Integrated Circuits Metering for Piracy Protection and Digital Rights Management: An Overview

Farinaz Koushanfar
Electrical and Computer Engineering
Rice University, Houston, TX
farinaz@rice.edu

ABSTRACT

This paper presents an overview of hardware and Integrated Circuits (IC) metering methods. *IC metering* or *hardware metering* refers to tools, methodologies, and protocols that enable post-fabrication tracking of the ICs. Metering enables prevention and detection of overbuilt and counterfeit ICs in the dominant semiconductor contract-foundry model. Post-silicon identification and tagging of the individual ICs fabricated by the same mask is a precursor for metering: In *passive metering*, the ICs are specifically identified, either in terms of their functionality, or by other forms of unique identification. The identified ICs may be matched against their record in a pre-formed database that could reveal unregistered ICs or overbuilt ICs (in case of collisions). In *active metering*, not only the ICs are uniquely identified, but also parts of the chip's functionality can be only accessed, locked (disabled), or unlocked (enabled) by the designer and/or IP rights owners using a high level knowledge of the design not transferred to the foundry. We provide a systematic overview of the field, along with a taxonomy of available methods.

Categories and Subject Descriptors

B [**Hardware**]: GENERAL; B.8 [**Performance and Reliability**]: Miscellaneous

General Terms

Design, Algorithms, Security

Keywords

Integrated circuits metering, Anti-piracy, Hardware security, Counterfeit prevention, Unique IC tracking

1. INTRODUCTION

Aggressive scaling of CMOS to nano-scale feature sizes alongside integration of multiple computing cores within single chips has increased the complexity of the ICs. The advanced chips that are currently in design and production incur expensive, complex, and sophisticated process steps that require very expensive state-of-the-art foundries. The cost of updating and maintaining a fabrication facility with the current technology is reportedly in billions of dollars and increasing, cited as the most costly manufacturing plants built by the mankind [17, 22].

Given the growing cost and complexity, the semiconductor business model has completely transformed to a third party contract fabrication business model (a.k.a the horizontal business model) during the past 30 years. The contract foundry model benefits from the economy of scale, since the same facility serves multiple design companies. Indeed, manufacturing of the ICs designed by the leading edge design houses is outsourced to developing offshore countries with a lower labor overhead and operational cost.

In the new business model, the relationship between the designer and the foundry is *asymmetric*: the designed IP is transparent to the manufacturers who can reproduce (overbuild) the ICs with a negligible overhead because of the ready availability of the masks; but the details of the fabrication process, quantity, and possible modifications to the original designer's chip blueprint (in form of layout files such as OASIS format) are clandestine to the design house. Furthermore, the scale, complexity, and opaqueness of the packaged ICs in the multi-layer VLSI designs and manufacturing process bring upon controllability and observability problems that in turn, make proving the authorship of packaged ICs a challenge. Many of the counterfeit products have not passed the fault tests or are otherwise wrongly packaged, damaging both the brand and the consumer reliability requirements. What exacerbates the problem is the wide-spread usage of ICs in various application with anti-cloning, high protection, and security requirements such as bank cards, embedded access control devices, and weapons. Preventing IC theft, piracy and overbuilding have become increasingly important for the defense, businesses, and consumers because of (i) the criticality of application requirements, and (ii) the losses caused by the counterfeiting, preventing IC theft, piracy, and overbuilding by the contract fabrication facilities.

This paper provides an overview of the field of IC metering. Hardware metering, or IC metering, refers to security methods and protocols that permit the designers (IP rights owner) to have post-silicon control over their designed ICs. The phrase "hardware metering" was first coined in [14, 13] in 2001, to refer to the first passive methods to uniquely identify each IC's functionality within the standard synthe-

GLSVLSI'11, May 2–4, 2011, Lausanne, Switzerland.

sis flow where the same mask is used for fabricating all the chips. Over the last decade, several new and exciting metering methods were proposed for supporting the IP owner's post-manufacturing control over their ICs, including [1, 3, 10, 19, 18, 20]. This paper provides a new classification for the ongoing work in this area.

The remainder of the paper is as follows. The next section presents our new taxonomy and classification of hardware metering. Passive hardware metering is discussed in Section 3. Active hardware metering flow and the related papers are discussed in Section 4. Section 5 concludes the paper.

