Chapter 1
Hardware Metering: A Survey

Farinaz Koushanfar

Abstract This chapter provides the first comprehensive overview of hardware integrated circuits (IC) protection by metering. Hardware metering, or IC metering refers to mechanisms, methods, and protocols that enable tracking of the ICs post-fabrication. Metering is particularly needed in the horizontal semiconductor business model where the design houses outsource their fabrication to (mostly offshore) contract foundries to mitigate the manufacturing and labor costs. The designers and/or the design intellectual property (IP) holders are vulnerable to piracy and overbuilding attacks due to the transparency of their designed IP to the foundry that requires a complete description of the design components and layout to fabricate the chips. Because of the prevalence of counterfeit and overbuilt items, and the widespread usage of ICs in a variety of important applications, the problem has recently gained an increased attention by the industry, government, and research community. Post-silicon identification and tagging of the individual ICs fabricated by the same mask is a precursor for metering: In *passive metering*, each ICs is specifically identified, either in terms of its 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 unclocked (enabled) by the designer and/or IP rights owners with a high level knowledge of the design that is not transferred to the foundry. We provide a systematic view of the field, along with the first detailed taxonomy and descriptions of the various passive and active hardware metering methods available.

Electrical and Computer Engineering Department, Rice University

1.1 Introduction

As the integrated circuits scale in feature sizes and exponentially grow in functionality, they also become increasingly complex. The sophisticated chips in design and fabrication today require costly, intricate, and complex processes that can only be performed in state-of-art fabrication facilities. Building or maintaining such facilities for the present CMOS technology is reported to be more than 3 Billion dollars and growing, described as the most expensive factories ever built by humankind [30, 40]. Given this increasingly glowing cost and complexity of foundries and their processes, the semiconductor business model has largely shifted to a contract foundry business model (a.k.a. horizontal business model) over the past two decades. For example, Texas Instruments (TI) and Advanced Micro Devices (AMD), two chip making giants that have traditionally used their in-house facilities for fabricating their chips, have both recently announced outsourcing most of their sub-45nm fabrication to major contract foundries worldwide.

In the horizontal 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. The existing business model and contract agreements are insufficient for full protection of the designer IP rights [1, 2]. The IP owners disclose the details of their IP and also pay for building costly masks based on their designs; they trust the foundry not to pirate their designs and not to overbuild more ICs. The mask's ready availability at the foundry, relative low cost of silicon, and lack of designer's (IP rights owner's) control over the foundry fabrication further ease piracy. The internal of the manufactured ICs are opaque due to the design's multiple layers and its complexity; interface circuitry that only allows limited external access to certain internal chip components; and the packaging. What exacerbates the problem is that the ICs are exploited for multiple applications with strict anti-cloning and security requirements, including but not limited to bank cards, security devices, and weapons. Given the criticality of applications and designer's huge losses to piracy, preventing IC theft, piracy, and overbuilding by the contract foundries have become increasingly crucial from the government, industry, business and consumer point of views.

IC metering is a set of security protocols that enable the design house to achieve post-fabrication control over their ICs. The term "hardware metering" was first coined in 2001 [26, 25] to refer to the first passive method for uniquely tagging each IC's functionality while maintaining the same input/output behavior and synthesis flow. Around the same time and independently, methods for unclonable identification of ICs including physical unclonable functions (PUFs) have been in development [28, 14]. Since then, multiple newer methods for giving a design house (IP rights over) control over their designs have been proposed for metering. Note that the levels of post-fabrication protection and control are dependent on the attack model, the desired protection level, and the security assumptions for the IC supply chain.

The focus of this chapter is on hardware metering. It is important to emphasize the difference between hardware watermarking and hardware metering. While metering attempts to uniquely tag each chip produced from a certain design by active or passive methods to facilitate tracing the chips, watermarking's objective is to uniquely tag the design itself so that all the chips produced by the same design/mask would carry the same watermark. Therefore, watermarking is not a good countermeasure against overbuilding, since it cannot distinguish between the different chips fabricated by similar masks. Hardware metering, provides a way to uniquely fingerprint or tag each chip and/or each chip's functionality, so it is possible to distinguish between the different chips manufactured by the same mask. A full coverage of watermarking is outside the scope of this chapter. For a more comprehensive coverage, we refer the interested readers to an excellent book [35], and several important papers on this topic [19, 27, 42, 32, 23, 21].

