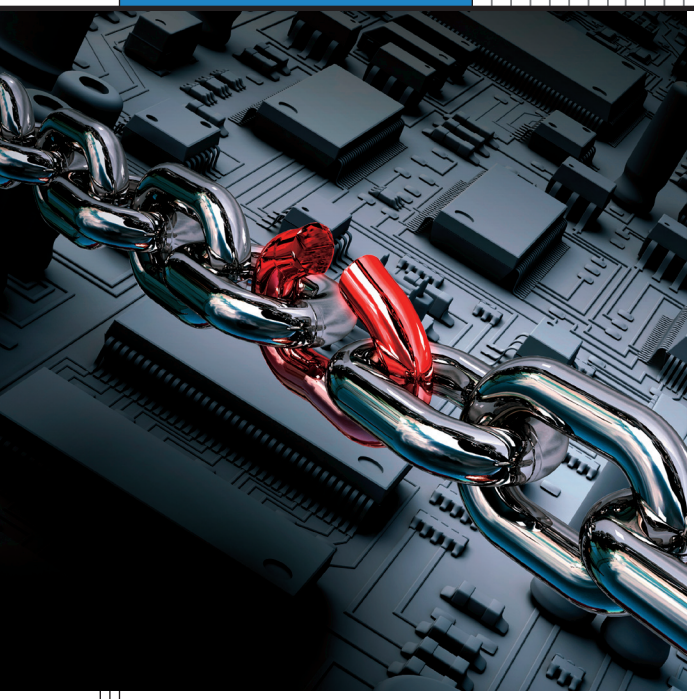


ENDING PIRACY OF INTEGRATED CIRCUITS

Jarrod A. Roy, *IBM Corp.*
Farinaz Koushanfar, *Rice University*
Igor L. Markov, *University of Michigan*

An effective technique to combat IC piracy is to render infringement impractical by making physical tampering unprofitable and attacks computationally infeasible. EPIC accomplishes this using a novel low-overhead combinational chip-locking system and a chip-activation protocol based on public-key cryptography.

Dramatic increases in the cost of fabrication technology have shifted the semiconductor business to a contract foundry model, where leading-edge design houses outsource fabrication and packaging to offshore facilities with smaller operational costs. For example, AMD contracts some of its IC production to foundries throughout the world, and Texas Instruments chose not to develop sub-45-nm fabrication in-house, partnering with major foundries worldwide to outsource manufacturing.

In recent years, preventing design theft and integrated circuit piracy/overbuilding by contract foundries has become increasingly important to both government and industry.[1] The "Integrated Circuits and IC Piracy" sidebar explains why.

EPIC (Ending Piracy of Integrated Circuits) is a novel method that protects chips at the foundry by automatically and uniquely locking each IC. In EPIC, every IC is locked by asymmetric cryptographic techniques that require a specific external key. This key is unique for each chip, cannot be duplicated, and can only be generated by the IP rights holder. In addition to such automatically produced chip identifiers, EPIC relies on a new combinational locking method and an innovative application of public-key cryptography (PKC).

EPIC does not require significant changes to established IC design, verification, or test flows. Experimental evaluation shows that it incurs minimal overhead in power and delay. The major components required for EPIC are already integrated in existing commercial ICs. We have considered various possible attacks and analyzed the new combinational locking method using formal methods. The results indicate that EPIC is robust to numerous such attacks.

BACKGROUND

With the growth of fabrication potential in Asia, piracy and counterfeiting have become rampant thanks to loose IP protection policies and weak enforcement.[1] This was recently illustrated by the discovery of a fake "NEC Corp." in China that offered 50 counterfeit products.[2] According to the VSI Alliance, global piracy of hardware and software

 Published by the IEEE Computer Society 0018-9162/10/$26.00 © 2010 IEEE

IP is costing companies up to $1 billion per day, with a major share in computers, peripherals, and embedded systems. Indeed, once a fabrication plant (fab) starts producing chips from a client's paid masks, unauthorized copies can be made cheaply. As the US Defense Science Board has pointed out, industrial and military spies can also steal masks.[1]

Hardware piracy is comparable to pirated and counterfeit medications in that some cloned drugs work like their brand equivalents but others are deficient in potentially disastrous ways. Hardware piracy is unique, though, because hardware parts cannot be copied directly and manufacturing recipes (masks) are much more difficult to alter, as compared with other pirated goods such as software.[3]

Until a few years ago, only passive IC protection based on unique chip IDs or programmable parts was available.[4-6] Passive protection facilitates the detection of unauthorized products but does not interfere with their product cycle. The first active scheme proposed to fight hardware piracy exploits ICs' inherent manufacturing variability to generate chip IDs.[7] The IDs are integrated within an augmented finite state machine (FSM) such that every chip starts in a unique locked state. Only the designer, knowing the augmented FSM structure, can send the key to activate (unlock) the IC. A newer remote activation scheme relies on a set of unique chip IDs to lock edge transitions on the design's FSM for pairs of consecutive transitions of a few replicated states.[8]

The design of active IC protection techniques is challenging because various types of overhead may increase the cost of a product and make it less attractive to consumers. In particular, previous plans to embed predefined identification sequences into each Intel CPU, verifiable at any time, drew condemnation by privacy advocates because this allows tracking of consumers.

