EPIC: Ending Piracy of Integrated Circuits

Jarrod A. Roy†, Farinaz Koushanfar‡ and Igor L. Markov†
†The University of Michigan, Department of EECS, 2260 Hayward Ave., Ann Arbor, MI 48109-2121
‡Rice University, ECE and CS Departments, 6100 South Main, Houston, TX 77005

Abstract

As semiconductor manufacturing requires greater capital investments, the use of contract foundries has grown dramatically, increasing exposure to mask theft and unauthorized excess production. While only recently studied, IC piracy has now become a major challenge for the electronics and defense industries [6].

We propose a novel comprehensive technique to end piracy of integrated circuits (EPIC). It requires that every chip be activated with an external key, which can only be generated by the holder of IP rights, and cannot be duplicated. EPIC is based on (i) automatically-generated chip IDs, (ii) a novel combinational locking algorithm, and (iii) innovative use of public-key cryptography. Our evaluation suggests that the overhead of EPIC on circuit delay and power is negligible, and the standard flows for verification and test do not require change. In fact, major required components have already been integrated into several chips in production. We also use formal methods to evaluate combinational locking and computational attacks. A comprehensive protocol analysis concludes that EPIC is surprisingly resistant to various piracy attempts.

1. Introduction

As LSI Logic quit semiconductor manufacturing in 2005 and Texas Instruments chose not to develop sub-45nm fabrication in-house, they and their former clients partnered with major foundries to outsource production. In Summer 2007, Qualcomm became the first fabless semiconductor company to rank among top 10 IC producers worldwide [10], and even AMD has been outsourcing some of its production to foundries throughout the world. However, with the growth of manufacturing potential in Asia, piracy has become rampant, thanks to loose IP protection policies and weak enforcement [6, 20]. This was recently illustrated by the discovery of a "fake NEC Corp." in China that offered 50 counterfeit products [5]. Global piracy of hardware and software IP is now approaching $1B per day, with a major share in computers, peripherals, and embedded systems [21]. Indeed, once a fab starts producing chips from client's masks, unauthorized copies can be made cheaply. As pointed out by the US Defense Science Board [6], masks can also be stolen by industrial and military spies.

The practice of hardware piracy is very different from that of software piracy because hardware cannot be cloned and masks are much more difficult to change compared to software. The technological and financial barriers to hardware piracy are higher, but pirates tend to be better prepared [3], which makes countering them more challenging.

Until recently, only passive IC protection was available, based on unique chip IDs or programmable parts [9, 11, 17, 18]. Alkabani and Koushanfar [1] proposed the first active scheme to fight hardware piracy. The method exploits the inherent unique manufacturing variability of the ICs to generate chip IDs. The IDs are integrated within an augmented finite state machine (FSM) in a way that every chip starts in a unique state (locked). The designer, knowing the augmented FSM structure, would be the only entity who could send the key to activate (unlock) the IC. A newer remote activation scheme in [2] relies on a set of unique chip IDs to lock edge transitions on the FSM of the design, for pairs of consecutive transitions of a few replicated states.

We propose a novel technique to end piracy of integrated circuits (EPIC). Before testing, each chip generates its own random identification number using well-known techniques. In order for a chip to become functional, the manufacturer must send that ID to the holder of IP rights (IP-holder), who then sends an activation code that only activates a chip with that ID. This allows the IP-holder to control exactly how many chips are made and prevents others from making functional copies. Our contributions include: (i) the first purely combinational lock embedding and IC activation scheme; (ii) an exact algorithm for key embedding into an IC, with rigorous empirical evaluation; (iii) an adaptation of the standard design flow to facilitate chip activation and secure communication with negligible overhead; (iv) security guarantees; (v) analysis of attacks and countermeasures.

The remainder of the paper is as follows. Section 2 outlines the necessary background. An overview of proposed techniques is presented in Section 3. Details of the combinational locking technique are described in Section 4, along with a framework for analysis by formal methods. Section 5 discusses security guarantees, attacks, and countermeasures. Section 6 evaluates EPIC in terms of overhead, scalability and security. We conclude in Section 7.