The remainder of the chapter is organized in the following way. Chapter 1.2 outlines a new taxonomy for the work in metering, beyond the traditional well-known passive and active classifications. Chapter 1.3 covers the work in passive metering including nonfunctional reproducible metering, nonfunctional unclonable metering, and functional metering. In Chapter 1.4, we outline and discuss the research in active metering which covers both the methods based on combinational and sequential locking and control embedded within the design, and also the methods that require external cryptography module(s) in addition to the combinational/sequential locking. Lastly, Chapter 1.5 concludes our metering discussions.

1.2 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 for specifically tagging an IC's functionality so that it can be passively monitored. Rudimentary passive metering methods have been used for many decades, by physically indenting a serial number on 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*. Both methods can be classified as *nonfunctional identification* methods, as the unique ID is separate from the chip's functionality. Since serial numbers (both indented and digital) can be easily copied and placed on new chips, we subclassify both methods to the *reproducible nonfunctional identification* category.

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 [28]. Since the randomness is existing in the process and cannot be controlled or cloned, we refer to this class of identification as *unclonable identification* or *intrinsic fingerprint extraction*. Unclonable identifiers (IDs) are a form of Physical Unclonable Functions (PUFs) that were comprehensively discussed and classified in Chapter 7 of this book. According to the classification provided in the referred chapter *weak PUFs* (a class of PUFs that is able to generate secret keys) can be used as the unclonable IDs for IC metering. We subclassify such chip identification methods based on the inherent process variation as *unclonable nonfunctional identification*.

Shortly after the introduction of ICID, a fundamentally new way of passive metering was introduced [26, 25]. 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 changing the design for holding the identifiers, or pre-synthesis modifications for functional metering. A passive metering method that can uniquely identify each chip with addition of components or modifications to the design is called *extrinsic*. In contrast, an *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 legacy designs. The intrinsic identification may be based on digital or analog values that are caused by the random physical disorder of the phenomena, in this case the inherent silicon process variations.

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 [17]. An example for an intrinsic analog ID that is applicable to both ASIC and FPGA, is a nondestructive method for extracting the existing variation signatures of the embedded circuits (caused by the inherent process variations) [13, 8]. The extracted signatures can be used as a fingerprint for each device. While the focus of the earlier work was on timing signatures [13], the more recent work has shown that the signatures from the other side-channels, such as IDDQ and IDDT measurements can be also used for intrinsic chip identification [8]. A unified framework by gate-level translation of the process variation components was formed and therefore, a presentation of the chip's unique signatures in the gate-level domain became possible.

Active metering, in addition to unique identification or remote passive monitoring of a device, provides an active way for the designers to enable, control, and disable the device. The term active metering was first coined in [6] where the authors introduced the first known method for actively controlling the ASIC chips.

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 has shown that the original method introduced in [6] can be constructed to be provably secure by showing a transformation of the provably obfuscatable family of generalized point functions [20]. 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 [6], a number of other interesting methods for this purpose have been proposed. Aside from the internal active hardware metering methods that only design modifications (sequential or combinational) for lock embedding [9], other methods of active metering based on inclusion of external cryptography circuits were introduced [18, 37, 36, 38]. 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, they cannot be used as a countermeasure for the foundry piracy attack. However, they may not be reproducible by average customers who may not have access to the fuses without depackaging and invasively probing the chips.

Figure 1.1 demonstrates a summary of the taxonomy described in this section. Notice on the chart the distinction between the analog and digital identification for both reproducible and unclonable IDs. Although both analog and digital identification are possible, the existing methods for passive metering based on functional identification and active metering (internal and external) all use the digital identifiers. This is because the digital IDs can be readily integrated within the logical flow. 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.

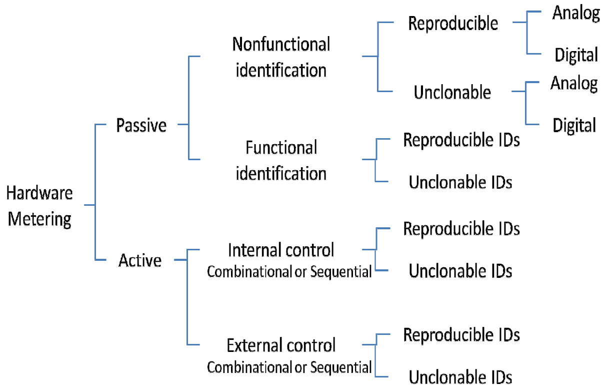


Fig. 1.1 A Taxonomy of metering methods.