INTEGRATED CIRCUITS AND IC PIRACY

Integrated circuits consist of more than 20 patterned layers of metals, insulators, and semiconductors, with the smallest feature sizes at 45 nm and decreasing. The patterns are "burned in" by shining a 193 nm argon fluoride (ArF) laser through chromium-quartz masks in a tightly controlled process at fabrication facilities, or fabs.[1] The overhead of this complex and costly process is amortized over the multitude of chips made from the mask.

A mask set or pattern contains a complete physical representation of an IC. Contract fabs, such as Taiwan Semiconductor Manufacturing Company and Universal Manufacturing Corp., produce masks from large computer files supplied by their clients. In 2008, there were approximately 1,300 fabless design companies, such as Qualcomm and Broadcom, with a total revenue of $51 billion (www.gsaglobal.org/resources/industrydata/facts.asp). 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 antipiracy measure is wafer banking—contracting out different layers of a chip to different manufacturers. However, not only is this expensive, it prevents fabs from testing ICs, which hampers yield analysis and improvement.

Fabricating features smaller than half of 193 nm (the ArF laser's wavelength) is increasingly difficult, and no viable replacements to ArF lasers are expected in the near future.[2] 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 ICs' fine-grained features after manufacturing, to defeat antipiracy measures, is very difficult. The focused ion beam technique is sometimes used to reconnect wires during postsilicon debugging, but FIB remains too slow and expensive for mass production and will likely be infeasible for ICs with 32-nm or smaller features.

References

1. C. Mouli and W. Carriker, "Future Fab," *IEEE Spectrum*, Mar. 2007, pp. 38-43.
2. B. Santo, "Plans for Next-Gen Chips Imperiled," *IEEE Spectrum*, Aug. 2007, pp. 12-14.

EPIC OVERVIEW

EPIC uses a novel approach to combat IC piracy. Before testing, each chip generates its own unique random ID number using well-known techniques. For a chip to become functional, the manufacturer must send that ID to the IP rights holder, who then sends an activation code that only activates a chip with that specific ID. This lets the IP rights holder control exactly how many chips are made and prevents others from making functional copies. While the IP rights holder stores the IDs of activated chips (or their hash values) to detect repetitions, there is no link between a physical chip and an ID, and the use of IDs is not required beyond activation. If additional features need to be activated later, EPIC can generate new random IDs on demand.

To accomplish this, EPIC features

- the first purely combinational locking mechanism for digital logic circuits and its interface with PKC,
- a specific algorithm for key embedding into an IC,
- an adaptation of the standard chip design flow to facilitate chip activation and secure communication with negligible overhead, and
- security guarantees derived from those for well-known asymmetric cryptography protocols.

In addition, we rigorously evaluate EPIC both empirically and theoretically by analyzing attacks and countermeasures.

As Figure 1 shows, EPIC modifies existing IC design flows to embed keys into the circuit and introduces a new protocol for chip activation. A piracy-aware design flow

