Analysis and Design of Active IC Metering Schemes

Roel Maes*, Dries Schellekens*, Pim Tuyls*† and Ingrid Verbauwhede*
* Katholieke Universiteit Leuven, ESAT-SCD/COSIC and IBBT
Kasteelpark Arenberg 10, 3001 Heverlee, Belgium
tel.: +3216321050, fax: +3216321969
Email: firstname.lastname@esat.kuleuven.be
† Intrinsic-ID
High Tech Campus 9, 5656 AE Eindhoven, The Netherlands
Email: pim.tuyls@intrinsic-id.com

***Abstract*—Outsourcing the fabrication of semiconductor devices to merchant foundries raises some issues concerning the IP protection of the design. Active hardware metering schemes try to counter piracy of integrated circuits by enforcing the fabrication plant to run an activation protocol with the IP owner for every chip that is produced. In this work, we analyze the protocols of two active hardware metering schemes that were recently proposed by Roy *et al.* in [1], [2]. We study how these schemes achieve security and based on this, we suggest more efficient and secure versions for both. Finally, we present a simplified and secure activation protocol based on physically unclonable functions.**

***Index Terms*—Intellectual property protection, active metering, activation protocol, physically unclonable function.**

I. Introduction

For almost halve a century now, digital computing power has risen exponentially over time and it continues to do so. This trend was empirically observed and formulated by Gordon Moore in 1965 and is hence best known under the name Moore's law. In recent years, advances in photolitography techniques for producing integrated circuits have scaled down the feature sizes on chips from 90nm in 2003-2004 downto 65nm in 2006-2007 and 45nm in 2007-2008, and 32nm manufacturing processes are currently being tested. However, the cutting edge technology needed to implement these processes has pushed the cost of a new semiconductor fabrication plant or *fab* towards several billions of dollars. This enormous investment gave rise to the so-called *foundry model*, where *fabless* semiconductor companies only design devices and the manufacturing is done by *merchant foundries* that only perform the actual production.

Outsourcing the manufacturing of advanced semiconductor devices raises issues concerning the IP protection of the innovations contained in the design files. Once a design house hands over its plans to a fab, it has no real control over further actions. Some legal restrictions, *e.g.* about the allowed production volume, are contained in a contract, but a technical enforcement of these restrictions is non-existing. Piracy of integrated circuits becomes a main industry problem and three major acts of piracy can be discerned:

1) The agreed production volume can be exceeded and the excess production is sold on the grey/black market at high profits, since the fab does not suffer from high non-recurring engineering or NRE costs. This is known as **overbuilding**, and can be done by a dubious manufacturer, but also by an underground sister company of a renowned fab.
2) A fab that does not possess any designs can **reverse engineer** them from a manufactured device. In that case, he can add changes or reuse intellectual property in its own products which he can sell at high profit, again without baring substantial NRE costs.
3) A malicious manufacturer might try to obtain the design files in an illegal manner, *e.g.* through theft, espionage or reverse engineering and **clone** the chip in order to sell it.

Another evolution in the development of digital systems that was induced by the high cost of ASIC production as well as the increasing NRE costs for a new design, is the use of programmable hardware devices, most notably field programmable gate arrays or FPGAs. Instead of having a hardware description manufactured

978-1-4244-4804-3/09/$25.00

as a hardwired circuit at a foundry, it can be compiled into a *soft* configuration file that can be loaded on a standard FPGA in the field. Since most commercial FPGAs are volatile, they have to be accompanied by a non-volatile memory, *e.g.* Flash, to store this configuration file when the FPGA powers down. This makes FPGA designs also very vulnerable to cloning, since the configuration file can easily be obtained from the memory, *e.g.* by eavesdropping on the configuration bus. Once an adversary possesses this file, he can clone the entire design without effort, just by loading it onto another FPGA of the same type.

It is clear that technical restrictions to overcome piracy of integrated circuits have to be implemented. Moreover, since larger designs of systems-on-chip even involve intellectual property from multiple parties, a more fine-grained structure of intellectual rights management will be necessary, allowing each party to control the use of its IP blocks. More advanced systems could even restrict the number of times or the duration of an activation of individual IP blocks on a chip.

A. Previous work

To actively prevent the acts of overbuilding and cloning, a number of techniques collectively known as *active hardware metering* have been proposed. Active metering schemes for FPGAs rely on cryptographic transformations of the configuration file, as proposed by Simpson and Schaumont [3] and improved by Guajardo *et al.* [4], [5], [6]. To implement active metering on an ASIC, some sort of locking mechanism needs to be in place, which makes sure the chip is in a locked state upon production. By running a cryptographic protocol with the IP owner, the fab can enable the chips. In this way, the IP owner has full control over the production volume. Locking techniques based on FSM obfuscation have been introduced in [7], [8], [9], based on combinational locking in [1] and on scrambling of system busses in [2].