1.3 Passive IC Metering

In this section, we separately review the work in nonfunctional and functional metering and fingerprinting of circuits.

1.3.1 Passive Metering by Nonfunctional Identification

1.3.1.1 Reproducible Identifiers

There is no clear record of when the IC companies have started to indent IDs on the packages, or to have a separate piece of ID set aside for storing a digital identifier for their devices. Also, there is no clear public record of when/if the IC companies have used the digital IDs to monitor their devices in hands of the users. Also, burn-in fuses are used at the design houses for carving identifiers on the chips for identification purposes.

Perhaps the best known (and controversial) incident is the Intel Pentium III processors that were publicly announced to a include a unique identifier, called the *Processor Serial Number (PSN)*. The PSN could be used to monitor the user activities via the networks. Although Intel made a utility that would give the control over enabling/disabling the PSN to the device owners, it was demonstrated that rogue web sites were able to access even the disabled PSN. In 1999, several consumer privacy groups jointly filed a complaint against Intel with the Federal Trade Commission. After much debate over the privacy concerns, Intel quietly decided that the next generation of its processor, starting from the Williamette family, would not include the PSN. Pentium-III processors were not recalled from the market and they still have the PSN installed [4, 3].

As mentioned earlier, the drawback of indented IDs, digitally stored IDs, and burn-in fuses is that they can be tampered with, removed, or reproduced. As an example, a plausible and documented attack is repackaging of the older technology as a new one. Digitally stored IDs and serial numbers can be often read by noninvasive probing, and can be rewritten with a small effort. Also none of the available reproducible identification methods are able to withstand the foundry attack where the products are overbuilt. The foundry can simply reproduce an ID by writing to the memory, or by reproducing the numbers on the package.

1.3.1.2 Unclonable Identifiers

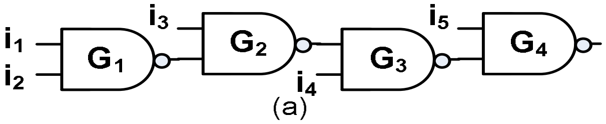
To overcome the limitations of the digital ID storage, methods for exploiting the random process variations in silicon to generate a random unique identifying (fingerprint) have been proposed. If exploiting the random process variations is done by introducing a new circuit structure, this is an instance of an extrinsic identification. If no new circuitry is introduced and only the existing on chip components are used for identification, the method would be an instance of an intrinsic identification. Note that the extrinsic/intrinsic distinction also holds for FPGAs: if the contents of the FPGA (without reconfiguring the cells) is used, then the identifiers are intrinsic. Any use of reconfiguration of the devices would render the identification method an extrinsic one.

Extrinsic methods. The earliest known instance of this class of work was ICID [28], where an an array of addressable transistors with common gate and source with sequentially selected drains was driving a resistive load. The transistors incurred threshold mismatches as a result of process variations. Therefore, the drain currents were randomly different, generating a sequence of unique random numbers on the transistors. Several work in more sophisticated identification and authentication circuit methods has followed this early work, including the physical unclonable functions that were later introduced [13, 14]. Chapter 7 of this book has a thorough discussion on PUFs and on other unique objects based on the random physical disorder, including a systematic and new classification of the various work in this area.

In this chapter, we limit the discussion on the other methods that can be used for passive and active metering. It is also possible to make a distinction between analog and digital extrinsic identifiers. For example, ICID generates random bits that are in digital form, and can be potentially readily included in a digital computation setting. It is plausible to design circuitry whose analog output depends on the underlying silicon process variations. Since for many applications it is desired to have digital outputs that can be readily integrated within the remainder of the digital design, most of the extrinsic ID extraction circuitry have a digital response.