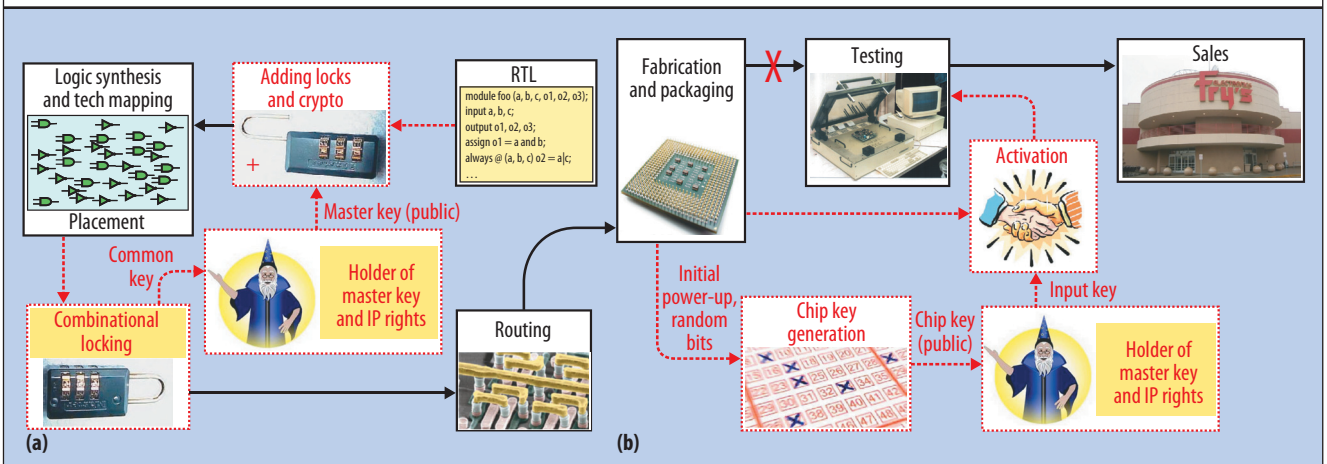


Figure 1. EPIC overview: (a) piracy-aware design flow and (b) chip activation and testing.

coupled with chip activation and testing empower the IP rights holder to unlock every manufactured chip. Without proper keys, none of the chips will function properly or pass routine circuit tests. The keys are constructed so that different chips, even from the same wafer, require different keys. Therefore, the IP rights holder must request each chip's key through secure communications.

Table 1 lists the various keys EPIC uses, whether they are transmitted, and where they are stored.

To support PKC, the IP rights holder must generate a pair of master keys (MKs)—public and private—that will remain unchanged. The private master key (MK-Pri) embodies IP rights for a given design and is never transmitted. This remote unlocking mechanism allows metering of activated ICs, logging of serial numbers, and limiting activation to certain parties only at particular rates and times of day.

Piracy-aware design flow

As Figure 1a shows, EPIC enriches register transfer level (RTL) descriptions with support for on-chip true random number generators, physically unclonable functions, and PKC. In particular, each manufactured IC should be able to generate its own random public and private keys upon start-up. Also embedded in RTL are the public master key (MK-Pub) and minimal circuitry to support the combinational locking mechanism. At this point, none of the newly added components are connected to the original logic.

Traditional logic synthesis and technology mapping tools produce a gate-level netlist from the enriched RTL, which is placed by a conventional circuit placer. With critical paths in the circuit known, the antipiracy logic can be connected without disturbing them.

EPIC performs combinational locking in the IC's most important modules by adding XOR gates on selected noncritical wires, with added controls connected to the common key (CK) register. When the correct 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. EPIC generates 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.

Table 1. Keys used by EPIC.

Key	Transmitted?	Location				
		RTL	Placed design	Masks	Working chip	IP rights holder
MK-Pri	–	–	–	–	–	✓
MK-Pub	*	✓	✓	✓	✓	✓
CK	*	–	–	✓	✓	✓
RCK-Pri	–	–	–	–	✓	–
RCK-Pub	✓	–	–	–	✓	✓
IK	✓	–	–	–	–	✓

*The MK-Pub and CK are transmitted before mask creation and have a smaller risk of interception.

Chip activation and testing

As Figure 1b shows, fabricated chips are packaged and must be activated before testing.

During the initial power-up, each chip generates a pair of public and private random chip keys (RCKs), using sources of randomness such as on-chip true random

number generators (TRNGs) or physically unclonable functions (PUFs). The "On-Chip TRNGs and PUFs" sidebar describes these in more detail. The chip keys are burned into electrically programmable fuses—for example, the electronic fuse unit in Sun's Niagara 2 processor[9]—to prevent multiple activation attempts.

To activate a chip, the fab must establish a secure link with the IP rights holder and transmit the RCK-Pub for the chip being activated. EPIC's protocol uses the fab's private key to authenticate the transmission. Extensions to this protocol may send a time stamp, serial number, and so on.

In response, the IP rights holder sends the input key (IK), which represents the CK encrypted with the PCK-Pub and then signed by the MK-Pri. The ordering of encryption and signing of the CK to produce the IK is crucial so that entities other than the IP rights holder cannot produce IKs, even if the CK is compromised. Using the RCK-Pub to encrypt communications makes statistical attacks against the MK-Pri more difficult. The IP rights holder can use the fab's public key to additionally encrypt the resulting IK so that only the fab can receive it.

The chip decrypts the IK using the RCK-Pri and MK-Pub, which authenticate it as being sent by the IP rights holder. Upon decryption, the CK is produced, which unlocks the chip and facilitates testing. After testing, the chip can be sold.