2. TAXONOMY AND MODELS

As mentioned earlier, metering has been classified into two categories, passive and active. Passive metering provides a way for unique identification of a chip, or its functionality so that it can be passively monitored. Rudimentary passive metering methods have been used for many decades, by physically indenting a serial number of each device, or by storing the identifiers in the permanent memory. We call the former method as *indented serial numbers*, while the second method is called the *digitally stored serial numbers.* Digital serial numbers also provide a way for passive monitoring of the devices. Because of vulnerability of the serial numbers and digital identification numbers to cloning and removal attacks, about a decade ago, the ICID approach introduced methods for generating unclonable IDs based on the inherent random process variations of the silicon [15]. Since the randomness is existing in the process and cannot be controlled or cloned, we refer to this class of identification as *unclonable identification.* Unclonable identifiers (IDs) are a form of Physical Unclonable Functions (PUFs) that were comprehensively discussed and classified in a recent book chapter [21]. A class of PUFs that is able to generate secret keys is *weak PUFs* that can be used as unclonable IDs for IC metering. In contrast to unclonable IDs, we call the earlier known chip identification methods as *reproducible identification.*

Shortly after the introduction of ICID, a fundamentally new way of passive metering was introduced [14, 13]. In this method, the identifiers were linked to the chip's internal functional details during the synthesis step, such that each chip's function would get a unique signature. We refer to this type of passive metering as *functional metering.* Both unclonable and reproducible identifiers can be interfaced to the chip's functionality to make a unique signature for it. Note that the functionality would remain unchanged from the input/output standpoint, and only a set of internal transactions would be unique to each chip.

Most passive metering methods described so far, rely on an added component or change of the design for holding the identifiers or for functional metering. A passive metering method that can uniquely identify each chip without addition of any components or modifications to the design is called *extrinsic.* In contrast, *intrinsic* passive metering methods do not need any added components or design modifications. The big advantage of intrinsic identification methods is that since they do not rely on an added component, they can be readily used on existing designs. The intrinsic identification may be based on digital or analog values (caused by the random physical disorder of the phenomena).

An example for an intrinsic digital identification is a weak PUF based on SRAM, where the existing SRAM memory cells inside the FPGA are used as unclonable IDs [9]. An example for an intrinsic analog ID that is applicable to both ASIC and FPGA, is a novel method for extracting the existing variation signature of the embedded circuits (caused by the inherent process variations) [7, 2]. The extracted signatures can be used as a fingerprint for the device. While the focus of the earlier work was on timing signatures [7], the more recent work has shown that the signatures from the other side-channels, such as IDDQ and IDDT measurements can be used [2]. A unified framework by gate-level translation of the process variation components can be formed so a presentation of the chip's unique signatures in the gate-level domain is possible.

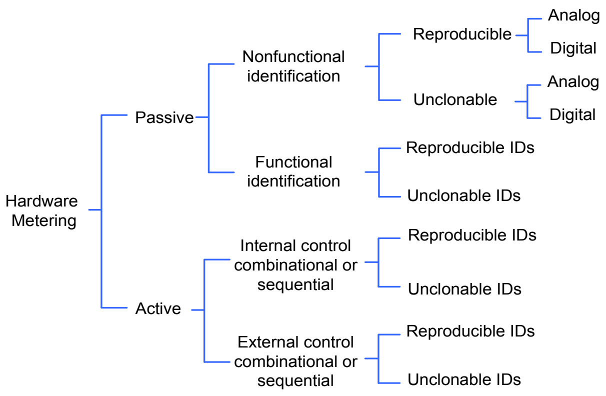

Figure 1: A Taxonomy of metering methods.

Active metering, in addition to unique identification of a device and/or its functionality that may also be remotely monitored, provides an active way for the designer to enable, control, and disable the device. The first known method for active metering of ASIC chips was introduced in [1]. Active metering enriched the realm of functional metering, by hiding states and transitions in the design that can only be accessed by the original designer. Very recent research results on active metering have shown that the original method introduced in [1] can be constructed to be provably secure, by demonstrating a transforming of the provably obfuscatable family of generalized point functions [11]. Since the states and transitions used for controlling (also called locking and unlocking) of the chips are integrated within the functional specification of the design, we refer to this type of metering as *internal active hardware metering.*

Since the introduction of the original active hardware metering in [1], a number of other interesting methods for this purpose have been proposed. Aside from the internal active hardware metering that only exploits design modifications (sequential or combinational) for lock embedding [3], other methods of active metering based on inclusion of external cryptography circuits were introduced [10, 19, 18, 20]. We refer to this type of active metering as *external active hardware metering.* Both internal and external active hardware metering exploit a random identifier in a digital form, that may or may not be unclonable. For example, as the digital random identifier, burned fuses can be used. Note that burned fuses are reproducible at the foundry, and therefore cannot be used as a countermeasure for the foundry effect. However, they may not be reproducible for average customers who may not have an access to the fuses without depackaging and invasively probing the chips.