Intrinsic methods. The work in [13] has also proposed using the ICs' timing path signatures that are unique for each state-of-the-art CMOS chip (because of process variations) as a PUF. The interesting studies and new results in [8, 33] have shown that unique identification is possible for almost all ICs designed in state-of-the-art technology, as long as external test vectors can be applied and structural (side channel) tests such as leakage, timing, and dynamic power measurements can be performed. This feature, allows identification of all legacy ICs designed in technologies that contain manufacturing variability, without adding any components to the chip. The requirement is that test measurement equipment for performing the structural tests is available and can be used.

The proposed method in this latter work was based on the gate-level characterization of each IC that could be structurally tested. Since the characterization is done nondestructively, there is no need for an additional circuitry. The extracted characteristics were continuous values, and multiple ways for discretizing and using the characteristics as unique chip IDs were suggested. Such nondestructive identification of legacy ICs can provide many opportunities for creation of new IC metering and security protocols. The other interesting note about using the structural tests is that for a given number of gates and their interconnections, there are many more test vectors, allowing one to generate an exponentially larger space of challenge-response pairs for one circuit.



(a)

NAND2	
Input	Leakage (nA)
00	37.84
01	100.3
10	95.7
11	454.5

(b)

Gate	Scaling factor	
	IC 1	IC 2
G_1	0.5	2.4
G_2	1.3	0.6
G_3	2.1	4
G_4	3	0.9

(c)

Input vector	Leakage (nA)	
	IC 1	IC 2
00011	1391	2055
10101	2082	1063
01110	1243	2150
11001	1841	1905

(d)

Fig. 1.2 (a) A design consisting of 4 NAND2 gates, (b) leakage current vs. input for NAND2,(c) scaling factors of gates on two ICs, and (d) total leakages of ICs for different input vectors (source [8]).

Perhaps the best way to review the hidden characteristic extraction approach is by showing an example (figure from [8]). In Figure 1.2(a) a small circuit consisting of four 2-input NAND gates. Table 1.2(b) shows the static (leakage) current of a nominal NAND2 for different possible inputs. However, because of process variations, the leakage current greatly varies from one chip to another. Table 1.2(c) shows scaling factors for the 4 gates in two example chips, and finally, Figure 1.2(d) demonstrates the total leakage current for the two chips respectively. Therefore, measurements from the total leakage over the different inputs are linearly decomposed to their gate-level components. Similar methods can be used for decomposing the timing measured for multiple test vectors and paths, and for separating the total dynamic current measured at the gate level.

Such multi-modal gate level characterization methods have been further improved, not just for IC fingerprinting applications, but also for accurate post-silicon characterization [22, 41] and for Trojan (malware) detection [34, 31, 7, 43, 24]. Another example of intrinsic identification methods are SRAM PUFs that are covered in detail in Chapter 7.

Aside from the work focusing on logic-level characteristic extraction of chips, later research in this area has shown that other forms of circuit parasitics, including the unique resistances on each chip is able to generate a unique response to the measurements [15, 16].

1.3.2 Passive Functional 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 internal control sequence. Despite the internal differences, the input and output behavior of all the chips coming from a single design and mask are the same. The challenge is fabricating the different chips from the same mask and the same design layout files. The work in [26, 25] proposed 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.

Subsequently, a new design methodology for realizing multiple control paths for one data path was suggested [26, 25]. For example, one solution was 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 overheads 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 would correspond 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 graph coloring tools. Programmable read logic to the registers enables selecting the correct permutation of the variables for each unique copy of the control sequence.

1.3.2.1 Analysis of Passive Functional Metering

The passive metering protocol for detection of the unauthorized chips is to monitor and evaluate the chips while they are 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 of the internal control structure. One way to realize online querying is by XORing the states of the FFs to generate a checksum of the states, or by performing other variants of integrity checking on the states.

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 one unlocked chip can replicate the programmable memory's content from one chip and then use the information to configure and enable other chips. To avoid such direct replication attacks, the idea of integrating the programmable part with the unclonable IDs coming from the chip was also suggested. At the time of writing the first passive hardware metering paper in 2000, the only known unclonable identifiers where the ICIDs [28]. Therefore, the data for the programmable part could not be replicated on other chips, naturally defending against the overbuilding attacks.

The evaluation results in [26, 25] demonstrate that it is possible to obtain multiple permutations and selection of the internal ordering of the control sequences 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. Two probabilistic analysis were presented for the original passive metering method: (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. Since these two analysis methods are generalizable to many other metering and unique identification scenarios, in what follows we present the details of the analysis.