COMBINATIONAL LOCKING

To protect a combinational circuit $C(\vec{x})$ with a k-bit key, we have developed a simple procedure that uses k new gates. First, select k wires $\{w_i\}$ and match them with the key's bits $\{y_i\}$. Wires should avoid critical paths and congested regions; matching can minimize wire length. Inputs and outputs of flip-flops are often on critical paths and not as numerous as internal wires. For each selected wire w_i, disconnect its driver from the sinks and insert either an XOR gate $w'_i = w_i \oplus y_i$ or an XNOR gate $w' = w_i \overline{\oplus} y_i$, 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 versus XNOR gate depends on the matched key bit's chosen value: if y_i is 0, $w'_i = w_i \oplus y_i$; otherwise, $w'_i = w_i \overline{\oplus} y_i$. Using the identity $w_i \oplus y_i = \overline{w_i} \overline{\oplus} y_i$ to complicate reverse-engineering, a chip designer or software tool can replace chosen XOR gates with XNOR gates and inverters and, similarly, XNOR gates with XOR gates and inverters, moving inverters along fan-in or fan-out wires using de Morgan's law.

Figure 2 shows an example of a combinationally locked circuit.

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

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

If the uniqueness requirement is 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 a satisfiability (SAT) solver alone is insufficient to find a key combination of nontrivial length.

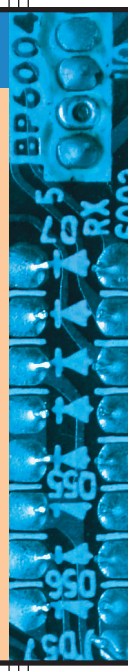
ON-CHIP TRNGS AND PUFS

Randomized algorithms often use pseudorandom number generators (PRNGs)—deterministic sequences with random appearance that are initiated by an input seed. However, cryptographic applications demand true randomness to circumvent attacks based on predictability.

Chips typically generate true random bits by sampling chaotic physical phenomena, such as thermal noise, quantum-mechanical measurement, metastability in latches, and so on. Such TRNGs are a major component in cryptographic applications and can be found in commercial ICs. For example, Sun's forthcoming Niagara 2 processor couples one TRNG in each of its eight cores with cryptographic units to support secure generation of public and private keys.

Physically unclonable functions are another source of true randomness on-chip. PUFs use manufacturing-process variations such as delay differences between logic paths, threshold voltage differences, ring oscillators, and so on and are commonly used to generate secret keys on-chip for cryptographic applications.

EPIC can use either TRNGs or PUFs to define randomized chip IDs upon initial power-up.



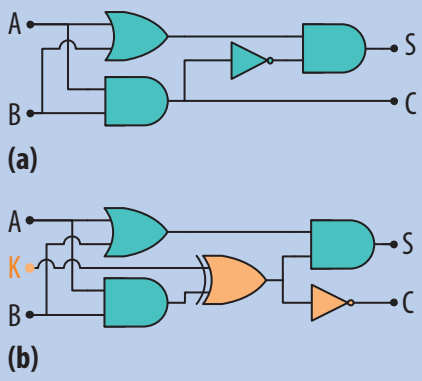


Figure 2. Combinational locking. We add an XNOR gate to a standard half-adder (a) and propagate inverters to produce a combinationally locked half-adder (b). The locked adder is equivalent to the unlocked adder when key-bit K is set to 1.

Reduced ordered binary decision diagrams (ROBDDs), canonical graph representations of logic functions, offer more appropriate tools.[10] Techniques for circuit analysis 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. The resulting ROBDD compactly represents all good key combinations by its paths, which can be counted in time O(size). This formal method can be used to check the key combination's uniqueness, but it may also help forgers to discover the CK if both $C'(\vec{x}, \vec{y})$ and $C(\vec{x})$ are available.

A broad range of attacks becomes possible by tampering with hardware itself, altering the hardware manufacturing process, and subverting the hardware design process.

A key should be long enough to withstand brute-force attacks, which are 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 just one key combination. In practice, most incorrect combinations can be weeded out by scanning in test patterns and comparing a 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 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 $\Omega(2^{k_1+k_2})$.

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

Consider a circuit $C'(\vec{x}, \vec{y})$ such that the key $\vec{y}$ locks n independently testable circuit modules and, for $j = 1 \ldots n$, 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

$$\mathcal{L}(\vec{y}) \log_2 \left(\sum_{j=1}^{n} \frac{2^{k_k}}{G_j} \right) - 1 \qquad (2)$$

In practice, having several good key combinations may be useful—for example, to trace activation by different parties. However, this would decrease the key's effective length. Based on our evaluations, we recommend $\mathcal{L}(\vec{y}) \geq 64$.