Figure 1 demonstrates a summary of the taxonomy described in this section. In the remainder of the chapter, we focus on detailed description of the mechanisms and structures that have been proposed for passive and active metering. For passive metering, we put an emphasis on functional identification. A comprehensive treatment of nonfunctional identification of chips is outside the scope of the current paper. We refer the interested readers to a recent comprehensive book chapter that covers this subject [12].

3. PASSIVE IC METERING

A notable progress in the field was the advent of methods for unique functional identification of chips. The first such known method was based on making the control path of each chip unique, so that each chip would have a specific control path. The challenge is fabricating the different chips from the same mask and the same design layout files. The work in [14, 13] proposes designing chips that have a single datapath that can be controlled by multiple versions of the same control path specifications. A small part of the chip is retained programmable, so the control path would be programmed into the chip post-silicon.

A new design method for realizing multiple control paths for one data path was suggested [14, 13]. One solution is to permute the subsets of variables that are assigned to a particular register. To achieve multiplicity, during the logic synthesis step, redundant equivalent states are created for a selected set of states. The selection is based on the existing constraints on the concurrent states that should be stored in separate variables, with the goal of keeping the register overhead very low. Each copy of the variable would obtain a different state assignment, and any permutation of the duplicate assignments could be used for the equivalent states. Since the state assignment is done by graph coloring, creation of redundant states corresponds to adding a vertex to the graph and replicating all the edges of the node to be duplicated for the new vertex. The state assignment for the modified graph can be solved by using the conventional tools. Programmable read logic to the registers enables selecting the correct permutation of the variables for each copy.

3.1 Analysis of Passive Metering

The passive metering protocol for detection of the unauthorized chips is to monitor and evaluate the chips in use. Before testing an authorized chip, the programmable part is loaded with a specific permutation of the control path. Now, if more than one copy of a single permutation is detected, a counterfeit component is flagged. This protocol would work well if many of the chips are online and can be queried for their permutation version. One way to realize online querying is by XORing the states of the FFs, or by performing other variants of parity checking on the system.

One interesting scenario for passive metering is where the chips are returned unprogrammed to the designer who would enter the controller specifications before testing the chips. The IP rights owner would ensure that each of the chips are uniquely programmed and that the foundry is not involved in the programming step. However, this approach by itself does not strongly deter the attackers, since an adversary with access to the chips can replicate the programmable memory's content and use the information to configure other chips. The work also suggests integrating the programmable part with the unclonable IDs coming from the chip using the logic functions, e.g., XOR. At the time of inception of this paper in 2000, the only known unclonable identifiers where the ICIDs [15]. Therefore, the data for the programmable part cannot be replicated on other chips, naturally defending against the overbuilding attacks.

The evaluation results in [14, 13] demonstrate that it is possible to obtain multiple permutations and selection with a very low overhead. An obvious drawback of the presented passive metering approach is the overhead of adding the programmable part to ASICs, as this would require extra mask steps, incurring an additional cost. Furthermore, two probabilistic analysis are presented: (i) the first set of analysis answers the question of how many experiments should be conducted before one can conclude the absence of unauthorized parts with a certain level of confidence; and (ii) the second set of analysis aims at estimating the number of unauthorized copies made, in case duplicate chips are detected on the market.

(i) Assume that the design house demands the foundry to fabricate n copies, but the foundry indeed fabricates N chips where $N >> n$. If the company makes $k-1$ copies of each, the total number of available ICs from the design would be: $N = k.n$. Note that it is proven that the foundry has the best chance of not getting detected by fabricating equal number of copies of each chip. If we draw l from the N objects consisting of k copies of distinct designs, the probability of no duplicate would be:

$$Prob[n,k,l] = [1-\frac{k-1}{N-1}].[1-\frac{2(k-1)}{N-2}]\ldots[1-\frac{(l-1)(k-1)}{N-l-l}], \quad (1)$$

that is upper bounded by:

$$Prob[n,k,l] \leq [1-\frac{p}{n}].[1-\frac{2p}{n}]\ldots[1-\frac{(l-1).p}{n}], \quad (2)$$

where $p = 1-\frac{1}{k}$. As can be seen above, as the k increases, the probability $Prob[n,k,l]$ of not finding unauthorized parts after l random tests (without replacement) decreases. The probability $Prob[n,k,l]$ decreases as the number of tests l, increases. In essence, the quantity $1 - Prob[n,k,l]$ measures the foundry's honesty and it increases as l increases. For a designer to obtain a desired level of confidence α, one need to find the smallest l such that $(1 - Prob[n,k,l]) \geq \alpha$. Since finding an exact closed form formula for Equation 1 is challenging, the solution is often numerically found or by using approximations in case of large n.

(ii) Assuming that k is uniformly distributed, one can immediately find the probability that the first unauthorized copy is found at the $l+1$-th test as:

$$Prob[n,k,l+1] = Prob[n,k,l].\frac{l.(l-1).(k-1)}{N-1}. \quad (3)$$

The authors concluded that the expected number of tests to find the first unauthorized copy is:

$$\sum_{k=1}^{\infty}\sum_{l=1}^{n(k-1)+1} l.Prob[n,k,l], \quad (4)$$

and if the first failure occurs at l, then the expectation for k is:

$$E[k] = \sum_{k=1}^{\infty} k.Prob[n,k,l]. \quad (5)$$

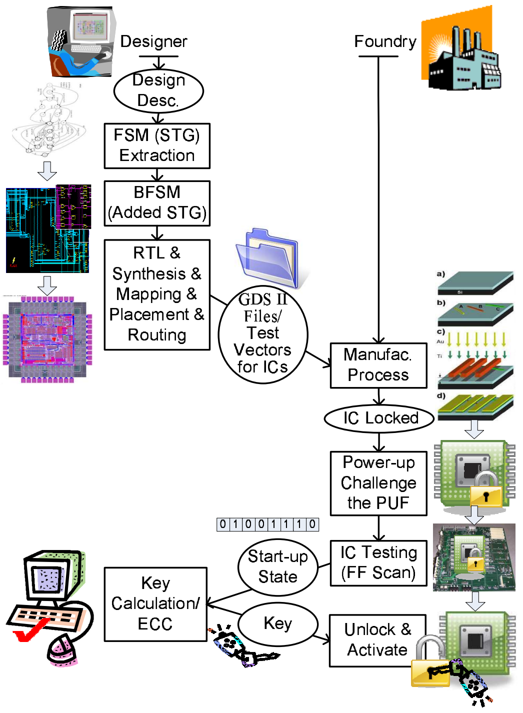

Figure 2: The global flow of the IC enabling by active metering.

4. ACTIVE IC METERING

Active hardware metering not only uniquely and unclonably identifies each chip, but also provides an active mechanism to control, monitor, lock, or unlock the ICs. To ensure irreproducibility, active metering requires a form of unclonable digital IC identifier such as a weak PUF [21]. One of the first presented applications of metering is for designer's IC enabling. Figure 2 demonstrates the global flow of the first known active hardware metering approach for enabling that was described in [1]. Similar IC enabling flows were later adopted for both internal and external active integrated circuits metering. There are typically two main entities involved: (i) a design house (a.k.a designer) that holds the IP rights for the manufactured ICs, and (ii) a foundry (a.k.a fab) that manufactures the designed ICs.

The steps of the flow are as follows. The designer uses the high level design description to identify the best places to insert a lock. The subsequent design phases (e.g., RTL, synthesis, mapping, layout and pin placement) take their routine courses. The foundry would receive the blueprint of the chip in form of OASIS files (or GDS-II) along with other required information for fabricating the chips including the test vectors. The design house typically pays the foundry an upfront cost for a mask to be lithographed from the submitted OASIS files and for the required number of defect-free ICs to be fabricated. Each IC typically contains an unclonable digital identifying unit, such as a weak PUF.

Building a mask is a costly and complex process, involving multiple fine steps that should be closely controlled [17, 22]. Once the foundry lithographs a mask, multiple ICs would be fabricated from this mask. Because of the specific PUF responses integrated within the locks on the chips, each IC would be uniquely locked (nonfunctional) upon fabrication. During a start-up test phase, the fab scans the unique identifier information out of each IC and sends the content back to the design house. The design house who uses the designer-specific knowledge or an asymmetric cryptography protocol, is the only entity who could compute the unlocking sequence for each locked chip. Additionally, the designer could compute the error correcting code (ECC) to correct for any further changes to the unclonable digital identifiers. The ECC is very important since a few of PUF response bits may be unstable and change at a later time because of noise, environmental conditions (e.g., temperature), or circuit instability. The key for unlocking the chip and the ECC would then be sent back to the fab.

