4] shows that the construction of locks by finite-state manipulation and compilation during the hardware synthesis and interfacing to a unique PUF state is an instance of an efficiently obfuscatable program under the random oracle model. The significance of the provably secure construction for the obfuscating FSM goes beyond hardware locking and metering and extends to the earlier work in information hiding in sequential circuits for which only heuristic solutions had been available [1], [5]. In addition to formal definitions and proofs, experimental evaluations are presented in this article.

In summary, the contributions of [4] are as follows.

- It presents the first formal proof for IC obfuscatability.
- It shows a construction of the logic obfuscation in FSM as an instance of a general output multi-point function family; this family is effectively obfuscatable in a random oracle model.
- It demonstrates an automatic low-overhead realization of secure IC locking/metering by obfuscatable topology construction, secure passkey selection, and iterative synthesis.
- It discusses the potential attacks and analyzes the security of the presented method against each attack.
- Proof of concept construction of the obfuscated FSM is evaluated on standard benchmark circuits demonstrating the applicability and practicality of the method.

Key ideas

The nominated paper has opened a new chapter in the important area of logic locking/obfuscation by bridging the gap between theoretical cryptography/provably secure obfuscation and the more experimental hardware locking. It targets secure and unbreakable realization of the first and perhaps the most promising active hardware metering method, digging into its security roots, and introducing new formal security proofs. The IC control mechanism utilizes: 1) the functional description of the design and 2) unique hardware identifiers of the IC obtained from a PUF. This article, for the first time, develops proofs of obfuscation and introduces the theoretical concepts underlying provable obfuscation to the hardware security community. In the particular case study emphasized in this article, the locks are embedded by modifying the structure of the hardware computation model, in the form of an FSM. However, the time has shown that the proposed obfuscation proofs and techniques are general; they have since found applications in combinational logic obfuscation/locking and other security tasks.

The paper calls the modified (obfuscated) FSM—a boosted FSM (BFSM), which includes multiple added states and transitions. The initial power-up and/or control states of BFSM are determined by a PUF which is unique for each IC. More specifically, for each IC, it is shown that hiding the locks within the BFSM can be constructed as an instance of a general output multi-point function that can be provably and efficiently obfuscated under the random oracle model. The hidden locks within the FSM may also be

used for remote enabling and disabling of chips by the IP rights owner during the IC's normal operation.

Technical contributions

Assume that the original FSM has $|S|$ states. Therefore, it can be implemented using $K = \log|S|$ flip-flops (FFs). The BFSM has $|S'| + |S|$ states that can be implemented by $K'' = \log\{|S'| + |S|\}$ FFs. Observe that for a linear growth in the number of FFs, $K' = K'' - K$, the number of states exponentially increases. The process is demonstrated in Figure 1 with a small example. On the left side of the figure, there is a PUF unit that generates random bits based on the unclonable process variations of the silicon unique to each chip. A fixed challenge is applied to the chip upon power-up. The PUF response is fed to the FFs that implement the BFSM. Since there are $K'' = \log\{|S'| + |S|\}$ FFs in the BFSM, one would need K'' response bits from the PUF for a proper operation.

Unless the design is in one of the functional states, it would be *locked*. The control of the locking states is determined by the unique response from a PUF placed on each chip. The value of K'' is set such that for a uniform probability of selecting the state, the probability of the controller being in the unknown (locked) states is extremely high, i.e., $2^{K''} \gg 2^{K}$. The PUF challenges are determined by fixed test vectors given by the designer. Therefore, the random FF states driven by the PUF response would place the design in a nonfunctional state. One would need to provide inputs to the FSM (the PUF challenge) so it can transition from this nonfunctional (locked) state to the functional state of the original FSM. For the IP rights owners with access to the BFSM state transition graph, finding the set of inputs for traversing from the locked state to the functional state is trivial. However, there is only one combination from an exponentially large number of possibilities for the input corresponding to each edge transition. Therefore, it is extremely hard for anybody without access to the BFSM edge transition keys to find the exact inputs that cause traversal to the functional states.

The set of inputs corresponding to each traversed edge during unlocking is the *passkey* of the chip. A set of passkeys (α_1, α_2, α_3) is shown in Figure 1. This locking and unlocking mechanism provides a way for the designer to *actively control (meter)* the number of unlocked functional *(activated)* ICs from one blueprint (mask), and hence the name active hardware metering.