2 Background

Public-key cryptography. Cryptography allows remote users to exchange messages (*plaintext*) through an untrusted medium, in such a way that transmissions intercepted by eavesdroppers do not reveal plaintext. The plaintext is *encrypted* by the sender and *decrypted* by the receiver. In 1976, Diffie and Hellman invented *asymmetric cryptography* also known as *public-key cryptography* (PKC) [7]. Each user independently generates a pair of keys, one *public* and one *private*. Public keys are made available to everyone, but private keys are never transmitted or revealed by their owners. Encryption and decryption rely on hard-to-reverse (*one-way*) mathematical functions, such as high-precision integer multiplication and modular exponentiation. No efficient algorithms are known to compute their inverses, i.e., for number-factoring and discrete logarithm.

The sender (B) encrypts plaintext with the public key of the receiver (A) and transmits the message, which can only be decrypted with A's private key. A system proposed in 1977 by Rivest, Shamir and Adleman (RSA), enriches public-key cryptography with a *digital signature* feature — if B additionally encrypts his message with his private key, then A can use B's public key to verify that the message is unaltered and coming from B. PKC is widely used for certificates of authenticity, generating and verifying digital signatures, and for exchanging symmetric keys that allow faster communication. RSA-style crypto-systems are among the most studied in the literature, but remain resilient against a variety of attacks 30 years after their inception.

On-chip true random number generators (TRNGs). Randomized algorithms often use *pseudo-random* number generators (PRNGs), i.e., deterministic sequences with random appearance that are initiated by an input seed. However, cryptographic applications demand *true randomness*, so as to circumvent attacks based on predictability. True random bits are typically generated by sampling chaotic physical phenomena, such as thermal noise, quantum-mechanical measurement, meta-stability in latches, etc [19]. Such TRNGs are a major component in cryptographic applications and can be found in commercial ICs. For example, the upcoming NIAGARA 2 processor from Sun couples one TRNG in each of its eight cores with cryptographic units to support secure generation of public and private keys [13]. We use TRNGs to define randomized chip IDs upon power-up, but such chip IDs can also be produced using on-chip variation [17, 18].

Manufacturing of integrated circuits and IC piracy. ICs consist of over 20 patterned layers of metals, insulators and semiconductors, with smallest feature sizes at 45nm and decreasing. The patterns are "burned in" by shining a 193-nm ArF laser through chromium-quartz masks in a tightly controlled process at fabrication facilities (*fabs*) [12]. A mask set contains a complete physical representation of an IC. Contract fabs, such as TSMC and UMC produce masks from large computer files supplied by their clients — approximately 1350 fabless design companies, such as Qualcomm and Broadcom, with total revenue of $49.7B (Source: Fabless Semiconductor Association). The IC descriptions given to fabs are often customized to satisfy a fab's specific requirements, but if stolen, they may conceivably be adjusted to another fab, and leading-edge fabs are concerned about this. Another form of piracy is for the contracted fab to produce more chips than authorized, at a very small additional cost, and sell them on the black market. A simple anti-piracy measure is *wafer banking*, i.e., contracting out different layers of a chip to different manufacturers. Not only is this expensive, but it prevents fabs from testing ICs which hampers yield analysis and improvement. Fabricating features smaller than half of 193nm (the ArF laser's wavelength) is increasingly difficult, and no viable replacements to ArF lasers are expected in the near future [14]. To compensate for optical diffraction, mask patterns are much more complex than the manufactured patterns and may be harder to reverse-engineer by delamination or otherwise. Physically modifying fine-grain features of ICs after manufacturing, to defeat anti-piracy measures, is very difficult. The Focused Ion Beam (FIB) technique is sometimes used to reconnect wires during post-silicon debugging, but remains too slow and expensive for mass production, and will likely be infeasible for ICs with 32nm features.