(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: $N = kn$. 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:

$$Prob[n,k,l] = [1 - \frac{k-1}{N-1}].[1 - \frac{2(k-1)}{N-2}]\dots[1 - \frac{(l-1)(k-1)}{N-l-l}], \quad (1.1)$$

that is upper bounded by:

$$Prob[n,k,l] \leq [1 - \frac{p}{n}].[1 - \frac{2p}{n}]\dots[1 - \frac{(l-1).p}{n}], \quad (1.2)$$

where $p = 1 - \frac{1}{k}$. As can be seen above, as the value of 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 1.1 is challenging, the solution is often found by numerical estimations 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:

$$Prob[n,k,l+1] = Prob[n,k,l].\frac{l.(l-1).(k-1)}{N-1}. \quad (1.3)$$

The expected number of tests to find the first unauthorized copy would be:

$$\sum_{k=1}^{\inf} \sum_{l=1}^{n(k-1)+1} l.Prob[n,k,l], \quad (1.4)$$

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

$$E[k] = \sum_{k=1}^{\inf} k.Prob[n,k,l]. \quad (1.5)$$

1.4 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 post fabrication. To ensure irreproducibility, active meter-

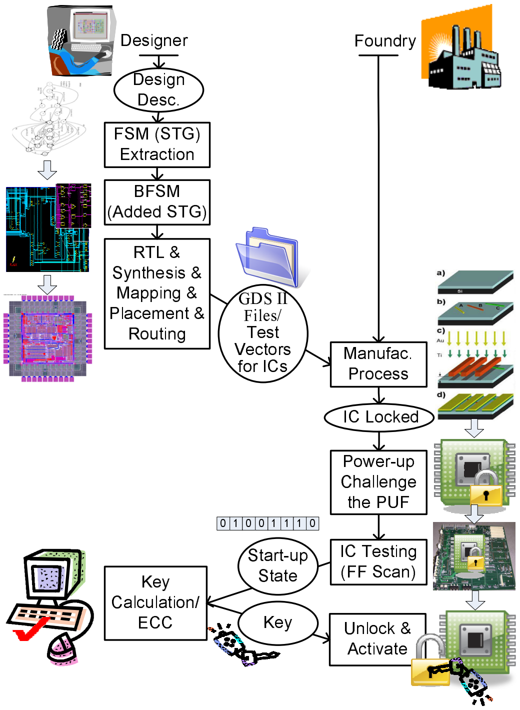


Fig. 1.3 The global flow of the IC enabling by active metering.

ing requires a form of unclonable digital IC identifier such as a weak PUF [39]. One of the first presented applications of metering was for designer's IC enabling. Figure 1.3 demonstrates the global flow of the first known active hardware metering approach for enabling that was described in [6]. 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 [30, 40]. 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 that 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 [6, 9] 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.

1.4.1 Internal (Integrated) Active IC Metering

The first set of introduced methods for metering were internal [6]. 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 designs 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 chapter, we use the terms FSM and STG interchangeably. Let us describe the approach in [6]. We use the term *original FSM* to refer to the design's finite state machine before modifications (with $|S|$ states). Therefore, the original FSM could be implemented using $K = log|S|$ FFs.

Now assume that we modify the original 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

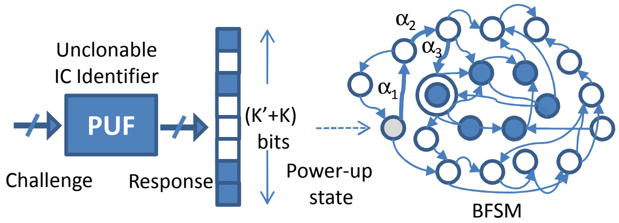


Fig. 1.4 The PUF response is fed to the FFs storing the states of the BSFM. The original states are shown in dark, and the added states are demonstrated in white color on the STG that represents the BFSM.

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. This is shown in the Figure 1.4. 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 [14]. The number of added FFs can be set such that the value $2^{K”} >> 2^{K}$. In other words, the values 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 transitions from this nonfunctional initial power-up state to the functional *reset state* of the original FSM shown by double circle on the example.