Many of the proposed schemes rely on a physically unclonable function (PUF) [10], [11], [4], [12] to derive a unique chip identifier.

B. Contributions

In this work, we analyze the activation *protocols* of two previous proposals by Roy *et al.*, based on combinational locking [1] and bus scrambling [2]. It is shown that the protocol from [1] is not secure against a malicious fab, even without the need to modify the chip design. Moreover, both protocols can be simplified and implemented in a more efficient way. Finally, we propose a simple and secure activation protocol based on PUFs. It is important to note that we did not investigate, nor make any judgement about the IC locking mechanisms of the proposals in [1], [2]. In fact, we show that the security of the whole scheme is independent of the difficulty of reverse engineering the used locking mechanism. In [1], the unlocking key cannot be kept secret and the security only depends on the verification of a digital signature. In [2], the security depends on the secrecy of a one-way function embedded in the chip design.

II. Setting

A. Protocol Parties

In hardware metering protocols, we discern three major parties, *i.e.*:

- The holder of the IP rights on the IC design to be manufactured, further called the *IP Owner*. We assume that the IP Owner possesses the *design plans* of the IC and has mask sets produced from these plans.
- The merchant foundry, further called the *Manufacturer*, that will manufacture the ICs using the mask sets it obtained from the IP Owner.
- The manufactured integrated circuit, further called the *Chip*, of which we want to meter the production volume.

The basic idea behind all hardware metering protocols is that, at some point, the Chip and the IP Owner will communicate, giving the latter the possibility to passively monitor or even actively control the production volume.

B. Adversary Model

As is explained in the introduction, the most powerful adversaries are merchant foundries with direct access to the original mask sets and the incentive to perform overbuilding. As an adversary, we will model such a malicious Manufacturer.

1) In active hardware metering, the Chip needs to be activated right after manufacturing, while still in the possession of the Manufacturer. Therefore, we assume that the adversary can effortless and unnoticeably eavesdrop on all communications between the IP Owner and the Chip. Moreover, the adversary can easily change transmitted messages, trying to actively attack the protocol.
2) The Manufacturer necessarily has access to the full mask set needed to manufacture the Chip. This mask set contains a full geometrical description of the Chip's design. It is assumed that reverse engineering a small part, *e.g.* a hard wired key, of the

geometrical mask set to a higher level description (RTL-level or even higher) is not easy, however not impossible. If needed, the Manufacturer will spend some effort/time/money doing it. A large or full reverse engineering of the entire mask set is assumed to be too expensive.

3) It is assumed that the Manufacturer will not alter the mask set. Although not impossible in practice, the Manufacturer is unwilling to bare the high cost of having a new mask set produced. Equivalently, we assume that it is infeasible for the adversary to alter the circuit on *every* Chip post-manufacturing, *e.g.* using a focussed ion beam. These assumptions are also made in [1], [2] and are crucial in any method of hardware metering, since an attacker with full knowledge of the circuit design and the possibility to change the mask or circuit at will, can easily remove or bypass the metering mechanism.

It is clear that in this practical security setting, we cannot solely rely on mathematical complexity assumptions, *e.g.* the hardness of factoring, as in theoretical cryptography. Instead, we also have to make some assumptions about the cost of certain production and analysis techniques, *e.g.* the price of producing a new mask set, with respect to the incentive of the adversary, *i.e.* making *profit* out of overbuilding a design.

III. EPIC: IC Metering through Combinational Locking

A. Protocol Description

In the EPIC scheme [1], the active metering of ICs is based on a *combinational locking* mechanism, which means that the original combinational logic is altered in such a way that it will only operate properly when the appropriate *common key* CK is applied. The IP Owner generates a master keypair $(\mathsf{MK_{pub}}, \mathsf{MK_{pri}})$ and enriches the Chip's design with support for public key cryptography and combinational locking, a true random number generator (TRNG) and $\mathsf{MK_{pub}}$. He generates a random key CK and locks the enriched design with it to obtain a locked design. He sends a mask set of the locked design to the Manufacturer for production. After production, every Chip can be unlocked according to the protocol described in Fig. 1.