3 Overview of proposed techniques

The proposed techniques consist of modifications to existing IC design flows to embed keys into the circuit (Figure 1a) and a new protocol for chip activation (Figure 1b). This empowers the holder of IP rights for the IC to unlock every manufactured chip. Without proper keys, none of the chips will function properly or pass routine circuit test. The keys are constructed so that different chips, even from the same wafer, require different keys. Therefore, the key for each chip must be requested from the IP rights holder through secure communications. To support public-key cryptography, the IP rights holder must generate a pair of Master Keys (MK) — public and private — that will remain unchanged. The private Master Key (MK-Pri) embodies IP rights for a given design and is never transmitted (see Table 1). This remote unlocking mechanism allows one to meter activated ICs, log serial numbers, limit activation to certain parties, only at certain rates and only at certain times of the day.

In our piracy-aware IC design flow in Figure 1a, RTL descriptions are enriched with support for on-chip TRNG and public-key cryptography. In particular, each manufactured IC should be able to generate its own random public and private keys upon start-up. Also embedded in RTL is the public Master Key (MK-Pub) and minimal circuitry to

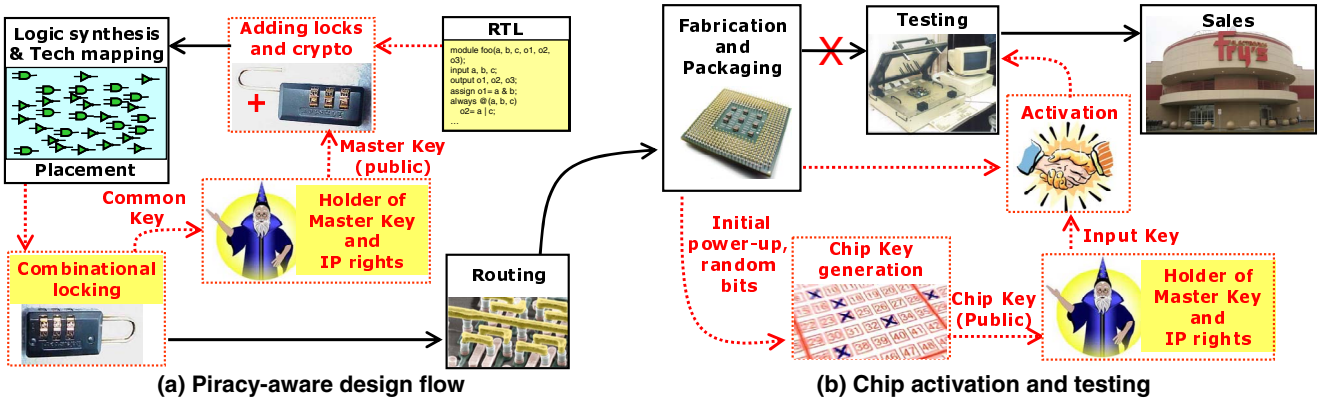

Figure 1. Overview of the proposed EPIC technique.

support the *combinational locking* mechanism described in Section 4 below. At this point, none of the newly added components are connected to the original logic.

As shown in Figure 1a, a gate-level netlist is produced from the enriched RTL using traditional logic synthesis and technology mapping, followed by circuit placement. Now critical paths in the circuit are known, and one may connect the anti-piracy logic without disturbing them. Combinational locking is performed in most important modules of the IC by adding XOR gates on selected non-critical wires, with added controls connected to the Common-Key register. When the correct Common Key (CK) appears, the circuit is equivalent to the original. Otherwise, the circuit's behavior is altered, as if stray inverters were placed on selected wires. The Common Key is generated at random, so as to prevent it from being stolen earlier. After modifying the placed design, one securely communicates CK to the holder of IP rights and erases all other copies. Routing and other physical optimizations then proceed as normal, followed by manufacturing.