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 of each edge transition. Thus, it would be extremely

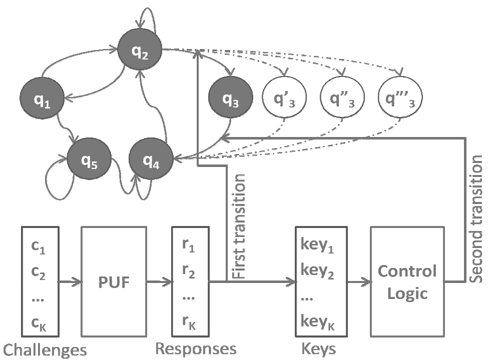


Fig. 1.5 Locking and unlocking by replicating a few states. The PUF response is used for the first transition to one of the redundant states $(q_3, q_3', q"3, q_3''')$. The passkey provides the transition from the pertinent redundant state to the next state.

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 are what define 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.

The recent results in [20] provide 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 and interfacing to a unique PUF state is an instance of an efficiently obfuscatable program under the random oracle model [10]. Even though heuristic methods for FSM obfuscation were proposed earlier, e.g., [44, 12], no provable security for such a constructions was 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 [44, 12]. 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 [20].

Another internal hardware metering method based on FSM modifications was proposed in [9]. The FSM modifications however, were drastically different than those described in [6]. Here, only a few states are added to the original FSM. Indeed, a few states of the FSM are selected for replication. A replicated state q'_i (i.e., a copy of an original state q_i) is an added state to the graph such that all the incident edges from/to the neighbors of q_i are copied to the state q'_i. Different passkeys are associated with the original edges and the replicated edges. The locking is performed such that only one of q_i or q'_i are selected for each chip by the random unique IDs of that chip. The ID can be scanned out from the chip. The provided input for edge transitions to and from the selected state is the one passkey that can perform unlocking.

An example for this operation is demonstrated in Figure 1.5. One the figure, the overall schematic for an FSM with a lock on the replicated state (q_3 on the example) is demonstrated. Three replications of the state q_3, denoted by q'_3, q''_3, and q'''_3 are shown for the sake of the example. The response from the PUF determines the transition edge to one of the redundant states (either q_3 or one of its replicates). The response is also XOR'd with the passkey (computed by the original designer) to provide the transition out of the redundant state. Without the passkey, the transition will not take place or will be incorrect with a very high probability. Note that in real settings, the key length should be properly selected to thwart the brute-force attacks and guarantee security by point function. The significance of this new structure is that the locks are embedded within the internal states that are also visited during the IC's normal operation. Therefore, the locks and keys are continually accessed during the IC's normal operation, creating an opportunity for continuous authentication and self checking.

It is interesting to note that FSM-based 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 [5]. In this approach, a designer or an integrator is the reuser of existing third party IP cores. Two other entities involved in this design and reuse protocol model are the fabrication plant and an authorized system verifier, referred to as a *certificate authority (CA)*. This latter entity a trusted third party who ensures a trust between hardware IP providers, reusers, and the fab.

Let us consider a scenario where multiple third party core blocks are to be protected. Each IP block contains a lock in its FSM structure. The resuer includes also two new modules in the design, the unclonable identification circuitry, and an embedded control module. The embedded control module interacts with and controls the various locks inside each third-party IP block. The blueprint of the design is sent to the CA before sending to fabrication. The CA certifies the IP core providers and the reuser. The post-silicon chips are tested for their unclonable IDs that are sent back to the CA. The CA contacts the third party IP providers and the reuser to get the passkeys for the IP block cores and the embedded control module. Note that similar third party IP protection models can be applied to cases where external active hardware metering is used for locking each core.

1.4.2 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 by EPIC [37]. 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, the 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 [38]. Note that an early version of EPIC was evaluated by other researchers in terms of security and overhead [29]. 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 fully addressed in the latest version of EPIC [38].

[36] presented 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 can be used to connect and interface the multiple cores on one chip.

[18] proposed an active IC metering approach that shared many of the ideas, protocols, and methods developed by EPIC. The work presented different design choices and an example on how to implement the protection scheme in resource-constrained environment. The low overhead approach was targeted to both device authentication and user authentication.

