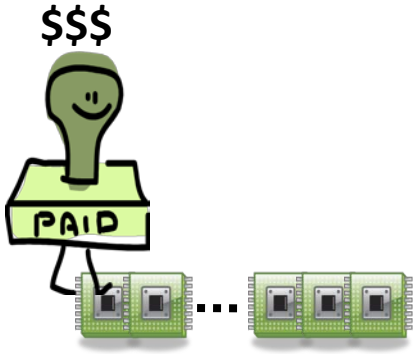
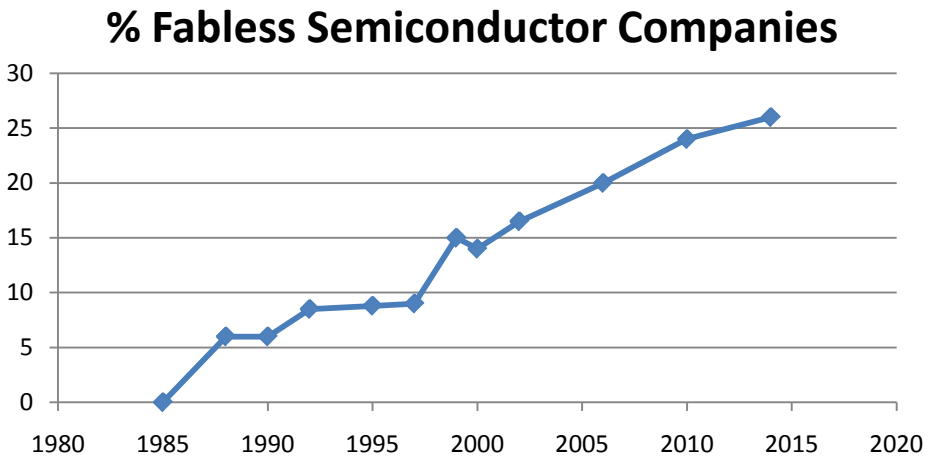
Hardware Metering: A Survey

Farinaz Koushanfar,
ECE Department, Rice University

GLS-VLSI, EPFL
May 3rd, 2011

Hardware IP piracy: $1B/day





- Partly because of exposure of fabless companies at foundries
- Cost of building a full-scale, 300mm wafer 65nm process fabrication (fab) plant is ~$3B and growing
- Asymmetric relationship between the designer ↔ fab
  - The fab has a full access to the IP and the design files
- HW vulnerabilities facilitate software and multimedia piracy

Virtual Socket Interface Alliance (VSIA) and KPMG IP piracy reports

Similar to software piracy*?

- Software is easy to copy
- Activation keys, e.g., MS Office
  - Every CD requires its own key
  - But this key can be copied too
- SW is easy to modify – cracked versions abound
  - E.g., computer games on Bit-Torrent, etc
- **HW is drastically different**
  - No known techniques for physically copying ICs
  - *Reproducing* IC requires masks & access to a fab
  - *Modifying* a chip requires FIB – very slow & expensive (impractical in large quantities)

* Roy, Koushanfar, Markov. EPIC: Ending Piracy of ICs, DATE'08, IEEE Computer'11

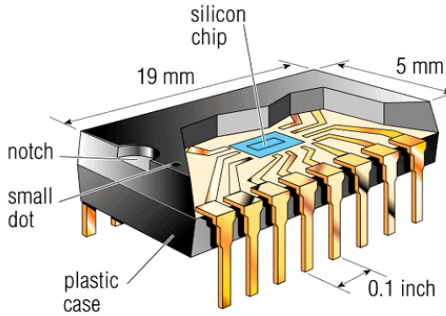
Hardware metering

- HW Metering is a system of security protocols that enables the design house to gain post-fabrication control by
  - Passive or active control of the number of manufactured ICs from one design*
  - The properties of IC and its usage
  - Remote runtime monitoring and disabling
- Unclonable IC identification and authentication
- Access control at the functional behavioral level

*Koushanfar and Qu. Hardware metering. DAC'01

Why is the problem challenging?