Fabricated chips are packaged and must be activated before testing (Figure 1b). During the initial power-up, each chip generates a pair of private and public Random Chip Keys (RCK), which are burned into electrically-programmable fuses, e.g. the Electronic Fuse Unit (EFU) in Sun's NIAGARA 2 processor [13], to prevent multiple activation attempts. To activate a chip, the fab must establish a secure link with the holder of IP rights and transmit RCK-Pub for a chip that is being activated. The transmission is authenticated using the fab's private key.[1] In response, the IP rights holder sends the Input Key (IK), which represents CK encrypted with MK-Pri and RCK-Pub. Using RCK-Pub to encrypt communications makes statistical attacks against MK-Pri more difficult. The resulting IK can be additionally encrypted using the fab's public key so that only the fab can receive it. When entered into the chip, IK is decrypted using RCK-Pri and MK-Pub, which also authenticates it as being sent by the holder of IP rights. Upon decryption, CK is produced, which unlocks the chip and facilitates test. After that, the chip can be sold.

[1]Extensions to this protocol may send a time-stamp, serial number, etc.

4 Combinational locking

To protect a combinational circuit $C(\vec{x})$ with a k-bit key, we develop a simple procedure that uses k new gates. First, k wires $\{w_i\}$ are selected and matched with the bits $\{y_i\}$ of the key.[2] For each selected wire w_i, its driver is disconnected from the sinks and either an XOR gate $w_i' = w_i \oplus y_i$ or XNOR gate $w_i' = w_i \overline{\oplus} y_i$ is inserted, where y_i is the matched key bit and w_i' is a new wire that drives all sinks previously driven by w_i. The choice of XOR gate versus XNOR gate depends on the chosen value of the matched key bit: if the chosen value of y_i is 0, $w_i' = w_i \oplus y_i$, otherwise $w_i' = w_i \overline{\oplus} y_i$. Using the identity, $w_i \oplus y_i = \overline{w_i} \overline{\oplus} y_i$, one can replace an XOR gate with an XNOR gate and an inverter and, similarly, XNOR gates can be replaced by XOR gates and inverters.[3]

In general, multiple key combinations are unlikely to unlock $C'(\vec{x},\vec{y})$ because $w_i \oplus 1 = w_i \overline{\oplus}\ 0 = \overline{w_i}$, i.e., incorrect input key bits correspond to an inverter inserted into $C(\vec{x})$. Notable exceptions are circuits consisting entirely of XOR and XNOR gates, e.g., an XOR tree can be unlocked by 50% of all key combinations. However, this is not typical for circuits that use few XOR gates (see also Section 6). We prefer $C'(\vec{x},\vec{y})$ to admit only a unique key combination, i.e.,

$$\exists!\, \vec{y}\ \ \forall\, \vec{x} \quad C'(\vec{x},\vec{y}) = C(\vec{x}) \tag{1}$$

With ! omitted, this expression gives a Boolean equation for finding a working key combination. However, solving such an equation is harder than NP-complete, due to alternating quantifiers. In practical terms, this means that a SAT solver alone would be insufficient to find a key combination of non-trivial length, but Reduced Ordered Binary Decision Diagrams (ROBDDs) offer more appropriate tools [8]. To this end, one can represent the operation $=$ by constructing a *miter circuit*, then build the ROBDD of the miter, followed by universal and existential quantification using well-known ROBDD algorithms [8]. The resulting ROBDD compactly represents all good key combinations by its paths, which can be counted in time $O(size)$. This

[2]Wires are selected to avoid critical paths and congested regions; matching can minimize wirelength. Note that inputs and outputs of flip-flops are often on critical paths, and are not as numerous as internal wires.

[3]Inversions can be moved further up or down using de Morgan's law.

KEY	Trans-mitted?	LOCATION				
		RTL	Placed design	Masks	Working chip	IP holder
MK-Pri	-	-	-	-	-	✓
MK-Pub	§	✓	✓	✓	✓	✓
CK	§	-	-	✓	✓	✓
RCK-Pri	-	-	-	-	✓	-
RCK-Pub	✓	-	-	-	✓	✓
IK	✓	-	-	-	-	✓

Table 1. Keys used by the EPIC technique. § MK-Pub and CK are transmitted before mask creation and have smaller risk of interception.

formal method can be used to check the uniqueness of a key combination, but may also help forgers to discover the Common Key, if both $C'(\vec{x},\vec{y})$ and $C(\vec{x})$ are available. Section 6 evaluates both uses.