[11] introduced another combinational locking method that like [26, 25] utilizes a small programmable part within the chip, that is referred to by *reconfigurable logic barriers*. However, unlike the earlier work that used the programmable parts for passive metering, the reconfigurable logic barriers in [11] were used for active metering. [11] decomposes the IP functionality $F(x)$ into F_{fixed} and $F_{reconfig}$. The idea is to protect the design by not disclosing the $F_{reconfig}$ to the fabrication plant. The withheld $F_{reconfig}$ partition is programmed into the reconfigurable locations during the activation stage using a secure key distribution. The suggestion combinational locking method is a mix of XORs (similar to [37]) and k-input lookup tables (LUTs). A selection heuristic was proposed to optimize the lock placement. The heuristic is aimed at making a barrier for the logic transition, such that the signals would not traverse to the output without passing through the locks. Except for heuristic methods, no security guarantee was provided by this work.

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. For more technical details, we refer the interested readers to the published papers on the external active metering subject, including [18, 37, 36, 38, 11].

1.5 Conclusions

This chapter 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 (which maybe reproducible or unclonable), or it assigns a signature to the internals of an IC, while the chip maintains its functionality and integrity. One interesting aspect of passive metering is that it is possible to uniquely monitor and track legacy ICs without the need for added circuitry by exploiting the inherent variations in silicon. 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 unclocked (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.

References

1. Defense science board (DSB) study on high performance microchip supply. http://www.acq.osd.mil/dsb/reports/2005-02-hpms_report_final.pdf.
2. Managing the risks of counterfeiting in the information technology industry. a white paper by kpmg and the alliance for gray market and counterfeit abatement (agma).
3. Pentium III serial numbers, http://www.pcmech.com/article/pentium-iii-serial-numbers/.
4. Pentium III wikipedia page, http://en.wikipedia.org/wiki/pentium_iii.
5. Y. Alkabani and F. Koushanfar. Active control and digital rights management of integrated circuit IP cores. In *International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES)*, 2007.
6. Y. Alkabani and F. Koushanfar. Active hardware metering for intellectual property protection and security. In *USENIX Security Symp.*, pages 291–306, 2007.
7. Y. Alkabani and F. Koushanfar. Consistency-based characterization for IC Trojan detection. In *International Conference on Computer Aided Designs (ICCAD)*, pages 123–127, 2009.
8. Y. Alkabani, F. Koushanfar, N. Kiyavash, and M. Potkonjak. Trusted integrated circuits: A nondestructive hidden characteristics extraction approach. In *Information Hiding conference (IH)*, pages 102–117. Springer, 2008.
9. Y. Alkabani, F. Koushanfar, and M. Potkonjak. Remote activation of ICs for piracy prevention and digital right management. In *The International Conference on Computer-Aided Design (ICCAD)*, 2007.
10. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In *Advances in Cryptology (CRYPTO)*, pages 1–18, 2001.
11. A. Baumgarten, A. Tyagi, and J. Zambreno. Preventing IC piracy using reconfigurable logic barriers. *IEEE Design & Test of Computers*, 27:66–75, 2010.
12. R. Chakraborty and S. Bhunia. Hardware protection and authentication through netlist level obfuscation. In *The International Conference on Computer-Aided Design (ICCAD)*, pages 674–677, 2008.