VULNERABILITY ASSESSMENT

Research on software security protocols often assumes a specific security model, limiting what attackers can do. However, proofs that make such assumptions are often vulnerable to hardware attacks that violate assumptions. A broad range of attacks becomes possible by tampering with hardware itself, altering the hardware manufacturing process, and subverting the hardware design process.[11] Some of these attacks can be mounted with limited resources, while others require access to multi-billion-dollar facilities and professional design services. Therefore, rather than assume a single security model and risk irrelevance, we have considered various attacker capabilities and what each can achieve.

Our main objective is to protect ICs against piracy through unauthorized excess production and stolen optical IC masks. However, pirates also can steal RTL or gate-level netlists, layouts, and 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. EPIC provides robust multilayered defense against all such attacks.

Consider a given IC design that reads data on inputs $x_1, \ldots, x_m$ and produces data on outputs $z_1, \ldots, z_n$. EPIC adds new inputs $y_1, \ldots, y_k$ such that for a small number of input combinations to $y_1, \ldots, y_k$ the new circuit has the exact functionality of the unmodified one. Our methodology's strength relies on the difficulty of finding correct values for $y_1, \ldots, y_k$, whether the given IC's functionality is known or not. The baseline construction for adding new inputs is combinational locking, which assumes that the same input combination to $y_1, \ldots, y_k$ (the CK) unlocks all chips manufactured from the same mask set.

Attacks against the common key

Simple attacks attempt to find the CK.

Proposition 1. If the protected IC's functionality is unknown, finding the CK for an arbitrary extended circuit is undecidable.

To circumvent this difficulty, the attacker may gain access to the original (unlocked) circuit design and use it to find the CK. Such access is unlikely in practice because the original design is never transmitted and not present in semiconductor masks. However, even with this information the attacker would find it difficult to break EPIC.

Proposition 2. Given the original circuit $C(\vec{x})$ and the locked circuit $C'(\vec{x}, \vec{y})$, finding $\vec{y}$ requires solving Equation 1.

Solving this quantified Boolean formula is, in general, more difficult than NP. Formal methods commonly used in digital circuit verification cannot solve this QBF for 64-bit $\vec{y}$ in reasonable time, and neither can brute-force searching.

Replay and man-in-the-middle attacks

EPIC's baseline construction is susceptible to replay attacks, in which the attacker records the combination

used to unlock one chip and replays it to unlock additional chips. While encrypting all communications related to unlocking can be generally useful, this does not disable replay attacks. To address this problem, as well as man-in-the-middle attacks often used to facilitate replay attacks and obtain secrets, EPIC leverages

- established cryptography concepts—salting and challenge-response authentication;
- recently developed hardware constructs—TRNGs and PUFs;[5,6,12] and
- the RSA cryptosystem, which provides encryption and authentication with nonrepudiation.

Salting is the process of adding random bits to data before encryption to defeat comparisons with encrypted data. EPIC can draw random bits from TRNGs or PUFs, which we treat as black boxes (cannot be viewed or modified) and assume to be cryptographically secure.

EPIC uses challenge-response authentication to defeat man-in-the-middle attacks, in which a perpetrator takes over the communication channel and impersonates one of the original parties. To authenticate its counterpart, each party sends a challenge to the other party and ceases further communications if the expected response is not received. EPIC performs challenge-response authentication using RSA keys (MK-Pub and MK-Pri) and random chip keys (RCK-Pub and RCK-Pri) derived from PUFs or TRNGs.

Proposition 3. Impersonating the IP rights holder is infeasible.

Messages from the IP rights owner are signed using the MK-Pri, which is never transmitted. Well-known results about the strength of RSA show that messages not sent by the IP rights owner will not decrypt correctly using the MK-Pub, even if the attacker knows the MK-Pub. EPIC relies on the properties of PKC, described in the "Public-Key Cryptography" sidebar, to guard against a compromised CK.

Proposition 4. An attacker who knows the CK but not the MK-Pri cannot activate chips using man-in-the-middle attacks (impersonation) alone.

During activation, the CK is not entered directly but produced by decrypting messages from the IP rights holder. The attacker would not be able to send such messages due to RSA guarantees, but if he had the MK-Pub, he could attempt to impersonate a chip and send a fake RCK-Pub* to the IP rights holder. The IP rights holder would respond with the CK signed by the MK-Pri and encrypted with the fake RCK-Pub*. This would expose the CK but not the MK-Pri due to RSA protections, and the attacker would be unable to produce a valid IK for any chip with a RCK-Pub different from the RCK-Pub*.