A key should be long enough to withstand *brute-force attacks*, which are defined as algorithms searching for a key that evaluate combinations and spend $\Omega(1)$ time per combination. For combinational locking, such attacks are additionally hampered by the NP-completeness of checking even one key combination. In practice, most incorrect combinations can be weeded out by scanning-in test patterns and comparing circuit's responses to expected values. With a single scan-chain, this will take $\Omega(2^k)$ time for a k-bit key. However, multiple scan-chains can be run separately, and brute-forcing a (k_1+k_2)-bit key, whose k_1 and k_2 bits can be checked by different scan-chains, would take $\Omega(2^{k_1}+2^{k_2})$ time rather than $\Omega(2^{k_1+k_2})$.

Definition 1 *Given a circuit $C'(\vec{x},\vec{y})$ locked with key $\vec{y}$, the* effective length *$\mathscr{L}(\vec{y})$ of the key is $\log_2$ of the expected number of combinations checked by best brute-force attack.*

Theorem 1 *Consider a circuit $C'(\vec{x},\vec{y})$ such that the key $\vec{y}$ locks n independently-testable circuit modules and, for $j = 1..n$, exactly k_j bits of the key are dedicated to module j, while G_j key combinations of 2^{k_j} unlock module j. Then*

$$\mathscr{L}(\vec{y}) \leq \log_2\left(\Sigma_{j=1}^{n}\, 2^{k_j}/G_j\right) - 1 \qquad (2)$$

In practice, having several good key combinations may be useful, e.g., to trace activation by different parties. However, this would decrease the effective length of the key. Based on results in Section 6, we recommend $\mathscr{L}(\vec{y}) \geq 64$.

5 Vulnerability assessment

The main objective of our work is to protect ICs against piracy through unauthorized excess production and stolen masks. However, pirates may also steal RTL or gate-level netlists, layouts, as well as test-vectors and correct responses. Additional conceivable scenarios include reverse-engineering and modification of masks, production-scale modification of manufactured chips, and real-time observation of transient signals in successfully-activated chips. As we show in this section, EPIC provides robust multi-layered defense against the considered attacks.

Obstacles to piracy. To prioritize possible attacks, we distinguish four categories of obstacles faced by forgers in their attempts to pirate ICs.

- Lack of information, e.g., not being able to obtain MK-Private because it is never transmitted.
- Computational complexity, e.g., not being able to break RSA-style public-key crypto-systems.
- Technological barriers, e.g., not being able to reverse-engineer the active layers of 45nm ICs or masks.
- Financial barriers, e.g., not being able to invest amounts larger than expected revenue from piracy.

Example 1 To break EPIC by obtaining keys and without modifying masks or chips, it is necessary to obtain RCK-Public for each chip, MK-Private and CK. While these three keys lead to IK, none of them is present in RTL or synthesized gate-level netlist, while RCK-Public and MK-Private are not present in masks either. CK may conceivably be discovered by watching transient signals on an activated chip, but for 45nm chips that would require very sophisticated technology. On the other hand, computational attacks seeking CK would require gate-level netlists for both $C(\vec{x})$ and $C'(\vec{x},\vec{y})$[4], as well as astronomical amounts of time, according to results in Section 6. Even if CK is discovered by pirates, and if they manage to read off RCK-Public from each chip, having a full understanding of all masks and full access to each IC will not reveal MK-Private, which is guaranteed by RSA-style public-key cryptography.

EPIC guarantees. EPIC's multi-layered protection requires two basic technology assumptions: (i) cryptographic security of RSA-like public-key crypto-systems, as well as (ii) good statistical properties of TRNGs or chip IDs [13, 17, 18], and their resilience to attacks (the randomness of RCK). Additionally, proper selection of CK in Section 6 ensures a limited number of good key combinations, and defeats brute-force and formal-methods attacks. The following logistical properties of EPIC can then be deduced.