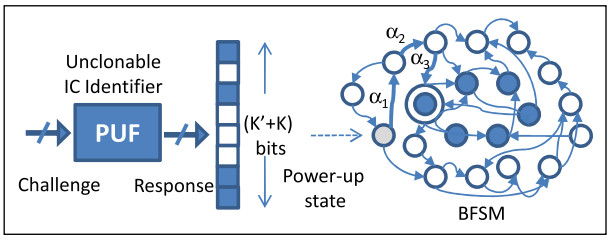

Figure 1. Construction of a BFSM. The PUF response is fed to the FFs storing the states of the BSFM. The original states are shown in dark, and the added states are demonstrated in white color on the state transition graph.

While constructing the BFSM for hardware metering purposes, a number of requirements must be satisfied. The first set of requirements has to do with the probability of randomly powering-up in a state that was not in the original FSM. Let us assume that by design, we require this probability to be lower than a given value of $\in$. This low probability is satisfied by the following two conditions.

- The value $|S'|$ should be selected such that the probability of not powering-up in an added state is smaller than $\in$

$$P(\text{power-up} \in S') = \frac{S' - S}{S' + S} \geq 1 \in . \qquad (1)$$

- The value $|S'|$ should be selected so that the probability of two ICs having the same start-up states is extremely low.

The paper provides the theoretical maximum value of the total number of chips given the value of $|S'|$ as well as the probability of two ICs starting up at the same state.

Theoretical contributions

The most significant contribution of this article is presenting the first set of provable security guarantees for active hardware metering. It demonstrates that a secure construction for the BFSM by the addition of states and transitions to the original graph can be modeled as a general output *multi-point function* and thus can be efficiently obfuscated. The structure creates a strong passkey mechanism to enable secure metering. It also shows that exploring parts of the passkeys for transitions on the BFSM would not reveal the remainder of passkeys, as long as the two passkey sets are not correlated and do not include

all the states. Therefore, as long as the traversal paths from the power-up state to the original set of states for a locked IC are different in at least one state from a previously unlocked IC, obfuscation of a multi-point function remains secure.

Formally, given an original specification of a circuit in the form of an FSM = $(\Sigma,\Delta,S,s_0,\delta,\lambda)$, where $\Sigma \neq 0$ is the set of input symbols, $\Delta \neq 0$ is the set of output symbols, $S = \{s_0,s_1,...\} \neq 0$ denotes the set of states, s_0 is the "functional" state, δ (s,i) is the transition function on s and i $(S \times \Sigma \rightarrow S)$, λ (s,i) is the output function on s and i $(S \times \Sigma \rightarrow \Delta)$, and given a compiler that transforms this functional specification to a netlist, the objective is to construct the BFSM = $(\Sigma + \Sigma',\Delta,S + S',s_0,\delta + \delta',\lambda + \lambda')$ such that transitions from a state $s' \in S'$ to one of original FSM states $s \in S$ would require "strong" passkeys. A passkey sequence of length l denoted by $\alpha = \{\alpha_1,...,\alpha_l\}$ applied to the state s' would result in a sequence of transitions before it gets to a functional state s. By strong passkeys, we mean that they cannot be broken by the brute force attack. The reached state s then would be $s = \delta(s', \alpha) = \delta(\delta(\delta(...\delta(s',\alpha_1)...),\alpha_{l-1}),\alpha_l)$. The corresponding output would be $\lambda(s',\alpha) = \lambda(\delta(\delta(...\delta(s',\alpha_1)...),\alpha_{l-1}),\alpha_l)$.

To target metering, the paper devises a BFSM construction such that the addition of states and transitions to the original graph is an instance of a general output multi-point function that is efficiently obfuscatable.

An interesting and important outcome of the proposed method accompanied by the formal security proof is that the mechanisms for hiding a number of states within the FSM such that only the designer knows about the traversal to/from those states could be used as a basis for a suite of other security protocols. For example, this state hiding can be used for remote authentication or identification by the designer, online integrity checking, and real-time monitoring, controlling, enabling, or disabling the chip. An application of the method for trusted integration of multiple IP cores was demonstrated in [6].

Practical contributions

The paper also makes contributions to practical metering by creating an automatic low-overhead implementation of secure metering during synthesis. The synthesis is performed through three steps: obfuscatable topology construction, secure passkey selection, and iterative synthesis.

In the first step, after *boosting* the original FSM, our method first uses a partition approach, where the low overhead implementation of smaller partitioned FSMs is determined by pre-synthesizing and evaluating various random configurations of small FSMs. The partitions are then combined by randomly added edges to form the added state-space and transitions. Even though our partitioning approach uses similar principles as [2], it is more systematic and guided. The key difference is that our secure construction method constraints the number of directed edges incident to one added state to be t, such that t ensures the graph connectivity and multiplicity of keys.

