Remote Activation of ICs for Piracy Prevention and Digital Right Management

Yousra Alkabani
Computer Science Dept.,
Rice University

Farinaz Koushanfar
Electrical & Computer Engineering and
Computer Science Depts., Rice University

Miodrag Potkonjak
Computer Science Dept.,
University of California, Los Angeles

***Abstract*— We introduce a *remote activation* scheme that aims to protect integrated circuits (IC) intellectual property (IP) against piracy. Remote activation enables designers to lock each working IC and to then remotely enable it. The new method exploits *inherent unclonable variability* in modern manufacturing for unique identification (ID) and *integrate the IDs into the circuit functionality*. The objectives are realized by replication of a few states of the finite state machine (FSM) and adding control to the state transitions. On each chip, the added control signals are a function of the unique IDs and are thus unclonable. On standard benchmark circuits, the experimental results show that the novel activation method is stable, unclonable, attack-resilient, while having a low overhead and a unique key for each IC.**

I. Introduction

The increasing manufacturing cost of ICs has bolstered the horizontal semiconductor business model, in which design and manufacturing are done by different companies. As a consequence, the sole asset of a design company is the hardware intellectual property (IP), since designers typically do not control the number of ICs manufactured from a design. Also, hardware piracy, the illegal manufacturing of ICs using IPs, is omnipresent.

With the horizontal business model, digital rights management (DRM) is becoming an increasingly challenging problem due to the following factors. (i) The designers have no control over manufacturing of individual ICs from a unique mask. (ii) The IC internals are intrinsically opaque, and hence there is limited controllability and observability inside manufactured ICs. (iii) Hardware piracy of state-of-the-art IPs may only be done at fabrication facilities (fabs), where there are large resources and access to the most advanced tools and techniques. (iv) Finally, there is an asymmetric relationship between the designer and the fab, since the fab has full access to the design files, netlists and test vectors.

To overcome the hardware IP piracy problem, various watermarking [9] and metering protocols have been proposed [4], [5], [1]. IC *metering* involves a set of security protocols that enables the design house to gain post-fabrication control through passive or active counts of produced ICs, through the monitoring of IC properties and use, and through remote runtime disabling. Note that, IP watermarking is not the same as metering. A watermark uniquely identifies each IP, not each IC, and hence the existence of the same mask does not affect the watermarked IP. With *passive* metering, each IC is uniquely registered into a database, so a suspicious chip's ID can be authenticated against the database. In *active metering*, the IDs lock the chip's functionality. Unless the corresponding key is provided, the chip will not be able to operate properly. We propose a new IC activation scheme that works by not just active metering of ICs during the chip activation, but by also remotely enabling the chip's regular operation. We denote the diverse ID generation circuitry by a Diverse Random Unique Block (RUB). For each IC, the block generates a unique RUB output for each RUB input vector.

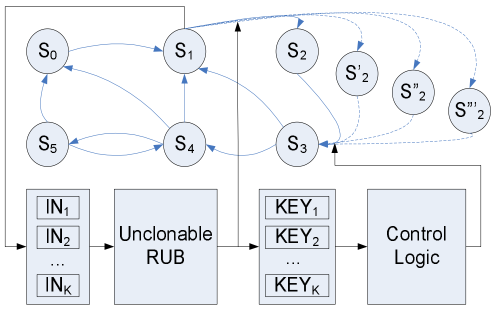

Fig. 1. FSM with a lock on the replicated state (S_2).

In Figure 1, we illustrate the new remote activation method. The FSM of a design with 6 states S_0 to S_5 is shown, where one of the states (S_2) is replicated three times: S'_2, S''_2,S'''_2. Once the design reaches the state S_1, it will transition to one of the four replicated states, depending on the RUB output (which is a function of its specific input selected by the control circuitry). Unless the correct key corresponding to the diverse RUB is provided, the state cannot transition to S_3 (hence it will be locked).