Proposition 1 *RCK-Public and MK-Public do not reveal information about their private counterparts.*

Proposition 2 *Knowing CK, all public keys and both RCKs is insufficient to generate IK (irreversibility of PKC).*

Proposition 3 *There are as many good CKs as good IKs.*

Proposition 4 *Good IKs are as random as RCKs.*

Additional properties of EPIC hold when forgers cannot modify masks or ICs (but may have access to source files).

Proposition 5 *Different ICs nearly always have different RCKs.*

Proposition 6 *Knowing a good CK is not sufficient to unlock multiple chips.*

Proposition 7 *Different chips nearly always have different IKs. Eavesdropping on data exchanged during activation of a chip will not reveal IKs for other chips.*

[4] Note that $C'(\vec{x},\vec{y})$ alone does not express the correct behavior of $C(\vec{x})$, and thus gives no criterion to check for possible CK.

Proposition 8 *A chip can only be unlocked by entering an appropriate IK.*

Attacks and countermeasures. As pointed out above, a full understanding of masks, intercepting all communications, and even inspecting all signals in a successfully activated chip is not sufficient to break EPIC. In the context when masks and chips cannot be modified by the forger, stealing RTL or gate-level netlists does not give much help either. Security can be further improved if chip-activation data are additionally encrypted by the fab, offering stronger cryptography that can be changed on demand. This also hampers man-in-the-middle attacks and denial-of-service attacks, where spurious activation data are sent to the holder of IP rights. Additionally, better traceability to fab will encourage better physical security.

Protecting against human breaches. One of the most serious attacks on EPIC is the theft of CK and MK-Private from the holder of IP rights — it is almost tantamount to the theft of IP rights and allows the pirates to produce IKs. As a countermeasure, EPIC can be reinforced with Fab Keys — FK-Public is embedded in RTL, while FK-Private is held by the Fab and is required to produce IK. This way, a pirate not associated with the fab will be unable to unlock chips.

Technologically advanced forgers. Without access to MK-Private, the pirates must modify chips or masks. Focused Ion Beam (FIB) would be too slow for production, but a full understanding of masks *and* the ability to arbitrarily change them gives the pirates an upper hand, at least in principle. Once they discover CK, they can hardwire it, bypassing input pins, TRNG and PCK hardware. However, this scenario is unlikely because, at 45nm and below, masks are much harder to read than the actual shapes on the chip, due to Resolution Enhancement Techniques (RET). Scanning the actual shapes *in silico* is even harder, and the investment required for this may not pay off because pirated chips sell at a lower cost, often at low volumes.

6 Evaluation of proposed techniques

In this section we evaluate EPIC in terms of its overhead, impact on traditional design flows and the difficulty of inserting the XOR gates that implement CKs. We also analyze the effectiveness of formal and brute-force attacks on EPIC.

Overhead reduction. Component overhead of EPIC includes: (i) additional pins to enter IK, (ii) additional gates and wires to implement combinational locking, (iii) true random number generator (TRNG), (iv) hardware for public-key cryptography (RSA). Since the majority of the chip remains dormant until activation succeeds, an existing pin can be multiplexed to enter IK using a proper data serialization protocol. Combinational locking does not affect critical path delays; it requires orders of magnitude fewer gates and wires than available on ICs, making its area and power overhead minor. A single TRNG is required, and existing TRNGs are rather small ($0.036mm^2$ in 130nm [19]). RSA can be implemented with fewer than 10,000 2-input gates [15]. RSA can also be turned off after activation (no power overhead) and does not affect critical paths (no delay overhead). Sun's NIAGARA 2 processor implements RSA in each of its 8 cores, with area overhead below 1% [13].

Impact on traditional design flows. EPIC is unique in that it does not require significant changes to established verification and testing flows. Indeed, test vectors developed for the original circuit remain valid after proposed changes because the unlocked IC behaves just like the original IC. Traditional verification techniques can be applied similarly. While the insertion of XORs during CK embedding is a relatively simple step, it can also be verified using SAT-based equivalence checking.