In this protocol:

- $\{\mathcal{G}, \mathcal{E}_{.}[.], \mathcal{D}_{.}[.], \mathcal{S}_{.}[.], \mathcal{V}_{.}[.]\}$ is a standard set of public key algorithms such that $\mathcal{G}$ generates a random public-private keypair: $(\mathsf{K_{pub}}, \mathsf{K_{pri}}) \leftarrow \mathcal{G}$. The algorithm $\mathcal{E}_{\mathsf{K_{pub}}}[m]$ encrypts a message m with the public key and $\mathcal{D}_{\mathsf{K_{pri}}}\left[\mathcal{E}_{\mathsf{K_{pub}}}[m]\right] = m$ is the decryption with the corresponding private key. The algorithm $\mathcal{S}_{\mathsf{K_{pri}}}[m]$ signs a message m with the private key and $\mathcal{V}_{\mathsf{K_{pub}}}\left[\mathcal{S}_{\mathsf{K_{pri}}}[m]\right] = m$ is the verification of the signature with the corresponding public key[1].
- $\mathcal{R}$ is an RTL-level description of a chip design, $\mathcal{R}^{+}$ is the enriched design and $\mathcal{R}^{\#}$ is the locked (enriched) design. $\mathsf{Mask}\{\mathcal{R}^{\#}\}$ is a mask set implementing the description $\mathcal{R}^{\#}$.
- One-Time-Programmable or OTP memory is used to store data in a non-volatile way.
- $\{\mathcal{L}_{\mathsf{CK}}[\mathcal{R}^{+}], \mathcal{U}_{\mathsf{CK}}\left[\mathcal{R}^{\#}\right]\}$ is the combinational IC-locking mechanism as described in [1]. $\mathcal{L}_{\mathsf{CK}}[\mathcal{R}^{+}]$ locks the enriched RTL-description $\mathcal{R}^{+}$ with a *common key* CK and generates a locked design $\mathcal{R}^{\#}$. By applying the correct key CK to the unlocking mechanism of a locked Chip (*i.e.* $\mathcal{U}_{\mathsf{CK}}\left[\mathcal{R}^{\#}\right]$) the behavior of the protected $\mathcal{R}^{\#}$ will be unlocked and hence identical to that of the original description $\mathcal{R}^{+}$.
- the communication between IP Owner and Manufacturer is secured and authenticated (not shown in Fig. 1). This is however not important, since it is assumed that the Manufacturer is the main adversary in this scheme.

B. Security Analysis

It is clear that in the EPIC protocol, the common key CK cannot be kept secret from the Manufacturer. Instead of transferring the Chip's public key $\mathsf{RCK_{pub}}$, a malicious Manufacturer with the objective to perform overbuilding can transmit the public key from his own asymmetric key pair $(\mathsf{Man_{pub}}, \mathsf{Man_{pri}})$ to the IP Owner because the IP Owner has no way of checking the authenticity of this communication. Using his own private key $\mathsf{Man_{pri}}$ and the public master key $\mathsf{MK_{pub}}$, which is assumed to be known, the Manufacturer can decrypt IK to obtain CK. This works, independently of the order of the encrypting and signing of CK, which is not entirely clear from the protocol description in [1]. In the light of this observation, it is clear that the alleged security from the EPIC scheme rests on the impossibility for the Manufacturer to forge a valid signature on CK, since he does not know the private master key. This gives rise to some issues.

The Chip uses $\mathsf{MK_{pub}}$ to verify the signature, and hence the security is based on the *integrity* of the public

[1]This notation assumes signature verification with message recovery.

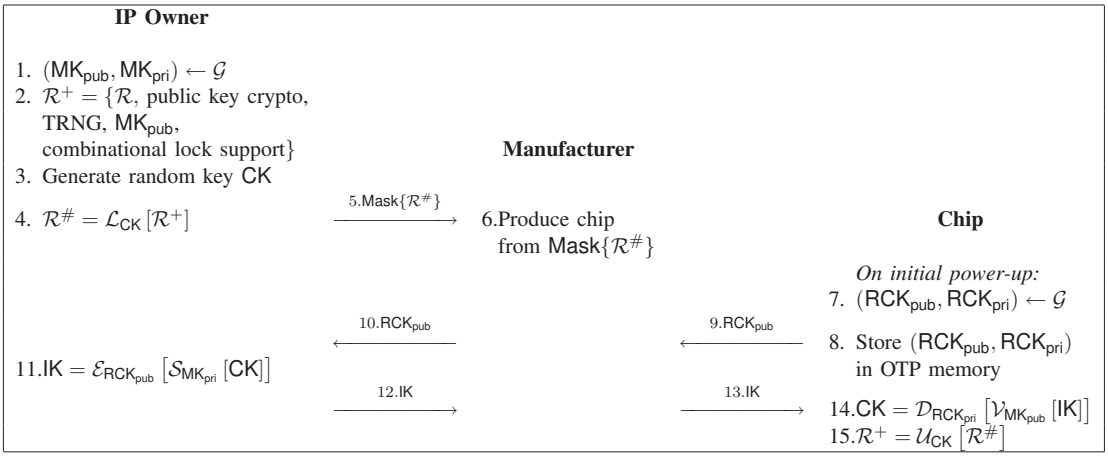


Fig. 1. The EPIC protocol.

master key contained in the chip design. Altering $\mathsf{MK}_{\mathsf{pub}}$ requires producing a new mask set or tampering with every Chip post-manufacturing, which we considered to be too expensive to be profitable for the Manufacturer. Equivalently, the scheme can also be broken if the Manufacturer can hard-wire the known CK in the Chip, which causes every Chip to be automatically unlocked upon production. This is considered infeasible for the Manufacturer for the same reasons.