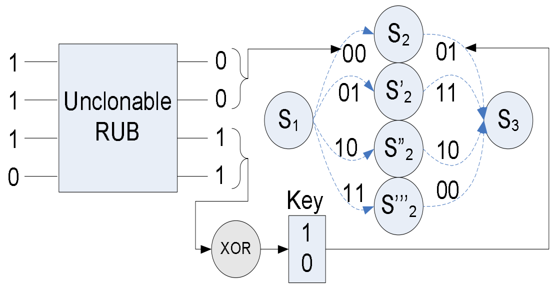

Fig. 2. Close-up of the locking/unlocking mechanism.

Figure 2 shows a closer look into the locking/unlocking mechanism. Assume that there is a two-bit input controlling the edge transitions of the FSM. Assume further that upon reaching the state S_1, some input key with the value 1110 is selected and the corresponding RUB output of the IC under test is 0011. The first two bits define the transition to one of the next replicated states (S_2 in Figure 2). Now, unless the next input is 01, the state cannot reach S_3 and the circuit will be locked. To enable the transition, the next two RUB outputs (11) should be XOR'd with a key that can generate the 01 output (in this case 10 is the key). For each authorized chip, a specific set of RUB inputs will be provided to the fab. Upon manufacturing, the fab will test the output of the specific input set by scanning the FF chains and reporting the stored values. The corresponding output set will be sent to the designer who will provide the corresponding key to unlock the chip. The specific input sets and corresponding unique keys will be stored on the chip to ensure proper operation. In practice, longer inputs and more replicated states will be used to guarantee security.

II. Related Work

Manufacturing variability (MV) has been used for generation of unique on chip IDs [7], [8], [10]; however, the IDs were not integrated into the functionality. The first IC metering scheme was proposed in 2001 [4], [5], but the scheme was passive. Variability-induced delays were used for authentication and security [2], [6], [11], [3]. The underlying mechanism was Physically Unclonable Functions (PUFs) that map a set of challenges to a set of responses, based on an intractably complex physical system. Because MV could cause delay differences among ICs made from the same mask, PUFs would be unique. A database of challenge-response sets is formed for each IC. The chip is authenticated if it can retrieve the output of one or more challenge inputs. Note that, PUFs were only used for authentication against a database, and have not been an integral part of the control path like our scheme.

The first active metering was recently proposed [1]. The method added exponentially many states to the original FSM of the design and exploited a static ID to lock the initial state of each IC. The ICs are unlocked with a key that uses the structural knowledge of the FSM graph to guide the transition from the unique start-up state to the initial functional state. The difference between this work and [1] is that, here, we do not add exponentially many states, but instead we replicate very few number of states. Also, unlike [1], our IDs are not static, but instead are a set of diverse unique IDs extracted from each chip. Perhaps most importantly, our scheme does not just lock the initial reset state of a design, but it also ciphers the functional states and their transitions. Thus, the diverse IDs are continuously checked against the transitions.

III. Remote IC Activation

Key exchange protocol. The remote activation protocol is designed to protect both the designer and the fab by requiring a key exchange mechanism for IC activation. It can be summarized using the following pseudocode.

1) The designer sends the design files to the fab, test vectors, and the number of required copies to the fab.
2) The fab manufactures the required number of ICs, applies the test vectors and sends the IC outputs to the designer.
3) The designer uses the values sent by the fab and computes a key to operate the chip properly.
4) The fab stores the key on the chip and tests the chip in the operational mode.

The remote activation scheme works in the following way.

A. Modification of the FSM. To modify the FSM we replicate a few states a number of times. For instance, if we have states S_0 to S_n, we can pick state S_i and replicate it 4 times to get S_i', S_i'', S_i''', and S_i''''. Note that adding one bit to the state will exponentially increase the number of states making the process of adding states have a low overhead. For each added state all the transitions to and from the replicated state are connected and are a function of a subset of the bits coming from the RUB. The transitions from the replicated states converge to the same state if they have the correct inputs from the RUB. Half the bits responsible for the transitions to and from the replicated states come directly from the RUB. The rest of the bits have to be keyed to be set to the correct value as shown in Figure 2. To maintain proper functionality when the circuit operates, the correct key value is used to cause the transition from one of the copies of S_i to the correct state.