13. S. Devadas and B. Gassend. Authentication of integrated circuits. US Patent 7,840,803, 2010. Application in 2002.
14. B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Silicon physical random functions. In *CCS*, pages 148–160, 2002.
15. R. Helinski, D. Acharyya, and J. Plusquellic. A physical unclonable function defined using power distribution system equivalent resistance variations. In *Design Automation Conference (DAC)*, pages 676–681, 2009.
16. R. Helinski, D. Acharyya, and J. Plusquellic. Quality metric evaluation of a physical unclonable function derived from an IC's power distribution system. In *Design Automation Conference*, DAC, pages 240–243, 2010.
17. 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.
18. J. Huang and J. Lach. IC activation and user authentication for security-sensitive systems. In *International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 76–80, 2008.
19. D. Kirovski and M. Potkonjak. Localized watermarking: methodology and application to operation scheduling. In *The International Conference on Computer-Aided Design (ICCAD)*, pages 596–599, 1999.
20. F. Koushanfar. Active integrated circuits metering techniques for piracy avoidance and digital rights management. Technical Report TREE1101, ECE Dept., Rice University, 2011.
21. F. Koushanfar and Y. Alkabani. Provably secure obfuscation of diverse watermarks for sequential circuits. In *International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 42–47, 2010.
22. F. Koushanfar, P. Boufounos, and D. Shamsi. Post-silicon timing characterization by compressed sensing. In *The International Conference on Computer-Aided Design (ICCAD)*, pages 185–189, 2008.
23. F. Koushanfar, I. Hong, and M. Potkonjak. Behavioral synthesis techniques for intellectual property protection. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 10(3):523–545, 2005.
24. F. Koushanfar and A. Mirhoseini. A unified submodular framework for multimodal IC trojan detection. *IEEE Transactions on Information Forensics and Security (TIFS), to appear*, 2011.
25. F. Koushanfar and G. Qu. Hardware metering. In *Design Automation Conference (DAC)*, Design Automation Conference (DAC), pages 490–493, 2001.
26. F. Koushanfar, G. Qu, and M. Potkonjak. Intellectual property metering. In *International Workshop on Information Hiding (IHW)*, pages 81–95. Springer, 2001.
27. J. Lach, W. Mangione-Smith, and M. Potkonjak. Signature hiding techniques for FPGA intellectual property protection. In *The International Conference on Computer-Aided Design (ICCAD)*, pages 186–189, 1998.
28. K. Lofstrom, W. R. Daasch, and D. Taylor. *IC* identification circuit using device mismatch. In *International Solid-State Circuits Conference (ISSCC)*, pages 372–373, 2000.
29. R. Maes, D. Schellekens, P. Tuyls, and I. Verbauwhede. Analysis and design of active *IC* metering schemes. In *International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 74–81, 2009.
30. C. Mouli and W. Carriker. Future fab: How software is helping intel go nano–and beyond. *IEEE Spectrum*, March 2007.
31. M. Nelson, A. Nahapetian, F. Koushanfar, and M. Potkonjak. SVD-based ghost circuitry detection. In *IH*, pages 221–234, 2009.
32. A. Oliveira. Techniques for the creation of digital watermarks in sequential circuit designs. *IEEE Transactions on. Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 20(9):1101–1117, 2001.

33. M. Potkonjak and F. Koushanfar. Identification of integrated circuits. US Patent Application 12/463,984; Publication Number: US 2010/0287604 A1, May 2009.
34. M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey. Hardware Trojan horse detection using gate-level characterization. In *Design Automation Conference (DAC)*, pages 688–693, 2009.
35. G. Qu and M. Potkonjak. *Intellectual Property Protection in* VLSI *Design*. Springer, 2003.
36. J. Roy, F. Koushanfar, and I. Markov. Protecting bus-based hardware IP by secret sharing. In *Design Automation Conference (DAC)*, pages 846–851, 2008.
37. J. Roy, F. Koushanfar, and I. Markov. EPIC: Ending piracy of integrated circuits. In *Design Automation and Test in Europe (DATE)*, pages 1069–1074, 2008.
38. J. Roy, F. Koushanfar, and I. Markov. Ending piracy of integrated circuits. *IEEE Computer*, 43:30–38, 2010.
39. U. Rührmair, S. Devadas, and F. Koushanfar. *Book Chapter in Introduction to Hardware Security and Trust, M. Tehranipoor and C. Wang editors*, chapter 7: Security based on Physical Unclonability and Disorder. Springer, 2011.
40. B. Santo. Plans for next-gen chips imperiled. *IEEE Spectrum*, August 2007.
41. D. Shamsi, P. Boufounos, and F. Koushanfar. Noninvasive leakage power tomography of integrated circuits by compressive sensing. In *International Symposium on Low Power Electronic Design (ISLPED)*, pages 341–346, 2008.
42. I. Torunoglu and E. Charbon. Watermarking-based copyright protection of sequential functions. *IEEE Journal of Solid-State Circuits (JSSC)*, 35(3):434–440, 2000.
43. S. Wei, S. Meguerdichian, and M. Potkonjak. Gate-level characterization: Foundations and hardware security applications. In *Design Automation Conference (DAC)*, 2010.
44. L. Yuan and G. Qu. Information hiding in finite state machine. In *Information Hiding Conference (IH)*, pages 340–354. Springer, 2004.