The EPIC scheme, as shown in Fig. 1, can however be broken without the need to modify the design. The Manufacturer cannot forge $\mathcal{S}_{\mathsf{MK}_{\mathsf{pri}}}[\mathsf{CK}]$ in step 11, but using his own keypair as described above, he can *obtain* this signature, which is the same for every Chip. He then can generate the input keys IK himself by encrypting the obtained signature with the public key $\mathsf{RCK}_{\mathsf{pub}}$ outputted by the Chip. This way, he can keep on activating additional chips without further involvement of the IP Owner. This attack only assumes an active participation of the Manufacturer in the metering protocol, which is much easier than modifying the design and feasible according to our adversary model! In more recent work [13], the authors of the EPIC protocol mention the possibility of *replay* and *man-in-the-middle attacks*, however, they fail to address this weakness in their protocol.

C. Suggested Improvements

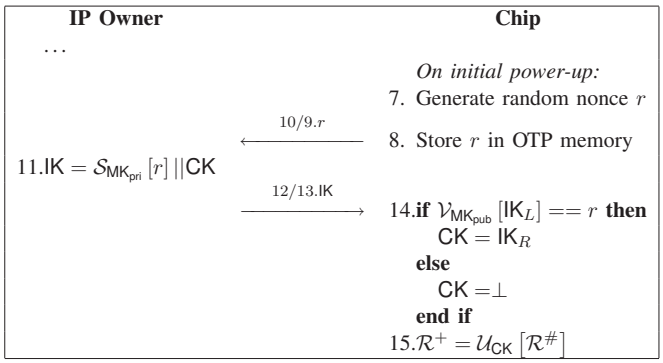
The protocol in Fig. 1 is derived from the informal description given by Roy *et al.* [1]. As observed above, this protocol is insecure, because the signature generated by the IP Owner is fixed and can be *replayed* by the Manufacturer once he obtains it. An obvious solution for this would be to introduce some *freshness* in the signature to avoid replay. A possible way of doing this is switching the encryption and signing of CK in step 11 of the protocol, and the corresponding decryption and verification in step 14. As noted above, the order of these executions is not completely clear from the informal protocol description in [1], but a reader is more likely to understand it as in our interpretation given in Fig. 1.

Switching encryption and signing of CK secures the EPIC scheme against the protocol level attack. However, a more simple and efficient protocol providing the same security notion is possible in that case. We suggest an improvement based on two observations:

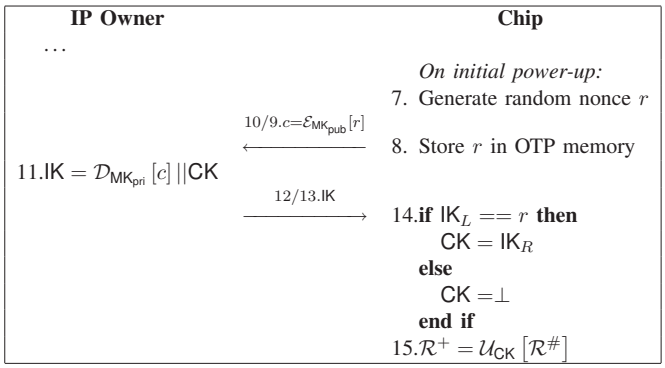
1) The IP Owner fails to transfer CK to the Chip whilst maintaining confidentiality towards the Manufacturer[2]. Therefore, it is superfluous to encrypt CK and it can as well be submitted in clear.
2) The security of the scheme is only based on the impossibility of the Manufacturer to forge a valid signature, *and* to obtain a valid signature that he can reuse. There is no need for the Chip to generate its own asymmetric keypair.

Two simplified but secure variants of the EPIC protocol are shown in Fig. 2. In the protocol in Fig. 2(a), the IP Owner uses his private master key to place a signature on a randomly generated nonce and in the protocol in Fig. 2(b) he uses $\mathsf{MK}_{\mathsf{pri}}$ to decrypt a nonce that the Chip encrypted with $\mathsf{MK}_{\mathsf{pub}}$. In both cases, involvement of the IP Owner in the activation of every Chip is insurmountable, as long as the freshness of the nonce and the integrity of $\mathsf{MK}_{\mathsf{pub}}$ are guaranteed. Also note that CK is not encrypted anymore, but just concatenated

[2]The ability of the adversary to easily learn CK is also acknowledged by the authors of the EPIC protocol in more recent work [13]. On the other hand, they still put much effort in showing that CK withstands brute-force attacks, which seems redundant in that case.