B. The unclonable random unique blocks (RUBs). Lee et al.[6] designed a circuit that generates many unique outputs using selector bits that select various path segments and creates racing paths. We build upon this architecture by adding nonlinearity to make the structure more secure and reverse engineering resilient compared to the original circuit in [6]. The added nonlinearity is not only placed in the delayed paths segments but is also used by the selectors. Figure 3 shows the block diagram of a new RUB circuit. The circuit has nine inputs I_1 to I_9 and three outputs O_1 to O_3. The main building blocks of the RUB are selector elements (Sel), delay elements (D), and arbiters (A). A selector element has two input lines and one selection line. Based on the value of the selection line the inputs either pass directly to the output or are switched. The arbiter gives an output that depends on the input that arrives first (with the shortest delay) at power-up.

C. Input and key memory. The input to the RUB and the key are stored in memory to maintain the proper operation of the circuit. At the initial stage the input memory is loaded with different values to give read out data to the designer to compute the key. Once the designer computes the key and gives it to the fab, the key in the memory will be loaded to ensure that the transitions that occur as a function of the RUB occur correctly and the chip is functional. Figure 2 shows the input and key memory and how they are XOR'd with the output of the RUB for correct functionality.

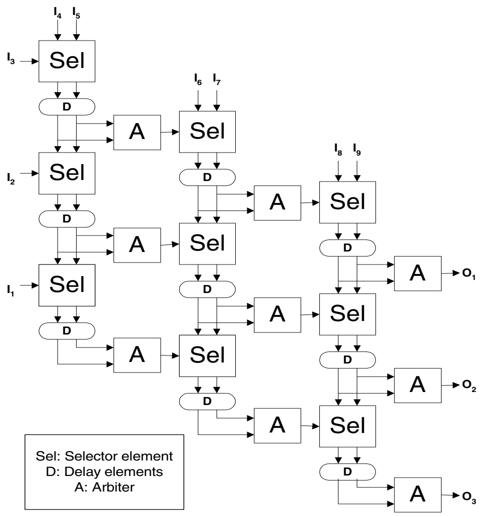

Fig. 3. A non-linear unclonable random unique block.

D. Modification of the FSM control circuitry. The control circuitry of the FSM is modified to ensure that the transitions to the replicated states are function of the output coming directly from the RUB. The transitions from these states are functions of the RUB and the key. For example, they can be combined using a simple XOR as shown in Figure 2.

IV. Attacks

In this section we discuss some of the potential attacks on the remote activation scheme and how we address them.

(i) Brute-force attack. The application of random inputs is not a feasible attack because of the exponentially low probability of correct guessing.

(ii) Reverse engineering of FSM. This attack is infeasible because the extraction of the corresponding state transition graph (STG) is a computationally intractable task.

(iii) Combinational redundancy removal. The attacker may use the combinational redundancy removal, a procedure that attempts to remove the combinational logic that is not necessary for the correct behavior of the circuit. The integration into the functionality of the circuit makes this attack impossible.

(iv) RUB emulation. The goal of this attack is to create a reconfigurable implementation capable of realizing hardware that has the identical functional and timing characteristics of a RUB for which a legal key is already received. However, the state of technology prevents this type of attack.

V. Experimental results

We studied the area, delay, and power overheads and the diversity of the keys generated by the RUB. We used sequential benchmarks from the MCNC'91 set, and Berkeley SIS for synthesis.

A. Area, delay, and power overheads

To modify the original FSM, a C program linked to SIS was written to manipulate the FSM and replicate some states.

TABLE I

Area overhead for adding six extra states for *Random* and *Heuristic* state selection.