- Very little is known about the tampering attacks
- Many possibilities: tampering can done at many levels of abstraction of the synthesis process
- The likely adversaries are financially and technologically strong
- The adversary has a full access to the structural specifications and often to the test vectors
- The internal parts of the manufactured ICs are intrinsically opaque



Metering taxonomy*

- Passive metering
  - Nonfunctional identification
  - Functional identification
- Active metering
  - Nonfunctional identification
  - Functional identification

All metering methods maybe based or reproducible or unclonable identifiers

*Classification by F. Koushanfar, Book Chapter in 'Intro to Hardware Security and Trust, Springer'11

Passive metering: nonfunctional identification

- No clear record of when the IC companies have started to indent IDs on the packages, or a separate piece of ID for storing a digital identifier
- No clear record of when/if the IC companies have used the digital IDs to monitor devices at user's
- Burn-in fuses have been used at the design houses for carving chip IDs
- Intel Pentium III was publicly announced to a include a unique identifier, called the *Processor Serial Number (PSN)*

Intel PSN: the controversy

- The PSN could be used to monitor the user activities via the networks
- Intel made a utility that would give the control over enabling/disabling the PSN to the owners
- It was demonstrated that rogue web sites were able to access even the disabled PSN
- In 1999, consumer privacy groups jointly filed a complaint against Intel with Federal Trade Commission
- Intel decided to not include PSN in future generations

Identification methods

- Reproducible IDs
  - Including indented IDs, burn-in fuses, and a separate non-volatile memory
  - Can be tampered or cloned
  - Subject to foundry attacks
- Unclonable IDs*
  - Unique objects
  - Weak PUFs
  - Strong PUFs

  (Covered in Prof. Verbauwhede's talk)

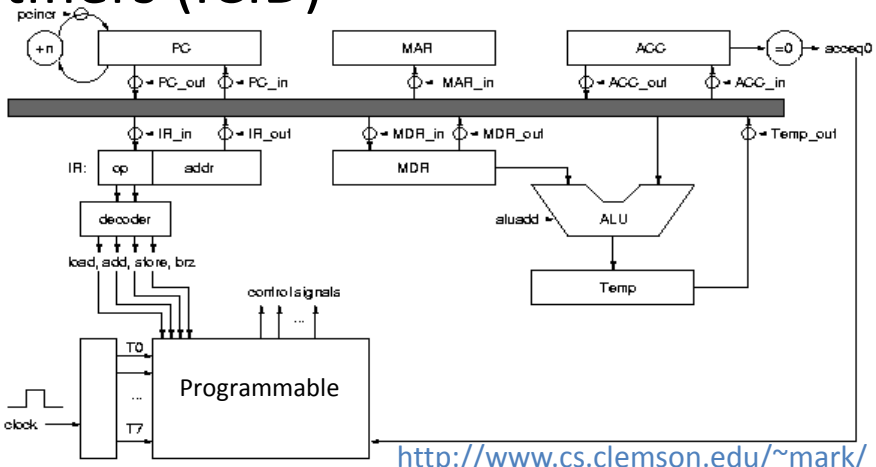
**Classification by Ruhrmair, Devadas & Koushanfar, Book Chapter in 'Intro to Hardware Security and Trust, Springer'11*

Passive functional metering

How can it be done?

Passive metering: functional identification

- The first proposal based on making the control path of each chip unique
- Many control sequences can achieve the same functionality
- During the design, control part (expressed by a finite state machine) was left (programmable)
- Post-fabrication, the programmable part would be programmed to one of the control sequences
- Suggested the programmable part could be made a function of the unclonable identifiers (ICID)



*Koushanfar and Qu. Hardware metering. DAC'01

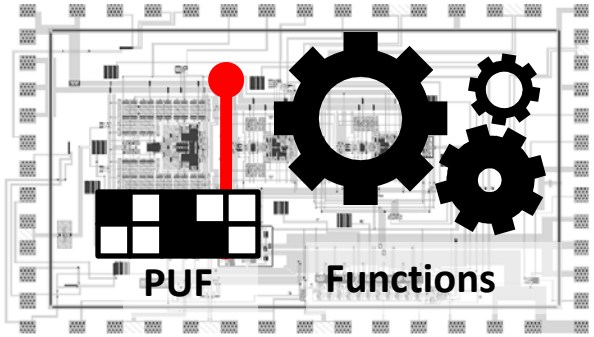
http://www.cs.clemson.edu/~mark/

Metering taxonomy*

- Passive metering
  - Nonfunctional identification
  - Functional Identification
- Active metering
  - Nonfunctional identification
  - Functional Identification

(All metering methods maybe based on reproducible or unclonable identifiers)

*Classification by F. Koushanfar, Book Chapter in 'Intro to Hardware Security and Trust, Springer'11