(a) The IP Owner signs a nonce with its private master key.



(b) The IP Owner decrypts a nonce with its private master key.

Fig. 2. Suggested improvements on the EPIC IC activation protocol. Steps 1 to 6 are identical to the original EPIC protocol shown in Fig. 1. The role of the Manufacturer is not shown.

$(||)$ to the signature/decryption to form the input key IK. The parts IK_L and IK_R respectively signify the left and right part of this concatenation. If the check of the signature/decryption on the Chip fails, the Chip will not use the received CK and it fails to unlock ($\mathsf{CK} = \perp$).

An interesting remark is that, since CK is the same for every produced Chip and cannot be kept confidential, it can as well be directly embedded in the enriched design $\mathcal{R}^+$ and loaded by the Chip when the signature/decryption verification is valid. This saves some communication bandwidth. There is no need for the IP Owner to send the same CK in plain to every Chip.

IV. IC Metering through Bus Scrambling

A. Protocol Description

In system-on-chip designs multiple IP modules typically communicate through a common on-chip bus. Roy *et al.* propose an alternative metering scheme based on *scrambling* of the system bus with a symmetric key [2]. The core idea of their bus based protection scheme is that an IC is rendered unusable as long as the IP modules do not share the same bus scrambling key. Before manufacturing the IP Owner introduces a bus scrambling algorithm, which will act as locking mechanism, in the design. Hardware support for a cryptographic activation protocol, using Diffie-Hellman key agreement, a TRNG, and a secret one-way function f, is added as well. A mask set of this enriched design is sent to the Manufacturer for production. On initial power-up the Chip interacts with the IP Owner to generate a Chip specific bus scrambling key BK, which is used by one of the IP modules to scramble its communication with the other modules. The IP Owner can now unlock the Chip by providing the same key BK to the other IP modules. The message flow of the metering scheme is described in Fig. 3.

In this protocol:

- $\mathcal{G}$ represent the generation of appropriate parameters for the Diffie-Hellman key agreement, namely a prime p and generator g of $\mathbb{Z}_p^*$. During the key agreement protocol the IP Owner chooses a random secret a and the Chip picks a random secret b and after exchanging $g^a \bmod p$ and $g^b \bmod p$, they can compute a shared key: $(g^a)^b \bmod p = (g^b)^a \bmod p$.
- f is a secret one-way function that is hidden in the mask set description by the IP Owner before manufacturing. Consequently, this function is only known by the IP Owner and the manufactured Chip. The bus scrambling key is derived from the shared Diffie-Hellman key using f: $\mathsf{BK} = f[(g^a)^b \bmod p]$.
- $\mathcal{R}$ and $\mathcal{R}^+$ are RTL-level descriptions of the original and the enriched design respectively.
- $\{\mathcal{L}_{\mathsf{BK}}[\mathcal{R}^+], \mathcal{U}_{\mathsf{BK}}[\mathcal{R}^+]\}$ is the bus scrambling based IC-locking mechanism as described in [2]. Contrarily to the EPIC scheme, $\mathcal{L}_{\mathsf{BK}}[\mathcal{R}^+]$ does not create a new RTL description. The lock mechanism tells a certain IP module to scramble its bus interface with the key BK, whereas the corresponding $\mathcal{U}_{\mathsf{BK}}[\mathcal{R}^+]$ operation gives the unscrambling key to the other IP modules.

B. Security Analysis

The security of activation protocol heavily relies on the *secrecy* of the one-way function f. If the Diffie-Hellman shared key would be used directly as bus scrambling key (without applying the secret f), the Manufacturer can activate a Chip without contacting the IP Owner. Assuming that (p, g) is known, the Manufacturer can generate its own random a, send $g^a \bmod p$ (after step 7 in Fig. 3) and compute the key $\mathsf{BK} = (g^b)^a \bmod p$. The usage of f prevents this attack.