The work in [1, 3] have also discussed methods such that the designer's asymmetric information about parts of the design could be utilized for other control purposes, including but not limited to online enabling/disabling and continuous authentication.

4.1 Internal (Integrated) Active IC Metering

The first set of introduced methods for metering were internal [1]. The active IC control mechanism in this class of work leverages: (i) the functional description of the design, and (ii) unique and unclonable IC identifiers. The locks are embedded within the structure of the common computation model in hardware design, in form of a finite state machine (FSM). The designer exploits the high level design description to form the design's behavioral model in the FSM format. FSM is typically represented by the State Transition Graph (STG) where the vertices on the graph correspond to the states in the FSM, and the transitions between the FSM states are represented by the directed edges incident to the vertices. In the remainder of this paper, we use the terms FSM and STG interchangeably. Let us describe the approach in [1]. We call the design's FSM before modifications the original FSM that has $|S|$ states. Therefore, the original can be implemented using $K = log|S|$ FFs.

Now assume that we modify the FSM by augmenting to its states and transitions. We call the modified design a boosted finite state machine (BFSM). To build a BFSM with $|S'| + |S|$ states, we would require $K" = log\{|S'| + |S|\}$ FFs. Additional edges are also introduced to the BFSM to ensure the reachability of its states. Observe that for a linear growth in the number of FFs denoted by $K' = K" - K$, the number of states exponentially increases. Indeed, by adding a number of FFs and tolerating the overhead of this addition, it is possible to set $S' >> S$ so that the number of new states are exponentially many more than $|S|$.

The IC also contains a PUF unit that generates random bits based on the unclonable process variations of the silicon that are unique on 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.

Upon the IC's power up, the initial values of the design's FFs (i.e., *power-up state*) is determined by the unique response from the PUF on each chip. The PUF challenges are determined by fixed test vectors given by the designer. For a secure PUF design, the probability of the response should be uniformly distributed over the possible range of values [8]. The number of added FFs can be set such that the value $2^{K"} >> 2^{K}$. In other words, the value $K"$ is set by the designer such that for a uniform probability of selecting the state, the probability of selecting a state in the original FSM is extremely low.

Because there are exponentially many added states, there is a high probability that the unique PUF response on each chip sets the initial power up state to one of the added states. Note that unless the design is in one of the original states, it would be nonfunctional. 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 so it can transition from this nonfunctional initial power-up state to the functional *reset state* of the original FSM shown by double circle on the example.

For the IP rights owners who have access to the BFSM state transition graph, finding the set of inputs for traversing from the initial power-up state to the reset-state (shown by double circle on the figure) is easy. All what is needed is to find a path on the graph and use the input values corresponding to the path transition (from the STG description) so the states transition to the reset state. However, there is only one combination from exponentially many possibilities for the input for each edge transition. Thus, it would be extremely hard for anybody without access to the BFSM edge transition keys to find the exact inputs that cause traversal to the original reset states.

The access to the full BFSM structure and the transition function on its edges is what defines the designer's secret. The passkey for unlocking the chip is the sequence of inputs that can traverse the state of the BFSM (describing the control component of the chip) from the initial random power-up state to the original state. Note that although the initial power-up state is random, the assumption is that for a given PUF input (challenge) the response remains constant over time for one chip. 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.

Reference [11] provides the first comprehensive set of proofs and a secure construction of the outlined internal active metering. The author shows the construction of locks by finite state manipulation and compilation during the hardware synthesis for interfacing to a unique PUF state, is an instance of an efficiently obfuscatable program under the random oracle model [4]. Even though heuristic methods for FSM obfuscation were proposed earlier, e.g., [23, 6], no provable security for such a construction has been available. The significance of the proposed construction and security proofs for the obfuscated FSM goes beyond hardware metering and extends to most previous work in information hiding and obfuscation of sequential circuits [23, 6]. A detailed description and security proofs for this work is outside the scope of this paper. The method has been shown to be resilient against a spectrum of proposed attacks [11].