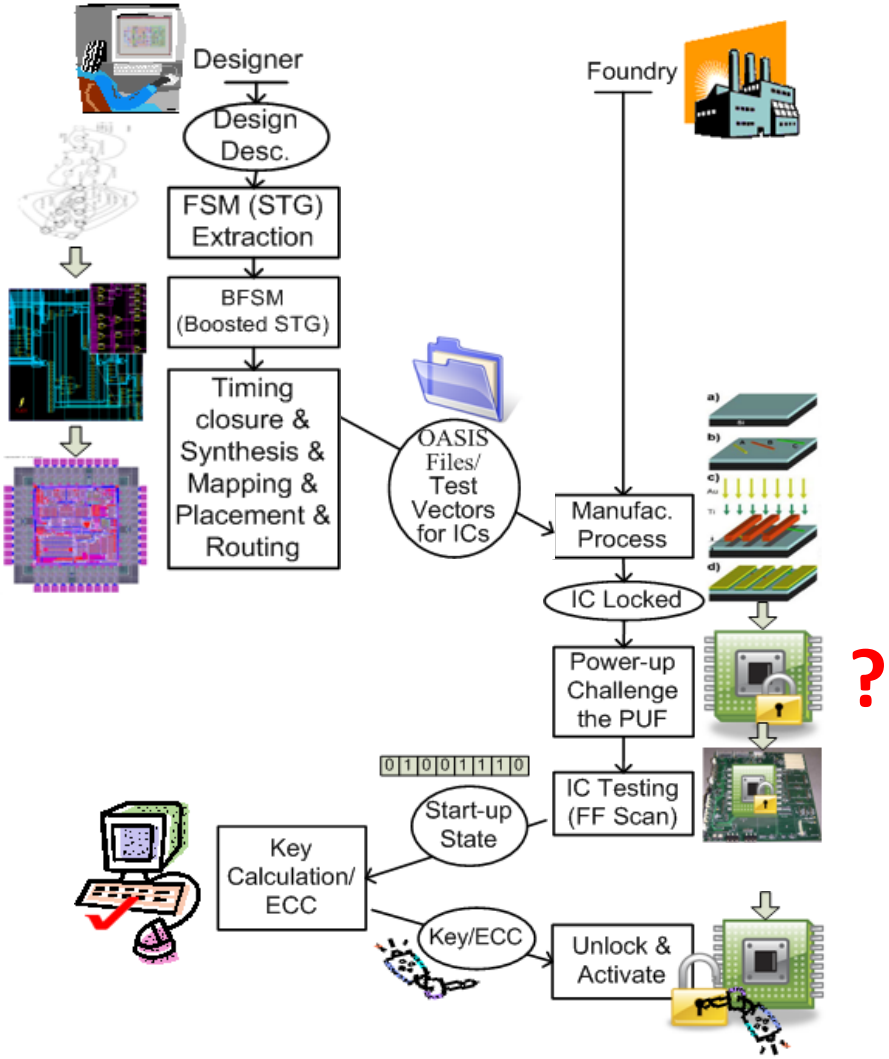


How to embed secure access entries (passwords) to the IC's functionality?

*Alkabani and Koushanfar. Active hardware metering. USENIX Security'07

** Koushanfar. Provably secure active metering for piracy protection. IEEE TIFS, to appear'11

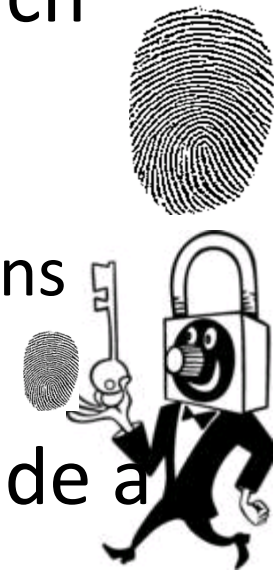
Active hardware metering flow*



*Alkabani, Koushanfar. USENIX Security'07

Active hardware metering: functional identification*

1. Process variation-based uniqueness of each chip
   - Lightweight secure identification and authentication by physical unclonable functions
   - Both weak and strong PUFs can be used
2. Alteration of the design structure to include a unique access control of each IC's functionality
   - Foundation, mechanism, synthesis method, evaluation, proof-of-concept implementation