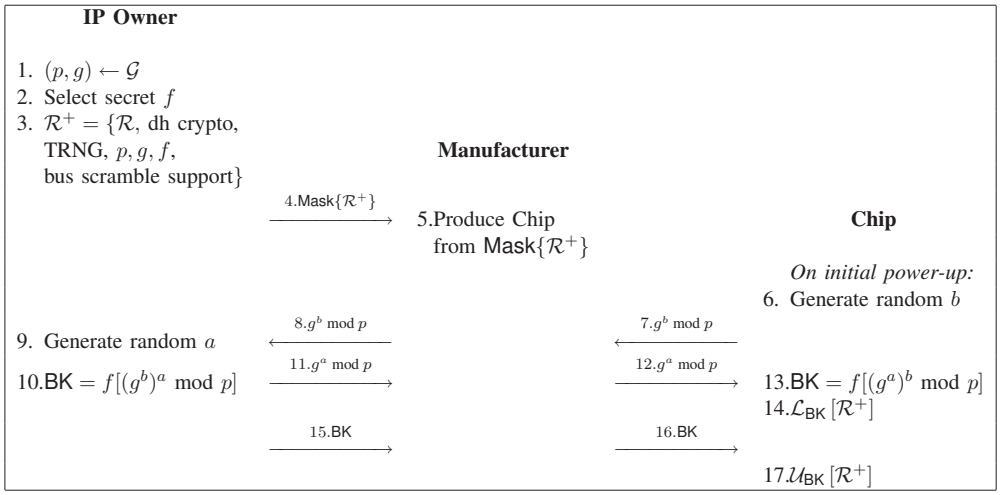


Fig. 3. Bus scrambling based IC activation protocol.

Roy *et al.* do not specify how f should be chosen nor if and how to *"hide"* it on the Chip. According to our adversary model, partial reverse engineering of Mask$\{\mathcal{R}^+\}$ might be feasible to the adversary, hence f should be hidden well enough on the mask to be hard to find. It is not entirely clear if this is possible, but for the remainder of this Section, we will assume that it is infeasible for an adversary to reverse engineer Mask$\{\mathcal{R}^+\}$ upto the full recovery of the functionality of f.

An interesting observation is that the Chip might be in an unlocked state at the beginning of the activation protocol. The scrambling key will typically be stored in volatile (*e.g.*, flip flops) or non-volatile memory (*e.g.*, one-time-programmable fuses). It is likely that this memory is initialized to the same value (*e.g.* BK $= 0$). This implies that the Chip is fully functional until the key BK is programmed. The Manufacturer can potentially exploit this behavior by aborting the protocol before the Chip gets locked in step 14.

C. Suggested Improvements

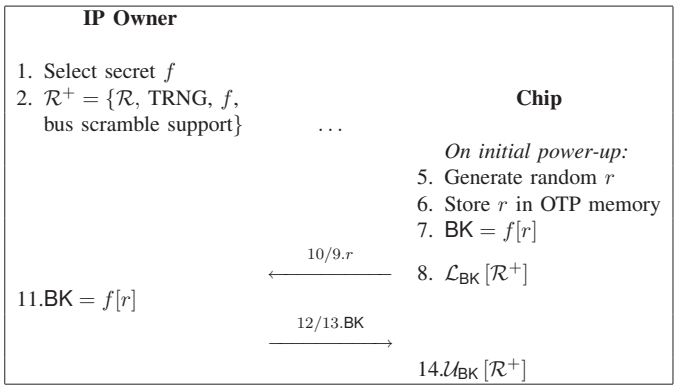
Although the scheme does not suffer from inherent security problems like the EPIC protocol, this activation protocol can also be simplified and improved.

As observed above, the scheme assumes that the IP Owner is able to hide a secret one-way function in Mask$\{\mathcal{R}^+\}$. However, given this assumption there is no need to use a Diffie-Hellman key agreement since there already is a shared secret between the IP Owner and a manufactured Chip. Therefore the protocol can be simplified to the version described in Fig. 4(a). The Chip generates a random number r and sends it to the IP Owner in the clear. Next it evaluates the secret function f with r as input and uses the output BK $= f[r]$ as bus scrambling key. Finally, the IP Owner performs the same function evaluation and provides BK to the other IP modules on the bus. During the activation of the Chip, the Manufacturer learns one evaluation of the secret function f. However, this knowledge is insufficient to active other chips, as they will generate a different unpredictable value r.

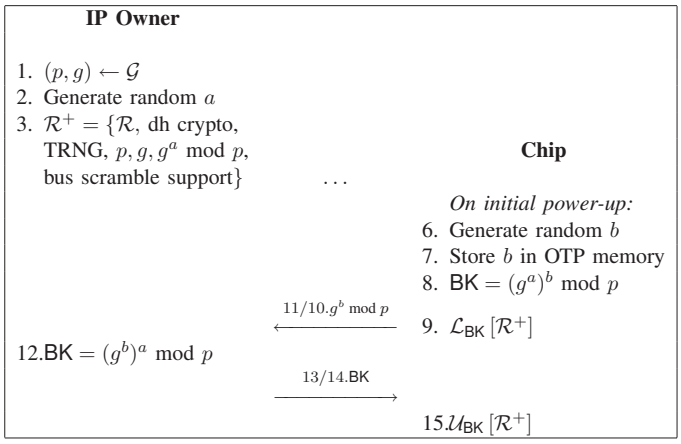
Fig. 4(b) describes an alternative simplification of the protocol without the need for a confidential one-way function. The suggested improvement uses *ElGamal key agreement*, a one-pass Diffie-Hellman variant which provides unilateral key authentication [14]. The IP Owner uses a fixed exponent a and embeds the corresponding exponential $g^a \bmod p$ in the (enriched) design. The contribution of the IP Owner in the key agreement will be the same for all manufactured ICs, but the resulting key BK is still unique for every Chip, as it depends on the random exponent b. The security of this scheme relies on the integrity of the IP Owner's public exponential that is embedded in the design. As explained earlier, changing this value is infeasible to the adversary. In both variants the key BK does not depend on input from the IP Owner and consequently the Manufacturer cannot block the locking operation by aborting the protocol early.