	Original			Random		Heuristic	
BM	**PI**	**states**	**area**	**area**	**%**	**area**	**%**
planet	7	48	888	1,373	54.62	1,191	34.12
s510	19	47	605	1,005	66.12	987	63.14
s1494	8	48	859	2,655	209	1,276	48.54
s1488	8	48	880	2,457	179	1,248	41.82
s298	3	135	2,951	5,924	101	3,769	27.72
dk16	2	27	460	964	109.6	898	95.22
sand	11	32	1,092	3,851	253	1,634	49.63
s820	18	24	430	918	113.5	1,040	141.9
s832	18	24	425	927	118.1	1,135	167.1
styr	9	30	633	1,777	180.7	1,306	106.3
Average	10.3	46.3	922.3	2,185	138.4	1,448	77.54

The number of replicated states was set as a parameter. We selected the states for replication using two methods,random selection and by selecting the state with the least number of outgoing edges in original STG. Table I shows the area overhead for the original FSM, the modified FSM with random state selection *Random*, and the modified FSM with states with least edges selected *Heuristic*. The first column shows the names of ten sequential benchmarks from MCNC'91. The second, third, and fourth columns show the number of primary inputs, the number of states, and the area of the original FSM after optimization using SIS. The remainder of the columns shows the area and the percentage overhead for adding six extra states using the *Random* and the *Heuristic* methods, respectively. All the circuits are optimized using the same parameters. The average overhead is 138.4% and 77.5% for the *Random* and the *Heuristic* methods, respectively.

Table II demonstrates the delay and power overheads for the *Random* and *Heuristic* methods compared to the original FSMs. It is interesting to note that in many cases the delay overhead is negative. The average delay overhead is 3.7% and -11.4% for the *Random* and *Heuristic* methods respectively, the power overhead is 148.9% and 84%. The results show that the *Heuristic* method outperforms the *Random* method on average by cutting the overheads into half. Using the *Heuristic* method the average area and power consumption of the modified circuit is less than the double of the original circuit. Because the FSM is usually a very small part of the circuit, a larger area and power overhead is not expected to affect the overall system area and power on a chip.

Figure 4 shows the area, delay, and power overhead of adding different numbers of extra states using the *Heuristic* method. It can be seen that the overheads are nonlinear due to the optimization of the circuits during synthesis. Also it can be seen that the delay overhead decreases as the number of the extra states increases.

B. Diversity of the keys

To study the diversity of the keys produced by the RUB, we simulate a four stages RUB. The RUB has 12 input bits

TABLE II

DELAY AND POWER OVERHEAD FOR *Random* AND *Heuristic* SELECTION METHODS.

	Original		**Random**				**Heuristic**			
BM	**delay**	**power**	**delay**	**%**	**power**	**%**	**delay**	**%**	**power**	**%**
planet	186.2	3,087	93.90	-49.57	4,963	60.77	67.70	-63.64	4,374	41.68
s510	47.60	2,280	81.10	70.38	3,679	61.34	70.90	48.95	3,489	53.02
s1494	115.60	2,958	143.60	24.22	9,435	218.98	84.60	-26.82	4,560	54.15
s1488	134.9	3,011	91.40	-32.25	8,637	186.86	86.70	-35.73	4,450	47.81
s298	201.50	10,798	134.3	-33.35	26,005	140.81	182.60	-9.38	13,896	28.68
dk16	104.70	1,662	87.50	-16.43	3,595	116.32	74.70	-28.65	3,361	102.26
sand	74.80	3,917	88.80	18.72	14,312	265.37	61.50	-17.78	6,084	55.31
s820	36.30	1,430	54.70	50.69	3,054	113.49	47.30	30.30	3,729	160.71
s832	33.60	1,441	44.20	31.55	3,320	130.37	45.40	35.12	4,014	178.50
styr	128.20	2,170	93.40	-27.15	6,401	194.96	68.90	-46.26	4,718	117.40
Average	106.34	3,276	91.29	3.68	8,340	148.93	79.03	-11.39	5,267	83.95

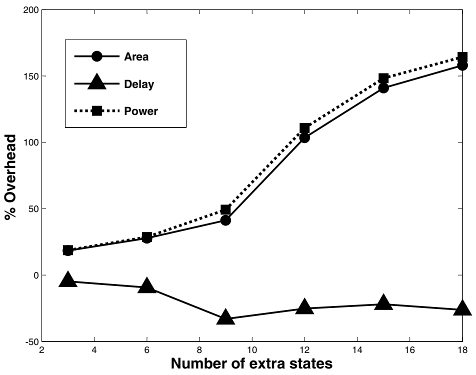

Fig. 4. Area, delay, and power overhead for different numbers of extra states for benchmark s298.