Proposition 5. In the context where a given mask set and all chips produced from the masks are black boxes, EPIC successfully protects all chips produced from this mask set even if an attacker has access to the original (unmodified) IC design.

Neither exhaustive search for the CK (or its encrypted version) nor formal methods are feasible to unlock a chip. If the attacker has access to information sent by the chip or IP rights holder, he may try to impersonate the chip, causing the IP rights owner to divulge additional information, but such man-in-the-middle attacks are futile against EPIC.

Technologically advanced attacks

More powerful attackers can see and understand locked-chip implementations.

Proposition 6. Having access to an implementation of the locked IC facilitates an attacker's discovery of the CK. However, the CK alone does not allow the attacker to unlock additional chips.

To unlock an arbitrary chip, the attacker must send the CK encrypted with that chip's RCK-Pub and MK-Pri. As shown earlier, knowing the original design's circuit schematics does not expose the MK-Pri and the attack fails.

An attacker may understand the locked IC design and know a running chip's internal logic values on every cycle. Seeing internal signals at runtime exposes the RCK-Pub and the CK, but the attacker can obtain these with less effort if he knows the MK-Pub. The RCK-Pri is available to

PUBLIC-KEY CRYPTOGRAPHY

Cryptography allows remote users to exchange messages through an untrusted medium, in such a way that transmissions intercepted by eavesdroppers do not reveal plaintext. The sender encrypts the plaintext and the receiver decrypts it.

In 1976, Whitfield Diffie and Martin Hellman invented asymmetric cryptography, better known as public-key cryptography.[1] In PKC, each user independently generates a pair of keys, one public and one private. Public keys are made available to everyone, but owners never transmit or reveal their private keys. 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—that is, for number-factoring and discrete logarithms.

The sender (B) encrypts plaintext with the receiver's public key (A) and transmits the message, which can only be decrypted with A's private key. The RSA system proposed by Ron Rivest, Adi Shamir, and Len Adleman in 1977 enriches PKC 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 exchanging symmetric keys that allow faster communication. RSA-style cryptosystems are among the most studied but remain resilient against various attacks 30 years after their inception.

Reference

1. N. Ferguson and B. Schneier, *Practical Cryptography*, John Wiley & Sons, 2003.

INDEPENDENT EVALUATION OF EPIC

Belgian researchers at Katholieke Universiteit Leuven evaluated a preliminary version of EPIC in terms of security and overhead.[1] They found that EPIC is weak 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 the resulting ciphertext signed by the MK-Pri, which is standard protocol for public-key communication with nonrepudiation. On the other hand, if the IK is calculated properly, no successful logic-level attacks against EPIC are known.

The KU Leuven researchers also propose modifications to EPIC to reduce overhead while maintaining security. Having in mind man-in-the-middle attacks that target the CK, the researchers suggest modifying EPIC to either send the CK as plaintext or hardcode it in the enriched RTL. Both of these methods obviate the need to generate the RCK-Pub and RCK-Pri on-chip but require chips to produce a random time-sensitive nonce upon power-up. In addition, embedding the CK on-chip may make it more practical for an adversary to modify a locked chip to subvert EPIC. Thus, it's possible to reduce EPIC's communication and computation (but not hardware) overhead.

Reference

1. R. Maes et al., "Analysis and Design of Active IC Metering Schemes," *Proc. 2009 IEEE Int'l Workshop Hardware-Oriented Security and Trust* (HOST 09), IEEE Press, 2009, pp. 74-81.

the attacker, but this key is only used to decrypt the CK at the end of activation, and the attacker can obtain the CK by other means. Nevertheless, none of this information compromises the MK-Pri, and the attacker cannot unlock a given chip.

Proposition 7. Observing internal signals in running chips does not offer an attacker new opportunities.

Consider an attacker who understands the locked IC design, can make a small number of changes in the locked IC design's masks, and can fabricate new chips from altered masks. If the attacker is not associated with the authorized fab, he cannot communicate with the IP rights holder if EPIC is augmented with fab keys (public and private) that authenticate when activation starts and additionally encrypt all EPIC communications.

Proposition 8. An attacker can circumvent EPIC without communicating with the IP rights holder, having access to unmodified activated chips, or observing signals in a running chip, but only at great cost and with considerable risk.

The attacker would first obtain the MK-Pub by design (mask) inspection, then execute a man-in-the-middle attack to obtain the CK. Next, the attacker would locate the wires that carry the decrypted CK to unlock the circuit, cut these wires, and reconnect each to power or ground to represent the CK. If the attacker cannot find the wires that carry the unencrypted CK, he can stage a randomness removal attack by disconnecting PUF or TRNG wires. If the attacker can subsequently perform a normal activation process on a modified chip, he can then use recorded keys to activate other chips with randomness removed because their "random" chip keys would be identical.