Selecting strong passkeys is integral to the security of active hardware metering. The strength of random passkeys depends on the entropy of the underlying random number generator. In our case, the selected passkeys can also serve a dual purpose: the designer can use them as proof of ownership in addition to using them for unlocking. For this, we use the hash value of the designer-generated words that are signed by a private key (PrK) of a public key cryptographic (PKC) system to generate passkeys. Now, others with access to the public key (PuK) of the same system can verify the ownership of the designer upon unlocking. However, an eavesdropper who can unlock the chip using a stolen BFSM structure, cannot claim ownership.

To perform the synthesis, we first modify the original FSM adding extra control inputs. These inputs represent the transitions from the BFSM with the correct passkeys. Next, we synthesize each partition of the BFSM with an output representing the incident edge transitions. After synthesizing each component separately, we connect the extra inputs of the original FSM to the outputs of the partitioned FSMs and resynthesize the whole system. This way, if the original design has K FFs and the BFSM has K'' FFs, then $|S| \leq 2^K$ and $|S'| = 2^{K''} - 2^K$ that implies $|S'|$ is much larger than $|S|$.

In addition to the automated synthesis techniques, potential attacks, and security of the presented method against each attack are also addressed in this article. Specifically, we show that the proposed methods are secure against brute force attack, BFSM reverse engineering, PUF removal/tampering attack, and state capture and replay attack.

Evaluation

We first perform the evaluation on the ISCAS benchmark suit. The area, power, and delay overheads are

presented in Table 1. We observe that both area and power overhead are largely independent of the size of the benchmark circuit. Therefore, the percentage overhead is high for smaller benchmarks, but low for the larger ones. Since the circuits in the benchmark set are small compared to the industrial-strength circuitry in design and use today, it is safe to say that the area and power overhead of the metering method is low, in particular since the control path by itself is very small compared to the entire design. Finally, the ratio of the added critical path delay overhead compared to the original delay seems to be independent of the benchmark size, with a mean of 1%, and a standard deviation of 1.15%. Therefore, the overhead in the critical path delay is rather low.

We evaluated the proposed obfuscation method and synthesis techniques on the H.264/MPEG-4 Part 10 or AVC (Advanced Video Coding)—video compression standard widely used in Blu-Ray systems, YouTube, iTunes, and other online video platforms. The earlier work in hardware metering was only evaluated on benchmark simulations. To the best of our knowledge, this article presented the first hardware implementation and overhead evaluation results for hardware metering.

Table 2 shows the design area after synthesis to the FPGA in terms of the number of equivalent gates and the number of occupied lookup tables (LUTs), along with the percentage overhead. The second column reports the number of equivalent gates and the number of LUTs for the original design. The third column shows the overhead for a BFSM with 20 new FFs and a key length of 1024. It can be seen that the percentage of added equivalent gates is about 6.7%, and the percentage of added LUTs is about 13%. We also experimented with two other cases with 40 and 64 added FFs, as shown on the fifth and sixth columns, with key lengths of 2048 and 5120, respectively. As we mentioned earlier, the key length is determined by the number of edge transitions required for unlocking and the passkey on each edge. The passkey on each edge is set to the number of inputs on the edge, $\alpha =$ 64 bits. A key length of 1024 means that we require 16 edge transitions in our unlocking sequence.

We observe that with the increase in the number of added FFs, the percentage overhead in terms of the number of equivalent gates and the number of LUTs increases linearly. Since adding to the number of FFs could yield exponentially stronger proofs for security, our protection method is relatively low overhead for very secure construction. For MPEG4 decoding and many other data-intensive applications that are implemented in hardware for efficiency reasons, the bottleneck is in the datapath optimization and not in the control path which is a much smaller part of the overall design.

Table 1. Metering overhead on the ISCAS benchmark suite.

Circuit	In	Out	FFs	Orig. Area	Added Area	OH (%)	Orig. Power	Added Power	OH (%)	Orig. Delay	OH (%)
s820	18	19	5	769	+1411	186	2773	+5924	213.6	28.2	0
s1196	14	14	18	1009	+1524	151	2558	+8223	321.4	35.8	3
s1238	14	14	18	1041	+1472	141	2709	+7153	264.0	34.4	1
s1423	17	5	74	1164	+1393	120	4883	+5564	114.0	92.4	0
s1488	8	19	6	1387	+1261	91	3859	+2804	72.7	38	1
s1494	8	19	6	1393	+1591	114	3913	+9632	246.1	38.4	2
s5378	35	49	164	4212	+1203	28.6	12459	+1874	15.0	32.2	3
s9234	36	39	211	7971	+1425	17.9	19386	+6312	32.6	75.8	0
s13207	31	121	669	11241	+1571	14	37844	+9334	24.7	85.6	1
s15850	14	87	597	13659	+1281	9.3	40003	+3404	8.5	116	0
s35932	35	320	1728	28269	+1383	4.9	122048	+5279	4.3	299.4	0
s38584	12	278	1452	32910	+1226	3.7	112707	+2357	2.1	94.2	2