Empirical evaluation of combinational locking. We develop two methods for counting the number of valid CKs in a circuit when XOR gates have been inserted. The first method is a formal technique that builds Equation 1 using ROBDDs and solves for all valid CKs. The second method is a brute-force approach that tries every possible CK and checks equivalence with the original circuit using ROBDDs. We use the CUDD [16] ROBDD package and implement both of these techniques in $\sim$ 400 lines of C++ code.

We evaluate the two techniques by inserting XOR gates into combinational circuits at random and counting valid CKs. All experiments were performed on a 2.4GHz Opteron processor with 8GB of RAM. Table 2 shows results of both techniques on two ALU circuits c880 and c3540 from the ISCAS'85 suite [4]. Surprisingly, the brute-force method is more efficient than the formal method on c880; in all cases, the formal method uses more runtime and memory. On c3540, brute-force is more memory efficient, but requires more runtime than the formal method. For 24-bit and larger keys, runtime for the formal method grows nearly exponentially, making it infeasible as an attack on EPIC.

We also observe that inserting XOR gates randomly produces relatively few duplicate keys. For up to 32 bits on the c3540 benchmark, the valid key is unique. On the c880 benchmark, 4 of 2^{32} key combinations are valid, which only reduces the effective bit length by 2. For a 64-bit key in c880 to be breakable in less than 1 year, more than 2^{20} key combinations would need to be valid. According to our experiments on these and the remaining ISCAS'85 circuits, such an explosion in the number of valid keys is highly unlikely. If an attacker parallelized the brute-force method with 10,000 times our resources, considering duplicate keys, it would take 100 years to find a valid 64-bit key on c880. In our experiments, random insertion of XOR gates to as many as 1/8 of the gates did not produce many duplicate keys. Therefore, our suggested key length of 64 bits can be supported by most circuits with 500 gates, as well as by many smaller circuits.

c880 (60 in, 26 out, 383 gates)				c3540 (50 in, 22 out, 1669 gates)			
Common Key		Runtime (sec)		Common Key		Runtime (sec)	
bits	# valid	formal	bruteF	bits	# valid	formal	bruteF
12	1	128	1	12	1	94	66
13	1	737	1	13	1	116	75
14	1	195	1	14	1	148	186
15	2	555	2	15	1	250	258
16	2	3291	2	16	1	298	413
17	2	584	4	17	1	310	608
18	2	383	9	18	1	382	1060
19	2	868	15	19	1	519	2008
20	2	5375	29	20	1	369	2296
21	4	$>$ 24 hrs	60	21	1	701	5562
22	4	6670	117	22	1	408	11560
23	4	3905	230	23	1	839	16907
24	4	26008	462	24	1	5560	35015
32	4	$>$ 72 hrs	$>$36 hrs	32	1	150889	$>$ 3 mnths
64	$\sim$16	$> 10^6$ years		64	$\sim$4	$> 10^6$ years	

Table 2. Counting the number of valid Common Keys for randomly inserted XOR gates on the c880 and c3540 ISCAS'85 circuits [4]. Trends on the remaining ISCAS'85 circuits are similar. Data for 64-bit keys are estimated.

7 Conclusions

Our approach to defeating piracy of ICs is to render infringement unprofitable by making the majority of attacks computationally infeasible. This is accomplished through a novel low-overhead combinational chip-locking system and a chip-activation protocol based on public-key cryptography (EPIC). Circumventing our methodology without modifying the masks or ICs is very difficult because of the strong security guarantees provided by public-key cryptography. On the other hand, production-scale modification of fabricated ICs is infeasible today, and especially so for advanced technology nodes. Mask modification and other related scenarios appear to require unacceptably high investment, which may not be justified by revenue from pirated ICs. To this end, we note that pirated ICs are normally *late to market*, while enjoying *smaller volumes* and *smaller margins* than original ICs. Additionally, pirates *cannot advertise openly* and must justify *higher risk* by higher margins. This limits pirates' investment and makes it nearly impossible to justify NRE costs or gradually ramp up yield on an alternative fab. EPIC can also be applied to modern FPGAs with bitstream encryption, introduced by Xilinx in 2001 [20], by locking combinational cryptographic circuits.