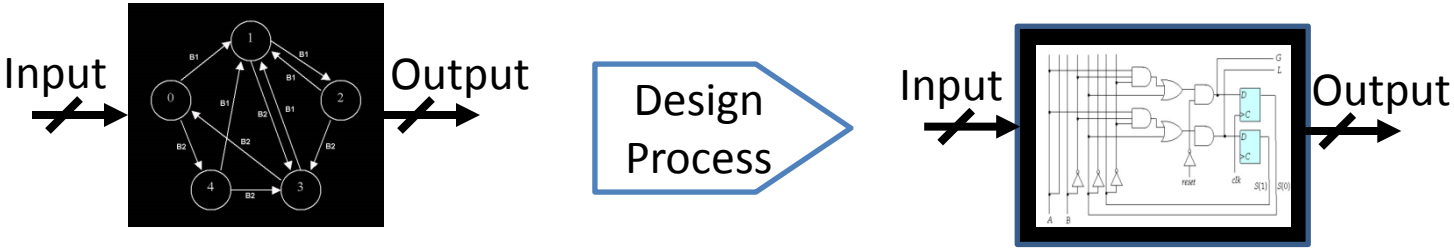
*Alkabani and Koushanfar. Active hardware metering. USENIX Security'07

Circuit functional description

- A finite state machine is a 6-tuple $(\Sigma, Q, \Delta, q_0, A, \Lambda)$:
  - An input alphabet Σ
  - A set of states Q
  - Δ is the state transition function: $Q \times \Sigma \rightarrow Q$
  - A start state q_0
  - A set of accepting states $A \subseteq Q$
  - Λ is output function (Mealy model) $Q \times \Sigma \rightarrow \Sigma$
  - Λ is output function (Moore Model) $\Sigma \rightarrow \Sigma$
- This functionality can be shown by a graph $G(Q,E)$, a.k.a state transition graph (STG)

Unique functional access

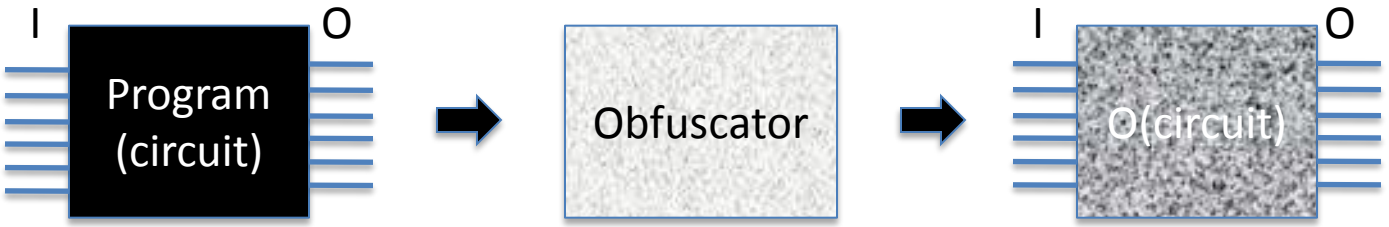
- Given a sequential circuit specification in the finite state machine format, can the designer embed access points (passwords) that are unique for each IC?



- Assumptions
  - The circuit netlist is publicly known
  - The access is activated by a sequence of inputs (password)
- But
  - Isn't this obfuscation?
  - Didn't Barak et al.* show the impossibility of obfuscation?

* Barak, Goldreich, Impagliazzo, Rudich, Sahai, Vadhan, Yang. Crypto'01

Quid: οβφυσχατιον?



- The general black-box obfuscator, where the O(circuit):
  - Has the same functionality ✗
  - A polynomial slow down
  - Is a virtual black-box
- Obfuscation in random oracle model, where the O(circuit):
  - Has approximately the same functionality
  - A polynomial slow down
  - Is a virtual black-box
- Provably secure obfuscation of a "point function" under the random oracle model*

* Lynn, Prabhakaran, Sahai. EuroCrypto'04

Obfuscation by point functions

- Consider the family of *point functions* $\{f_\alpha\}$

$$f_\alpha = \begin{cases} f_\alpha(x)=1, & \text{if } x=\alpha, \\ f_\alpha(x)=0 & \text{otherwise.} \end{cases}$$

- For a random Oracle *R* with a large enough range, the program storing the $R(\alpha)$ is an obfuscation of f_α with a high probability
- Example: password system
  - Weak password vs. strong password

Functions with general output and multi-point

- A point function with a general output $F_{\alpha,\beta}=\{0,1\}^k \rightarrow \{0,1\}^{s(k)}$ $F_{\alpha,\beta}(x)=\beta$ if $x=\alpha$ and $\perp$ otherwise
  - In the random oracle model, the point function with general output can be obfuscated
- A multi-point function with general output on $\{0,1\}^k \rightarrow (\{0,1\}^{s(k)})^t$ $F_{(\alpha 1,\beta 1),\ldots,(\alpha t,\beta t)}(x)=b_i$, where $b_i=\beta_i$ if $x=\alpha_i$, and else $b_i=\perp$
  - This is also efficiently obfuscatable in the random oracle model, in a self-composable manner