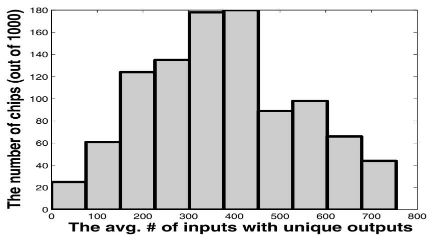

Fig. 5. The average number of inputs producing unique output for 1,000 different RUBs.

and 4 output bits. We randomly generated 1,000 RUBs and calculated the number of inputs generating different outputs for every RUB. Figure 5 shows a histogram of the average number of inputs that have unique outputs for the 1,000 simulated RUBs. The figure shows diversity in the keys generated by the different RUBs.

VI. CONCLUSION

We have developed a new approach for remote enabling, disabling and metering of integrated circuits. The approach leverages inherent manufacturing variability of modern and pending Si technologies. The key conceptual novelty is that designers can control ICs remotely continuously and concurrently with execution. The approach is evaluated in terms of delay, power, and area overheads as well as in terms of the achieved security.

ACKNOWLEDGMENT

This work is supported by the Defense Advanced Research Projects Agency (DARPA)/MTO Trust in Integrated Circuits and Young Faculty Awards (YFA) under grant award W911NF-07-1-0198.

REFERENCES

[1] Y. Alkabani and F. Koushanfar. Active hardware metering for intellectual property protection and security. In *USENIX Security*, 2007.
[2] B. Gassend, D. Lim, D. Clarke, M. Dijk, and S. Devadas. Identification and authentication of integrated circuits. *Concurr. Comput. : Pract. Exper.*, 16(11):1077–1098, 2004.
[3] F. Koushanfar and M. Potkonjak. CAD-based security, cryptography, and digital rights management. In *Design Automation Conference (DAC)*, 2007.
[4] F. Koushanfar and G. Qu. Hardware metering. In *DAC*, pages 490–493, 2001.
[5] F. Koushanfar, G. Qu, and M. Potkonjak. Intellectual property metering. In *Information Hiding Workshop*, pages 81–95, 2001.
[6] J.W. Lee, L. Daihyun, B. Gassend, G.E. Suh, M. van Dijk, and S. Devadas. A technique to build a secret key in integrated circuits for identification and authentication applications. In *Symposium of VLSI Circuits*, pages 176–179, 2004.
[7] K. Lofstrom, W.R. Daasch, and D. Taylor. IC identification circuits using device mismatch. In *International Solid State Circuits Conference (ISSCC)*, pages 372–373, 2000.
[8] S. Maeda, H. Kuriyama, T. Ipposhi, S. Maegawa, Y. Inoue, M. Inuishi, N. Kotani, and T. Nishimura. An artificial fingerprint device (AFD): a study of identification number applications utilizing characteristics variation of polycrystalline silicon TFTs. *IEEE Trans. Electron Devices*, 50(6):1451–1458, 2003.
[9] G. Qu and M. Potkonjak. *Intellectual Property Protection in VLSI Design*. Kluwer, 2003.
[10] Y. Su, J. Holleman, and B. Otis. A 1.6J/bit stable chip ID generating circuit using process variations. In *ISSCC 2007*, pages 406–407, 2007.
[11] G.E. Suh, C.W. O'Donnell, I. Sachdev, and S. Devadas. Design and implementation of the AEGIS single-chip secure processor using physical random functions. In *ISCA*, pages 25–36, 2005.