Long-term impact

The nominated paper has been very influential in the hardware security research community. The theoretical proofs established in [4] have been demonstrated to be instrumental in shaping the knowledge in both sequential and combinational logic obfuscation and locking. According to Google Scholars, the proposed framework for active hardware metering and its improvements and proofs received 700+ citations cumulatively as of May 2020 [2], [4], [7], [8]. This indicates the importance of the subject.

The proposed theory of obfuscating the FSM such that it is a generalized point function that is provably secure (obfuscatable); this theory has served as the foundation for subsequent research and more secure realizations of sequential and combinational logic locks. Here, we describe how the theoretical

Table 2. Metering overhead for H.264/MPEG4 on FPGA.

	Original H.264	(+20 states) Key: 1024b	(+40 states) Key: 2048b	(+64 states) Key: 5120b
Number of Gates	381,176	407,068	429,075	457,574
Gate overhead	-	6.79%	12.56%	20.04%
Number of LUTs	26,485	29,996	33,160	37,106
LUT overhead	-	13.25%	25.19%	40.09%

grounds set by this article found usage in combinational circuit locking. The landmark paper EPIC introduced the concept of combinational logic locking to the community [9] (~600 citations as of May 2020). Several influential works in combinational logic locking followed since—references to combinational locking are omitted due to space constraints. The papers in this category have suggested inserting the lock and key gates based upon various methodologies, including but not limited to random logic locking (RLL), fault analysis-based logic locking (FLL), and strong logic locking (SLL).

Perhaps the most powerful attacks that broke all prior existing combinational locking techniques are Boolean-Satisfiability (SAT)-based key pruning attack and ML-based attacks [10]. The SAT attack works by eliminating the incorrect keys with the help of distinguishing input patterns (DIPs) [11]. DIPs are input patterns for which two key values lead to differing responses. The assumption is that an Oracle based on a functional IC with the secret key loaded in its memory can identify the incorrect keys iteratively. Two subsequent logic locking methods, namely SARLock and Anti-SAT developed techniques for thwarting SAT attack. Unfortunately, anti-SAT was shown to be vulnerable to signal probability skew analysis (SPS) attack. SARLock performs better in terms of attack resiliency, but practicality and scalability are big challenges for this method since it only can protect one min-term.

A notable recent work in the area of logic locking is called stripped-functionality logic locking (SFLL) [12]. It provides both theory and practice for combinational logic locking while being much more practicable and scalable than SARLock. Interestingly, the theoretical proofs for this kind of obfuscation are built upon the same generalized point functions that our nominated paper introduced to the hardware security community five years earlier. Thus far, SFLL has not been attacked in practical settings, and even recent papers claiming the impossibility of general obfuscation are building proofs based on the original Barak et al. [13] work leveraged in our paper for the first time in this community. (Note that generalized point functions are obfuscatable while not all functions are.)

The reach of the contributions of the nominated paper to the hardware security community is way broader than just logic locking/provable obfuscation. As the citations of the paper show, the methods in the paper can form the basis for addressing several other important hardware security challenges, including but not limited to attacks on obfuscation [11], FPGA IP protection [14], securing against hardware Trojans by obfuscation [15], anti-reverse engineering [16], and successful hiding (obfuscation) of watermarks and fingerprints. Several authors that have cited the nominated paper are well-known authorities in the hardware security community.

Last but not least, the original sequential locking methodology has received a number of awards and honors, including but not limited to DARPA Young Faculty Award, MIT TR-35 Award, and the Presidential Award for Early Career Scientists and Engineers (PECASE). The funding provided via these sources has enabled the author to work on the theoretical proofs for this important topic and make the aforementioned contributions. The most recent program from DARPA called Obfuscated Manufacturing for GPS (OMG), includes several Principle Investigators (PIs) in the hardware security community who are transferring the obfuscation and logic locking methodologies to military and industry. In particular, the ECLIPSE team (PIs: Koushanfar, Makris, Rajendran, Sianoglu) is developing point-function-based sequential and combinational locking for the military test chips including the toolchains and security analysis.