A STG graph-based access control

- A directed STG graph $G(Q+Q', E+E')$
  - Each $q \in \{Q+Q'\}$ has at most d ordered neighbors $\mu_q^{(1)},\ldots,\mu_q^{(d)}$
  - $E+E'=\{(q,v,i): v=\mu_q^{(i)}$ for some $i \in [d]\}$ be the set of all edges
  - A set of passwords on the edges $\{\pi_e | e \in E'\}$
  - A set of nodes not accessible without the password $\{\sigma_q | q \in [Q]\}$

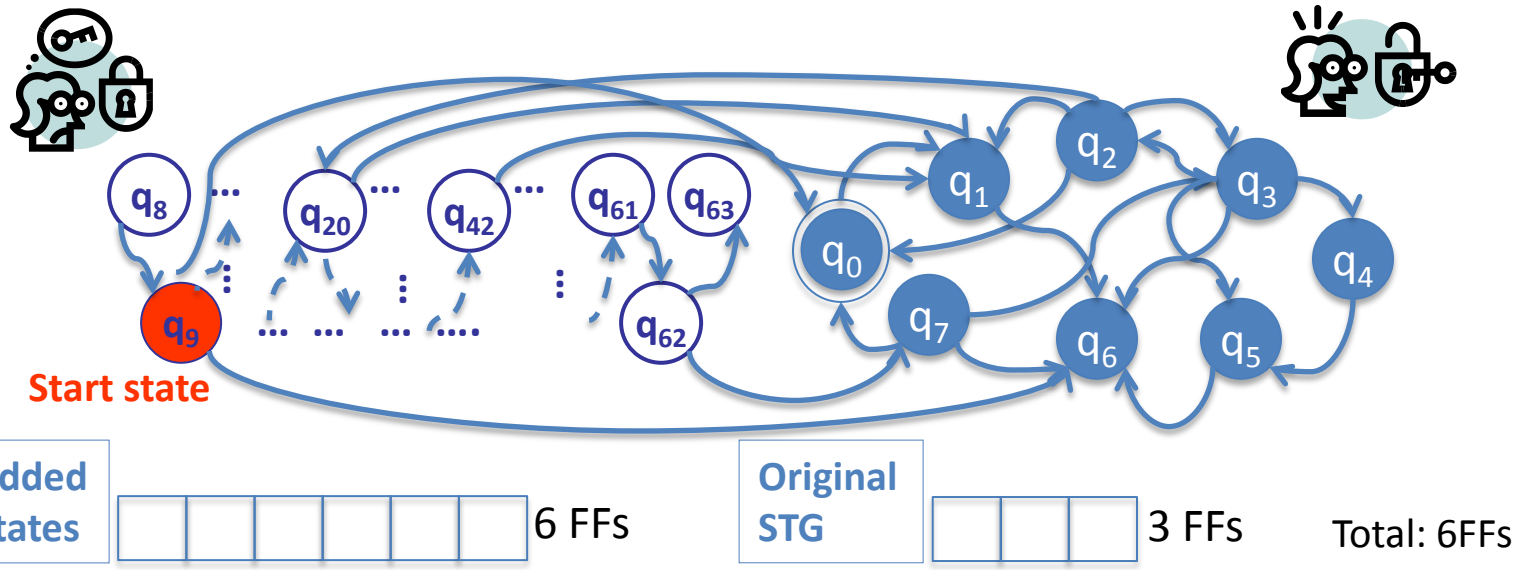
$$ACCESS_G((i_1,x_1),\ldots,(i_n,x_n))= \begin{cases} (v_n,\sigma_{vn}) & \text{if } \exists v_0,\ldots,v_{n-1} \in [Q'] \ \&\ v_n \in [Q] \ \&\ e_0,\ldots,e_{n-1} \in E' \\ & \text{s.t. } v_0=1, e_j=(v_j,vj+1,i_j) \text{ and } x_j=\pi_{ej} \\ \perp & \text{otherwise} \end{cases}$$