In addition to actively preventing piracy (*active hardware metering*), EPIC also facilitates passive hardware metering by requiring serial numbers to be transmitted during chip activation. Overall, we hope that by limiting piracy, the use of EPIC will improve the economics of the IC industry.

Acknowledgments. We would like to thank Dr. Mark Brehob and Prof. Andrew Kahng for their feedback on earlier drafts of this work. Prof. Koushanfar's work is partially supported by the DARPA/MTO Trust in Integrated Circuits/Young Faculty Award (YFA) under grant award W911NF-07-1-0198, and by the NSF ITR-CYBERTRUST under grant number 0716674.

References

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

[2] Y. Alkabani, F. Koushanfar, and M. Potkonjak. Remote activation of ICs for piracy prevention and digital rights management. In *IEEE/ACM ICCAD*, pp. 674–677, 2007.

[3] R. Anderson. *Security Engineering: A guide to building dependable distributed systems*. John Wiley and Sons, 2001.

[4] F. Brglez and H. Fujiwara. A neutral netlist of 10 combinational circuits and a target translator in FORTRAN. In *IEEE ISCAS*, 1985.

[5] P. Clarke. Fake NEC company found, says report. EE Times, May 4, 2006. `http://www.eetimes.com/showArticle.jhtml?articleID=187200176`

[6] Defense Science Board (DSB) study on High Performance Microchip Supply. `http://www.acq.osd.mil/dsb/reports/2005-02-HPMS_Report_Final.pdf`

[7] N. Ferguson and B. Schneier. *Practical Cryptography*. John Wiley and Sons, 2003.

[8] G. D. Hachtel and F. Somenzi. *Logic Synthesis and Verification Algorithms.* Kluwer, 2000

[9] F. Koushanfar, G. Qu, and M. Potkonjak. Intellectual property metering. In *Inf. Hiding Workshop*, pp. 81–95, 2001.

[10] M. LaPedus, Qualcomm cracks top-10 in chip rankings. EE Times, August 23, 2007. `http://www.eetimes.com/news/semi/showArticle.jhtml?articleID=201801923`

[11] K. Lofstrom, W. Daasch, and D. Taylor. IC identification circuits using device mismatch. In *ISSCC*, pp. 372–373, 2000.

[12] C. Mouli and W. Carriker. Future fab. *IEEE Spectrum* 44(3), pp. 38–43, March 2007. `http://www.spectrum.ieee.org/mar07/4941`

[13] U. M. Nawathe et al. An 8-Core 64-thread 64b power-efficient SPARC SoC. In *ISSCC*, pp. 108–611, 2007. `http://www.opensparc.net/opensparc-t2/index.html`

[14] B. Santo, Plans for next-gen chips imperiled. *IEEE Spectrum* 44(8), pp. 12–14, August 2007. `http://www.spectrum.ieee.org/aug07/5394`

[15] Sciworx RSA Co-Processor. `http://www.sci-worx.com/products/cryptography/rsa-co-processor.html`

[16] F. Somenzi, CUDD: CU decision diagram package. ver. 2.4.1, Univ. of Colorado at Boulder, 2004. `http://vlsi.colorado.edu/~fabio/CUDD/`

[17] Y. Su, J. Holleman, and B. Otis. A 1.6J/bit stable chip ID generating circuit using process variations. In *ISSCC*, pp. 406–611, 2007.

[18] G. E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In *DAC*, pp. 9–14, 2007.

[19] C. Tokunaga, D. Blaauw and T. Mudge. True random number generator with a metastability-based quality control. In *IEEE ISSCC*, pp. 404-405, 2007.

[20] S. Trimberger. Trusted design in FPGAs. *DAC*'07, pp. 5–8.

[21] VSI Alliance - IP Protection Development Working Group. The value and management of intellectual assets. 2000. `http://vsi.org/documents/datasheets/TOC_IPPWP210.pdf`