Another internal hardware metering method based on FSM modifications was proposed in [3]. We emphasize that the method in [3] is drastically different from the one described above, since only a few states would be added. However, many more transitions (implemented by combinational logic) are added such that the transitions in between the states are functions of the unique and unclonable chip identifiers. It is interesting to note that this hardware metering method can be also used in the context of third party IP integration, where each of the IP cores on a chip can be enabled, disabled, or otherwise controlled.

4.2 External Active IC Metering

External active IC metering methods lock every IC by asymmetric cryptographic techniques that require a specific external key. The use of asymmetric cryptography for external IC metering was first proposed in EPIC [19]. Since EPIC has been a basis for most of the subsequent work in external active metering, we discuss this methodology in detail.

To support Public Key Cryptography (PKC), the IP rights holder must generate a pair of master keys (MKs) (public and private) that will remain unaltered. The private master key (MK-Pri) embodies IP rights for a given design and is never transmitted. Each fabricated chip produces its own random public and private key pairs upon start-up. Also embedded in the register transfer level (RTL) are the public master key (MK-Pub) and the minimal circuitry to support the EPIC's combinational locking mechanism.

EPIC implements combinational locking in the chip's main modules by adding XOR gates on a number of selected noncritical wires, with added control signals connected to the common key (CK) register. When the correct CK is present, the circuit is equivalent to the original; otherwise, the chip's behavior is changed, as if stray inverters were located on selected wires. EPIC produces the CK at random to prevent it from being stolen earlier. After modifying the placed design, the designer securely communicates the CK to the IP rights holder and erases all other copies. Routing and other physical optimizations then proceed as normal, followed by manufacturing. Upon manufacturing, each IC will be uniquely locked because of the interface with the random and unclonable IDs generated by the IC.

While activating a chip, the foundry must have a secure link with the designer (IP rights holder) and must send the RCK-Pub for the IC to be activated. EPIC's protocol uses the foundry's private key to authenticate the transmission. Extensions to this protocol may send a time stamp, serial number, or other identifying sequences. In response, the designer (IP rights holder) transmits the input key (IK) that shows the CK encrypted with the PCK-Pub and signed by the MK-Pri afterwards. The ordering of encryption and signing of the CK for generating the IK is crucial so that entities other than the designer (IP rights holder) cannot produce IKs, even if the CK is compromised. Using the RCK-Pub to encrypt the messages makes statistical attacks against the MK-Pri more complex. The designer can use the foundry's public key to additionally encrypt the resulting IK so that only the manufacturer can receive it. The IC decrypts the IK using the RCK-Pri and MK-Pub that authenticate it as being sent by the designer. Upon decryption, a CK is generated that unlocks the chip and facilitates testing. After the testing step, the IC can be sold.

EPIC is shown to be resilient against a number of proposed attacks, as described in [20]. Note that an early version of EPIC was evaluated by other groups in terms of security and overhead [16]. They found that EPIC is vulnerable if the IK is calculated from the CK, MK-Pri, and RCK-Pub in the wrong order; the CK must first be encrypted with the PCK-Pub and then the resultant ciphertext must be signed by the MK-Pri that is a standard protocol for public-key communication with nonrepudiation. On the other hand, if the IK is computed properly, no successful logic-level attacks against EPIC are known. These issues were discussed and addressed in the latest version of EPIC [20].

Note that the work in [18] presents an external IC locking method built upon secret sharing. The chip and the design plant share a secret key that is interfaced with the combinational logic on the circuit and is used for locking and controlling of the buses that are used to connect and interface the multiple cores on one chip. Another variant of hardware metering by asymmetric key cryptography was discussed in [10]. The work in [5] introduces another combinational locking method that like [14, 13] utilizes a small programmable part within the chip, that is referred to by *reconfigurable logic barriers*. As mentioned earlier, the advantage of such programmable parts is keeping a part of the design to the IP rights holder. The drawback is the added process and mask overhead incurred for implementing the programmable components within ASIC. Because of space constraints, we refer the interested readers to the specific papers for more details for the other proposed external metering methods [10, 19, 18, 20], and to a recent survey [12].