** Koushanfar. Provably secure active metering. IEEE TIFS, to appear'11

STG graph-based access control

- There may be exponentially many inputs to enable ACCESS_G
- Each node is represented by a tuple $(v, \sigma_v, e_1, .., e_d, \pi_{e1}, \ldots, \pi_{ed})$, where $e_i \in \{E'\}$
- $\forall q,\ 1 \le q \le |Q'|$ pick a random key κ_q from $\{0,1\}^l$; Let $\kappa_1 = 0^l$
- Define $W_\kappa^G(q,z,i,x) = \begin{cases} (v,\sigma_q,\kappa_q) & \text{if } z=\kappa_q \text{ and } \exists v \in Q' | \text{ s.t. } \pi_{q,v,i}=x \\ \perp & \text{otherwise} \end{cases}$
- The W_κ^G is a *multi-point function* with at most $|Q'|d$ points where the output is not $\perp$ and hence can be obfuscated
- The structure of the graph cannot be learned, except for the small paths from the start state to the reached access point

Access generation and embedding



- Create the passwords
- Sign it by a private key of a public key system
- Get a fixed width message by a one-way hash
- Use this value to create a sequence of inputs
- Modify the STG to include the added edges by adding also to the states

Ensuring a proper operation

- Powering-up in one of the added states Q'>>Q
  - The probability of powering-up in an added state is $(2^{|Q'+Q|}-2^{|Q|})/2^{|Q'+Q|}$
- Diversity of power-up states (unique IDs)
  - The probability $P_{ICID}(|Q'|,m)$ that no two ICs out of a group of m will have matching IDs out of $2^{|Q'|}$ possible options

$$P_{ICID}(|Q'|,m)=\frac{2^{|Q'|}-1}{2^{|Q'|}}\cdot\frac{2^{|Q'|}-2}{2^{|Q'|}}\cdots\frac{2^{|Q'|}-(m-1)}{2^{|Q'|}}=\frac{2^{|Q'|}!}{(2^{|Q'|}-m)!2^{m|Q'|}}$$

- Low overhead of the added states
- Diversity of keys
- Storing the input sequence and error correcting codes for traversal to the original reset state

Attacks on active hardware metering

1. Brute-force attack
2. Brute-force attack with memorization
3. Reverse engineering of FSM
4. PUF attacks
5. Initial power-up state capturing and replaying (CAR)
6. Initial reset state CAR
7. Control signals CAR
8. Creation of identical ICs using selective IC release
9. Combinational redundancy removal

Secure communications from/to the chip*,**:

*Roy, Koushanfar, Markov. DAC'08

**Roy, Koushanfar, Markov. IEEE Computer'10

Countermeasures

- Two important observations:
  - In modern designs, FSM is <<1% of the overall
  - The multi point access control is obfuscated

1. Brute-force attack
2. Brute-force attack with memorization
3. Reverse engineering of FSM

Countermeasures:

- Selection of strong passwords
- Provably efficient obfuscation of access points to FSM

Countermeasures (Cont'd)

4. PUF removal attack
5. Initial power-up state capturing and replaying (CAR)
6. Initial reset state CAR
7. Control signals CAR

Countermeasures:

- Interleave PUF within the circuit, so it's removal would require redesign and retiming
- Actively check the existence of PUF in the circuit*

*Alkabani, Koushanfar, Potkonjak. ICCAD'07

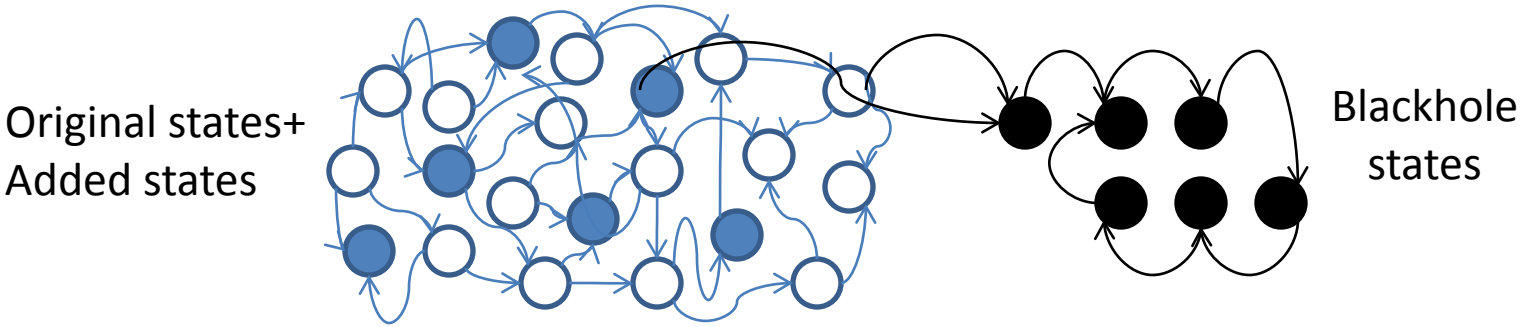
Countermeasures (Cont'd)