Understanding the entire mask set and knowing the CK is not enough to break EPIC—an attacker also has to modify the masks and produce chips from them. These are fairly different and expensive procedures that require lengthy preparation; a small company may not be able to accomplish either, and for a large company there is significant risk of being caught and denied future contracts. State agencies could pursue or fund such attacks for strategic rather than economic reasons.

EPIC EVALUATION

We evaluated EPIC in terms of its overhead, its impact on traditional design flows, and the difficulty of inserting the XOR gates that implement CKs. We have also analyzed the effectiveness of formal methods and brute-force attacks on EPIC. Other researchers have verified our findings on the resilience of EPIC to attacks, as detailed in the "Independent Evaluation of EPIC" sidebar.

Overhead reduction

EPIC's component overhead includes

- additional pins to enter the IK,
- extra gates and wires to implement combinational locking,
- a TRNG, and
- hardware for RSA public-key cryptography.

As most of the chip remains dormant until activation succeeds, it can multiplex an existing pin to enter the 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 are available on ICs, making its area and power overhead minor. A single TRNG is required, and existing TRNGs are small: 0.036 mm^2 in 130-nm technology.[13]

RSA can also be implemented compactly: Silicon Image sells an RSA implementation that uses fewer than 10,000 two-input gates. Processors also can turn off RSA after activation (no power overhead) without affecting critical paths (no delay overhead). Sun's Niagara 2 processor implements RSA in each of its eight cores, with area overhead less than 1 percent.[9]

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 relatively simple, this step can also be verified using SAT-based equivalence checking.

Empirical evaluation of combinational locking

We developed two methods for counting the number of valid CKs in a circuit when XOR gates have been inserted. The first is a formal technique that builds Equation 1 using ROBDDs and solves for all valid CKs. The second is a brute-force approach that tries every possible CK and checks equivalence with the original circuit using ROBDDs. We evaluated the two techniques by inserting XOR gates into combinational circuits at random and counting valid CKs. All experiments were performed on a 2.4-GHz Opteron processor with 8 Gbytes of RAM.

We extensively tested two arithmetic logic unit (ALU) circuits from the ISCAS-85 suite: c880 (60 inputs, 26 outputs, 383 gates) and c3540 (50 inputs, 22 outputs, 1,669 gates). Surprisingly, the brute-force method was more efficient than the formal method on c880; the formal method used more runtime and memory. On c3540, brute-force was more memory efficient but took longer than the formal method.

In these experiments, we were interested in minimal lengths of EPIC keys that provide sufficient resilience against attacks. In practice, much larger keys can be supported—for example, by using RSA cryptography. For 24-bit and larger EPIC keys, runtime for the formal method increased exponentially, making it infeasible as an attack on EPIC. We estimate that finding a valid 64-bit key on either benchmark would take at least 10^6 CPU years.

We also observed that inserting XOR gates randomly produces relatively few duplicate keys. In our experiments, the valid key was unique for up to 32 bits on the c3540 benchmark. On the c880 benchmark, four of 2^{32} key combinations were valid, which only reduces the effective key length by two bits. For a 64-bit key in c880 to be breakable in less than one 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. We found that random insertion of XOR gates to as many as 1/8 of the gates did not produce many duplicate keys. Therefore, most circuits with 500 gates, as well as by many smaller circuits, can support our suggested key length of 64 bits.

Modern digital ICs exhibit much larger modules, often with hundreds of thousands of gates. Such modules can support keys with several hundred bits. Inserting an n-bit key will require n new XOR gates and n new wires—a modest overhead in area and power given that the key bits do not switch during the circuit's operation. The circuit's critical paths can be spared when inserting XOR gates, or the number of XOR gates per path can be limited to one; this ensures minimal performance overhead.

EPIC's main overhead is in requiring an on-chip PKC implementation. Many high-end chips already include such implementations, which can be reused. In other cases, it is possible to employ compact, low-speed implementations that only use PKC to exchange several hundred bits during activation.

Given any chip, developers can design knockoffs from scratch to closely mimic the chip's behavior. However, this is not economically viable for pirates. Our approach to defeating IC piracy is to render infringement unprofitable by making attacks computationally infeasible. This is accomplished through EPIC, a novel low-overhead combinational chip-locking system and a chip-activation protocol based on PKC. In addition to preventing IC piracy via active hardware metering, EPIC can require the transmission of serial numbers during chip activation.