V. PUF BASED ACTIVATION PROTOCOL

In the previous sections we described the activation protocol of two active metering schemes and proposed improvements to increase their security and reduce their complexity. In this section we will illustrate how



(a) Using secret one-way function f.



(b) Using ElGamal key agreement.

Fig. 4. Simplified variants of the bus scrambling based IC activation protocol. The role of the Manufacturer is not shown.

physically unclonable functions can be used to further strengthen and simplify cryptographic IC metering.

A. Security Assumption

In general, the security of a cryptographic activation protocol can rely on two assumptions: (1) the integrity of the IP Owner's public key embedded in the IC or (2) the confidentiality of a shared secret key hidden in the Chip. The first assumption enables a confidential channel from the Chip to IP Owner and an authenticated channel from the IP Owner to the Chip, while the latter allows confidential and authenticated communication in both directions.

As is clear from our adversary model in Section II-B, assuming the existence of a shared secret key between the IP Owner and the Chip might be presumptuous, as the Manufacturer is able to partially reverse engineer the mask set. For this reason, we assume that schemes based on the *integrity* of a public key provide a *stronger* security guarantee than symmetric protocols. In order

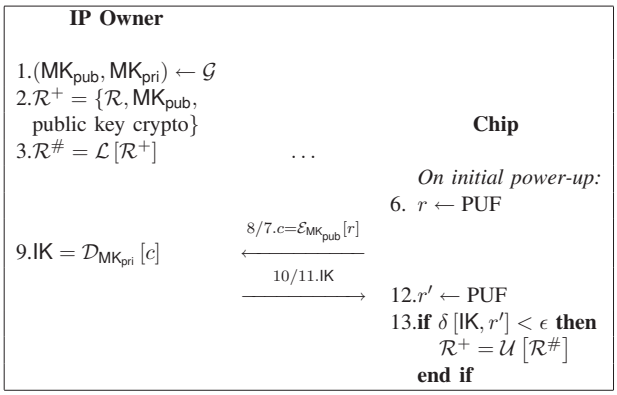


Fig. 5. PUF based activation protocol. The role of the Manufacturer is not shown.

to break this type of schemes, the adversary first has to discover the public key and next substitute it with its own, which we assessed to be infeasible for an adversary with the incentive to make profit.

B. Locking Mechanism

The choice to solely rely on the integrity of the IP Owner's public key has implications on the locking mechanism. It is impossible for the IP Owner to deliver an activation key in a confidential manner. For this reason, we propose to not send any unlocking secret at all during the activation protocol. If the locking technique depends on an unlocking key, as in [1], [2], it should just be embedded in the enriched design. Alternatively, the Chip can be made non-operational using a simple internal enable signal.

C. Physically Unclonable Function as Chip Identifier

The improved activation protocols described in the previous sections follow a similar structure: (1) the Chip generates a random number acting as a challenge and (2) the design is unlocked if a correct response is received from the IP Owner. This challenge-response nature implies that an online connection is present between the IP Owner and the Chip.

Offline activation of IP cores can be accomplished by storing the challenge and response of the initial online protocol in non-volatile memory. The security requirements for this non-volatile storage are different for both messages:

- The challenge must be stored in one-time-programmable memory (*e.g.* fuses) *inside* the IC in order to guarantee its integrity.
- The response can be stored in untrusted storage, because its authenticity is protected cryptographically.

Physically unclonable functions provide an alternative means to randomly generate and persistently store the challenge. This is illustrated in Fig. 5, which describes a PUF based variant of Fig. 2(b). It is important to note that a PUF response is unique for every Chip, but its measurement is *noisy*. Therefore, during an offline verification of the activation code IK the Hamming distance δ between the new and initial measurement of the PUF response, r' and r respectively, must be bounded.