8. Creation of identical ICs by selective release
9. Combinational redundancy removal (CRR)

Countermeasures

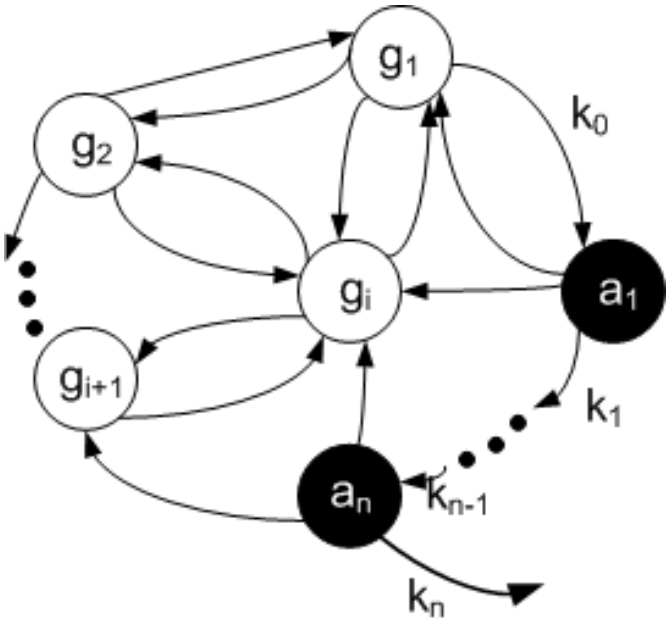
- Decrease the collision probability by design
- Selective release is economically not viable
- CRR is only possible for small netlists

- The access points enable remote control, disabling, enabling, and 3rd party IP protection*



*Alkabani, Koushanfar. CASES'08

Generation of the access graph



(1) Generate a key string $k_0k_1…k_n$

(2) Generate a graph of m nodes, $m>>n$

(1) n nodes are chosen to be access nodes $a_1,…,a_n$

(2) The key is used to transition from one access node to the other

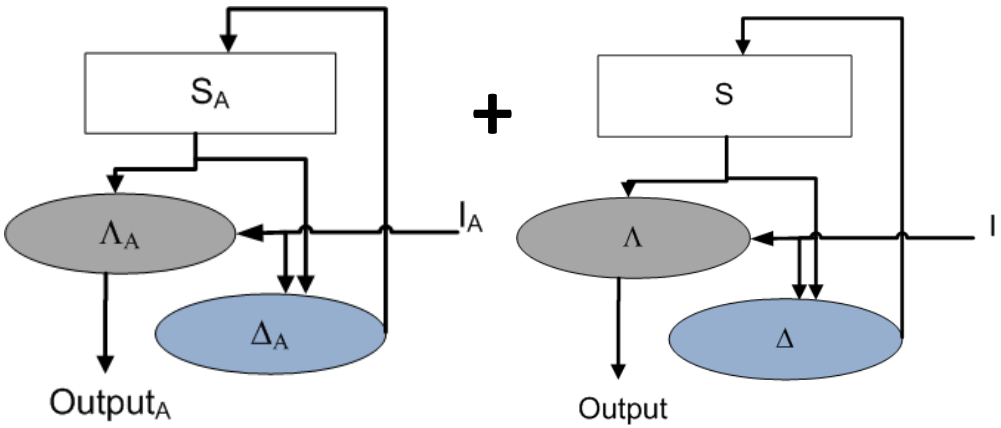
(3) The last access node has an edge to the starting state of the design

(4) Each node has exactly d edges coming out of it

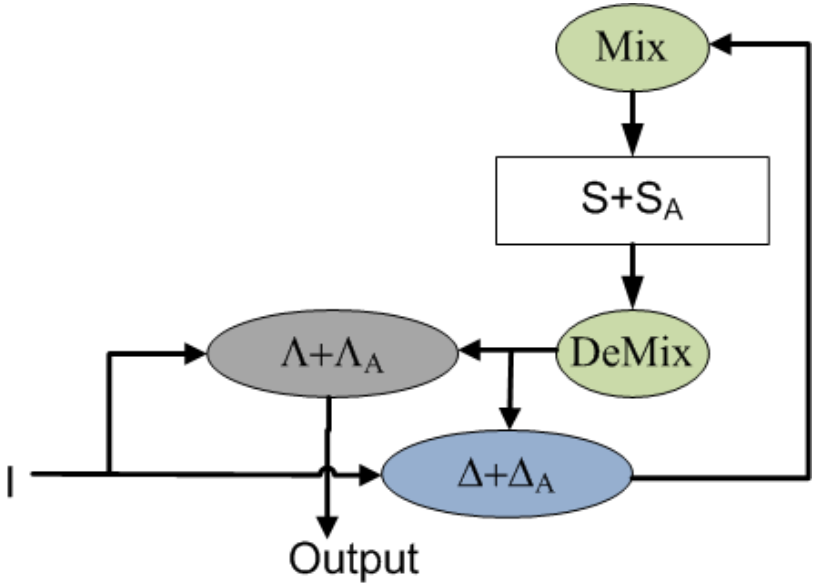
(5) The key is used to move from one access state to another