5. CONCLUSIONS

This paper provided a comprehensive overview of hardware integrated circuits (IC) protection by metering. IC metering refers to mechanisms, methods and protocols that enable tracking of the chips post-silicon. IC metering is motivated by the increased rate of outsourcing of leading-edge chip fabrication to offshore foundries that creates opportunities for overbuilding and piracy. IC metering helps the designers to identify and/or track their designs post-fabrication. In our new taxonomy, hardware metering was divided into two categories: passive and active. Passive metering either assigns an identifier to the chip (maybe reproducible or unclonable), or it assigns a signature to the internals of an IC, while the chip maintains its functionality and integrity. In *active metering*, not only the ICs are uniquely identified, but also parts of the chip's functionality can be only accessed, locked (disabled), or unlocked (enabled) by the designer. We discussed both internal and external active hardware metering. Overall, the hope is that by limiting the opportunities for piracy and theft of ICs using post-silicon control mechanisms and protocols, hardware metering would be able to directly and significantly improve the business and practice of semiconductor design and manufacturing.

6. REFERENCES

[1] Y. Alkabani and F. Koushanfar. Active hardware metering for intellectual property protection and security. In *USENIX Security Symp.*, pages 291–306, 2007.

[2] Y. Alkabani, F. Koushanfar, N. Kiyavash, and M. Potkonjak. Trusted integrated circuits: A nondestructive hidden characteristics extraction approach. In *IH*, pages 102–117, 2008.

[3] Y. Alkabani, F. Koushanfar, and M. Potkonjak. Remote activation of ICs for piracy prevention and digital right management. In *ICCAD*, pages 674–677, 2007.

[4] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In *CRYPTO*, pages 1–18, 2001.

[5] A. Baumgarten, A. Tyagi, and J. Zambreno. Preventing IC piracy using reconfigurable logic barriers. *IEEE Design and Test of Computers*, 27:66–75, 2010.

[6] R. Chakraborty and S. Bhunia. Hardware protection and authentication through netlist level obfuscation. In *ICCAD*, pages 674–677, 2008.

[7] S. Devadas and B. Gassend. Authentication of integrated circuits. US Patent 7,840,803, 2010. Application in 2002.

[8] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Silicon physical random functions. In *CCS*, pages 148–160, 2002.

[9] D. Holcomb, W. Burleson, and K. Fu. Power-up SRAM state as an identifying fingerprint and source of true random numbers. *IEEE Transactions on Computers*, 58(9):1198–1210, September 2009.

[10] J. Huang and J. Lach. IC activation and user authentication for security-sensitive systems. In *HOST*, pages 76–80, 2008.

[11] F. Koushanfar. Active integrated circuits metering techniques for piracy avoidance and digital rights management. Technical Report TREE1101, ECE Dept., Rice University, 2011.

[12] F. Koushanfar. *Book Chapter in Introduction to Hardware Security and Trust, M. Tehranipoor and C. Wang editors*, chapter Hardware metering: A survey. Springer, 2011.

[13] F. Koushanfar and G. Qu. Hardware metering. In *Design Automation Conference*, DAC, pages 490–493, 2001.

[14] F. Koushanfar, G. Qu, and M. Potkonjak. Intellectual property metering. In *IH)*, pages 81–95, 2001.

[15] K. Lofstrom, W. R. Daasch, and D. Taylor. Ic identification circuit using device mismatch. In *ISSCC*, pages 372–373, 2000.

[16] R. Maes, D. Schellekens, P. Tuyls, and I. Verbauwhede. Analysis and design of active *IC* metering schemes. In *HOST*, pages 74–81, 2009.

[17] C. Mouli and W. Carriker. Future fab: How software is helping intel go nano–and beyond. *IEEE Spectrum*, March 2007.

[18] J. Roy, F. Koushanfar, and I. Markov. Protecting bus-based hardware ip by secret sharing. In *DAC*, pages 846–851, 2008.

[19] J. Roy, F. Koushanfar, and I. Markov. EPIC: Ending piracy of integrated circuits. In *DATE*, pages 1069–1074, 2008.

[20] J. Roy, F. Koushanfar, and I. Markov. Ending piracy of integrated circuits. *IEEE Computer*, 43:30–38, 2010.

[21] U. Rührmair, S. Devadas, and F. Koushanfar. *Book Chapter in Introduction to Hardware Security and Trust, M. Tehranipoor and C. Wang editors*, chapter Security based on Physical Unclonability and Disorder. Springer, 2011.

[22] B. Santo. Plans for next-gen chips imperiled. *IEEE Spectrum*, August 2007.

[23] L. Yuan and G. Qu. Information hiding in finite state machine. In *IH*, pages 340–354, 2004.