Circumventing EPIC without modifying the masks or ICs is very difficult because of PKC's strong security guarantees. On the other hand, production-scale modification of fabricated ICs is infeasible today, especially for advanced technology nodes. Mask modifications and other related scenarios require high investment that may not be economically profitable for attackers. To this end, we note that pirated ICs are typically late to market, while enjoying smaller volumes and margins than original ICs. This limits pirates' investment and makes it nearly impossible to justify nonrecurring engineering costs or to gradually ramp up yield on an alternative fab. EPIC can also be applied to modern FPGAs with bitstream encryption by locking combinational cryptographic circuits.

Overall, we hope that by limiting theft, EPIC will improve IC industry economics.

Additional details on research reported here can be found in patent applications filed by the University of Michigan and Rice University with the US Patent Office.

References

1. Defense Science Board, "Defense Science Board Task Force on High Performance Microchip Supply," Feb. 2005, US Dept. of Defense; www.acq.osd.mil/dsb/reports/ADA435563.pdf.
2. P. Clarke, "Fake NEC Company Found, Says Report," *EE Times*, 4 May 2006; www. eetimes.com/showArticle.jhtml?articleID=187200176.
3. R.J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, John Wiley & Sons, 2001.

4. F. Koushanfar, G. Qu, and M. Potkonjak, "Intellectual Property Metering," *Proc. 4th Int'l Workshop Information Hiding*, LNCS 2137, Springer, 2001, pp. 81-95.
5. Y. Su, J. Holleman, and B.P. Otis, "A Digital 1.6pJ/Bit Chip Identification Circuit Using Process Variations," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, 2008, pp. 69-77.
6. G.E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," *Proc. 44th Ann. Design Automation Conf.* (DAC 07), ACM Press, 2007, pp. 9-14.
7. Y. Alkabani and F. Koushanfar, "Active Hardware Metering for Intellectual Property Protection and Security," *Proc. 16th Usenix Security Symp.* (Security 07), Usenix Assoc., 2007, pp. 291-306.
8. Y. Alkabani, F. Koushanfar, and M. Potkonjak, "Remote Activation of ICs for Piracy Prevention and Digital Rights Management," *Proc. Int'l Conf. Computer-Aided Design* (ICCAD 07), IEEE Press, 2007, pp. 674-677.
9. U.G. Nawathe et al., "An 8-Core 64-Thread 64b Power-Efficient SPARC SoC (Niagara2)," presentation, IEEE Int'l Solid-State Circuits Conf. (ISSCC 07), Feb. 2007; www.opensparc.net/pubs/preszo/07/n2isscc.pdf.
10. G.D. Hachtel and F. Somenzi, *Logic Synthesis and Verification Algorithms*, Springer, 2006.
11. J.A. Roy, F. Koushanfar, and I.L. Markov, "Extended Abstract: Circuit CAD Tools as a Security Threat," *Proc. IEEE Int'l Workshop Hardware-Oriented Security and Trust* (HOST 08), IEEE Press, 2008, pp. 65-66.
12. D. Lim et al., "Extracting Secret Keys from Integrated Circuits," *IEEE Trans. Very Large Scale Integration* (VLSI) *Systems*, vol. 13, no. 10, 2005, pp. 1200-1205.
13. C. Tokunaga, D. Blaauw, and T. Mudge, "True Random Number Generator with a Metastability-Based Quality Control," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, 2008, pp. 78-85.

Jarrod A. Roy *is an advisory software engineer at IBM Corp. His research interests include VLSI placement and routing, Boolean satisfiability and QBF solving, and hardware intellectual property protection. Roy received a PhD in computer science and engineering from the University of Michigan, Ann Arbor. He is a member of IEEE and the ACM. Contact him at jaroy@us.ibm.com.*

Farinaz Koushanfar *is an assistant professor in the Electrical and Computer Engineering and Computer Science departments and the director of Texas Instruments DSP Leadership at Rice University. Her research interests include hardware and embedded systems security; intellectual property protection; content and data integrity; low power, distributed embedded systems; and statistical modeling and optimization. Koushanfar received a PhD in electrical engineering and computer science from the University of California, Berkeley. She is a member of IEEE, the ACM, and the American Association for the Advancement of Science. Contact her at farinaz@rice.edu.*

Igor L. Markov *is an associate professor in the Computer Science and Engineering Department at the University of Michigan, Ann Arbor. His research interests include computers that make computers (software and hardware), secure hardware design, and combinatorial optimization with applications to the design, verification, and debugging of integrated circuits and quantum logic circuits. Markov received a PhD in computer science from the University of California, Los Angeles. He is a senior member of IEEE and the ACM. Contact him at imarkov@eecs.umich.edu.*