We remark that an equivalent scheme can be constructed using a TRNG and non-volatile memory instead of a PUF. However, the use of a PUF alleviates the need for *on-chip* non-volatile memory, which is not always available. Moreover, it enhances the resilience against physical attacks. Experiments with coating PUFs [11] have demonstrated that invasive attacks substantially alter the PUF's response behavior.

VI. Conclusion

We analyzed the cryptographic activation protocols of two IC metering schemes by Roy *et al.* The first scheme uses a combinational locking mechanism and relies on the integrity of a public key for secure activation. In the second proposal the system bus is scrambled such that the chip is non-functional on start-up. The security of the bus-scrambling protocol to (re-)activate the IC depends on the confidentiality of a secret function embedded on the chip.

We suggest improvements to simplify both schemes and to fix a security vulnerability in the EPIC scheme. Finally we present a new activation protocol using physically unclonable functions.

Acknowledgments

This work was in part supported by the IAP Program P6/26 BCRYPT of the Belgian State, by K.U.Leuven-BOF funding (OT/06/04), by the FWO project G.0300.07 (Security components for trusted computer systems) and by the European Commission through the IST Programme under Contract IST-027635 OPEN_TC. The first author's research is funded by IWT-Vlaanderen under grant number 71369.

References

[1] J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending Piracy of Integrated Circuits," in *Design, Automation and Test in Europe, DATE 2008, Munich, Germany, March 10-14, 2008*. IEEE, 2008, pp. 1069–1074.

[2] ——, "Protecting Bus-based Hardware IP by Secret Sharing," in *Proceedings of the 45th Design Automation Conference, DAC 2008, Anaheim, CA, USA, June 8-13, 2008*, L. Fix, Ed. ACM, 2008, pp. 846–851.

[3] E. Simpson and P. Schaumont, "Offline Hardware/Software Authentication for Reconfigurable Platforms," in *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, ser. Lecture Notes in Computer Science, L. Goubin and M. Matsui, Eds., vol. 4249. Springer, 2006, pp. 311–323.

[4] J. Guajardo, S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection," in *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, ser. Lecture Notes in Computer Science, P. Paillier and I. Verbauwhede, Eds., vol. 4727. Springer, 2007, pp. 63–80.

[5] ——, "Physical Unclonable Functions, FPGAs and Public-Key Crypto for IP Protection," in *FPL 2007, International Conference on Field Programmable Logic and Applications, Amsterdam, The Netherlands, 27-29 August 2007*, K. Bertels, W. A. Najjar, A. J. van Genderen, and S. Vassiliadis, Eds. IEEE, 2007, pp. 189–195.

[6] ——, "Brand and IP Protection with Physical Unclonable Functions," in *International Symposium on Circuits and Systems (ISCAS 2008), 18-21 May 2008, Sheraton Seattle Hotel, Seattle, Washington, USA*. IEEE, 2008, pp. 3186–3189.

[7] Y. Alkabani, F. Koushanfar, and M. Potkonjak, "Remote Activation of ICs for Piracy Prevention and Digital Right Management," in *2007 International Conference on Computer-Aided Design (ICCAD'07), November 5-8, 2007, San Jose, CA, USA*, G. G. E. Gielen, Ed. IEEE, 2007, pp. 674–677.

[8] Y. Alkabani and F. Koushanfar, "Active Hardware Metering for Intellectual Property Protection and Security," in *Proccedings of 16th USENIX Security Symposium*. USENIX Association, 2007, p. 291306.

[9] ——, "Active Control and Digital Rights Management of Integrated Circuit IP Cores," in *Proceedings of the 2008 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, CASES 2008, Atlanta, GA, USA, October 19-24, 2008*, E. R. Altman, Ed. ACM, 2008, pp. 227–234.

[10] B. Gassend, D. E. Clarke, M. van Dijk, and S. Devadas, "Silicon Physical Random Functions," in *ACM Conference on Computer and Communications Security – CCS 2002*, V. Atluri, Ed. ACM, 2002, pp. 148–160.

[11] P. Tuyls, G.-J. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, and R. Wolters, "Read-Proof Hardware from Protective Coatings," in *Cryptographic Hardware and Embedded Systems – CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, ser. LNCS, L. Goubin and M. Matsui, Eds., vol. 4249. Springer-Verlag, 2006, pp. 369–383.

[12] S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "The Butterfly PUF: Protecting IP on every FPGA," in *IEEE International Workshop on Hardware-Oriented Security and Trust, HOST 2008, Anaheim, CA, USA, June 9, 2008. Proceedings*, M. Tehranipoor and J. Plusquellic, Eds. IEEE Computer Society, 2008, pp. 67–70.

[13] J. A. Roy, F. Koushanfar, and I. L. Markov, "Ending Piracy of Integrated Circuits (to appear)," *IEEE Computer*, 2009.

[14] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, Inc., 1996.