In brief, the nominated paper had a profound impact on the future of IP and IC protection in the present internationally distributed horizontal market with untrusted foundries and continues to inspire novel technologies. It will directly contribute to national security and industrial growth of manufacturing secure and protected chips. The methodology is already included in several hardware security courses taught by the leading researchers and will continue to be a part of a lively education in this important emerging realm.

THE SELECTED TOP picks paper [4] presents the first formal proofs for devising obfuscatable integrated circuits. The proof is based on formulating IC obfuscation as an instance of a general output multi-point function family, known to be effectively obfuscatable in the random oracle model. Two prior fundamental contributions by Koushanfar and her team/collaborators have invented sequential (FSM) [2] and combinational [9] IC obfuscation and locking (also known as metering) concepts. However, the earlier works did not include proof of obfuscatability. The emphasis of

this top picks paper is on the sequential IC metering application, where the control logic, expressed as an FSM, is locked upon fabrication; only the designer (IP rights owner) who had access to the full state encoding of the design can provide the passkey to unlock. The locks are provably obfuscatable within the FSM description of the design. The pertinent article opened novel directions that inspired some of the most distinguished works in the field. For example, follow-up work by other researchers [12] has shown that it is also possible to construct combinational obfuscatability proofs that are resilient against attacks by formulating as an instance of a general output multipoint function family. ■

References

[1] F. Koushanfar, G. Qu, and M. Potkonjak, "Intellectual property metering," in *Proc. Int. Workshop Inf. Hiding (IH)*, 2001, pp. 81–95.

[2] Y. Alkabani and F. Koushanfar, "Active hardware metering for intellectual property protection and security," in *Proc. USENIX Secur. Symp.*, 2007, pp. 291–306.

[3] C. Herder et al., "Physical unclonable functions and applications: A tutorial," *Proc. IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug. 2014.

[4] F. Koushanfar, "Provably secure active IC metering techniques for piracy avoidance and digital rights management," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 1, pp. 51–63, Feb. 2012.

[5] L. Yuan and G. Qu, "Information hiding in finite state machine," in *Proc. Inf. Hiding Conf. (IH)*, 2004, pp. 340–354.

[6] Y. Alkabani and F. Koushanfar, "Active control and digital rights management of integrated circuit IP cores," in *Proc. Int. Conf. Compil., Archit. Synth. Embedded Syst. (CASES)*, 2008, pp. 227–234.

[7] Y. Alkabani, F. Koushanfar, and M. Potkonjak, "Remote activation of ICs for piracy prevention and digital right management," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2007, pp. 674–677.

[8] F. Koushanfar, "Hardware metering: A survey," in *Introduction to Hardware Security and Trust*. New York, NY, USA: Springer, 2012.

[9] J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending piracy of integrated circuits," in *Proc. Design, Autom. Test Eur.*, Mar. 2008, pp. 1069–1074.

[10] H. Chen et al., "GenUnlock: An automated genetic algorithm framework for unlocking logic encryption," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2019.

[11] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, May 2015, pp. 137–143.

[12] M. Yasin et al., "Provably-secure logic locking: From theory to practice," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1601–1618.

[13] B. Barak et al., "On the (im)possibility of obfuscating programs," in *Proc. Int. Cryptol. Conf. (CRYPTO)*, 2001, pp. 1–18.

[14] J. Zhang et al., "A PUF-FSM binding scheme for FPGA IP protection and pay-per-device licensing," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1137–1150, Jun. 2015.

[15] R. S. Chakraborty and S. Bhunia, "Security against hardware trojan attacks using key-based design obfuscation," *J. Electron. Test.*, vol. 27, no. 6, pp. 767–785, Dec. 2011.

[16] M. Yasin et al., "CamoPerturb: Secure IC camouflaging for minterm protection," in *Proc. 35th Int. Conf. Comput.-Aided Design*, Nov. 2016, pp. 1–8.

Farinaz Koushanfar is currently a Professor and a Henry Booker Faculty Scholar of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA, USA. Her research interests include privacy preserving computing, safe and secure AI, embedded systems security, and design automation of domain-specific AI systems. Koushanfar has a PhD in electrical engineering and computer science and an MA in machine learning from the University of California Berkeley, Berkeley, CA, USA. She is a Fellow of IEEE.

■ Direct questions and comments about this article to Farinaz Koushanfar, Electrical and Computer Engineering Department, University of California at San Diego, La Jolla, CA 92161 USA; farinaz@ucsd.edu.