(3) Merge the access graph with the original STG

Merging the two state transition graphs



=



(1) The combinational logic in both STGs are combined

(2) Mix and DeMix circuits that share the bits of both STGs among the state elements are constructed

(3) Synthesis is run again to optimize the combinational circuits

Evaluation results – area overhead

- ISCAS benchmark, ABC Synthesis tool, Q'=64bits

circuit	PI	PO	reg	area	area	%
s344	9	11	15	269	1083	302.6
s349	9	11	15	273	1090	299.3
s641	35	23	19	539	1346	149.7
s713	35	23	19	591	1396	136.2
s820	18	19	5	757	1563	106.5
s832	18	19	5	769	1574	104.7
s1196	14	14	18	1009	1835	81.9
s1238	14	14	18	1041	1867	79.3
s1423	17	5	74	1164	1985	70.5
s1488	8	19	6	1387	2199	58.5
s1494	8	19	6	1393	2223	59.6
s5378	35	49	164	4212	5018	19.1
s9234	36	39	211	7971	8777	10.1
s13207	31	121	669	11241	12053	7.2
s15850	14	87	597	13659	14470	5.9
s35932	35	320	1728	28269	29078	2.9
s38584	12	278	1452	32910	33718	2.5

- Average area overhead=88%
- Recall that FSM is <<1%

Evaluation – power/delay overhead

- ISCAS Benchmark, ABC Synthesis tool, Q'=64bits

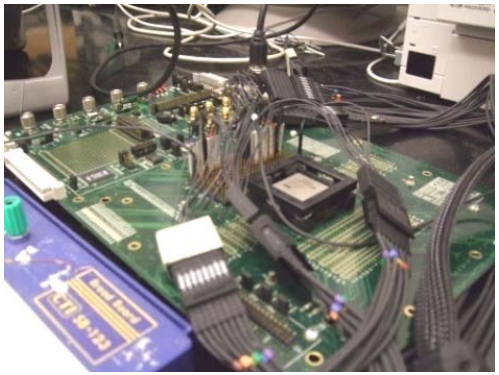
circuit	delay	power	delay	%	power	%
s344	27	1030.2	27	0.0	6012.7	483.6
s349	27	1039.1	27	0.0	6028.1	480.1
s641	97.6	1560.6	97.6	0.0	6518.8	317.7
s713	100	1670.7	100	0.0	6621.8	296.3
s820	28.2	2773.3	28.2	0.0	7729.2	178.7
s832	28.8	2849.6	28.8	0.0	7800.8	173.8
s1196	35.8	2557.6	35.8	0.0	7569	195.9
s1238	34.4	2709.4	34.4	0.0	7720.8	185
s1423	92.4	4882.7	92.4	0.0	9913.1	103
s1488	38	3859.3	38	0.0	8838	129
s1494	38.4	3913.4	38.4	0.0	8973.5	129.3
s5378	32.2	12459.4	32.2	0.0	17413.9	39.8
s9234	75.8	19385.5	75.8	0.0	24340.1	25.6
s13207	85.6	37843.6	85.6	0.0	42825.2	13.2
s15850	116	40002.7	116	0.0	44976.4	12.4
s35932	299.4	122048.4	299.4	0.0	127018.9	4.1
s38584	94.2	112706.8	94.2	0.0	117669.2	4.4

- Average power overhead=163%
- Recall that FSM is <<1%

Proof-of-concept:

FPGA implementation

- Benchmark H.264
- Xilinx Virtex 5, Xilinx ISE



	LUT	gates
h.264	26116	388321

#access	LUT	gates	total states	input key	%LUT	%gates
12	1004	7188	2^{20}	192	3.84	1.85
10	984	7048	2^{20}	160	3.77	1.81
8	944	6768	2^{20}	128	3.61	1.74
6	928	6656	2^{20}	96	3.55	1.71

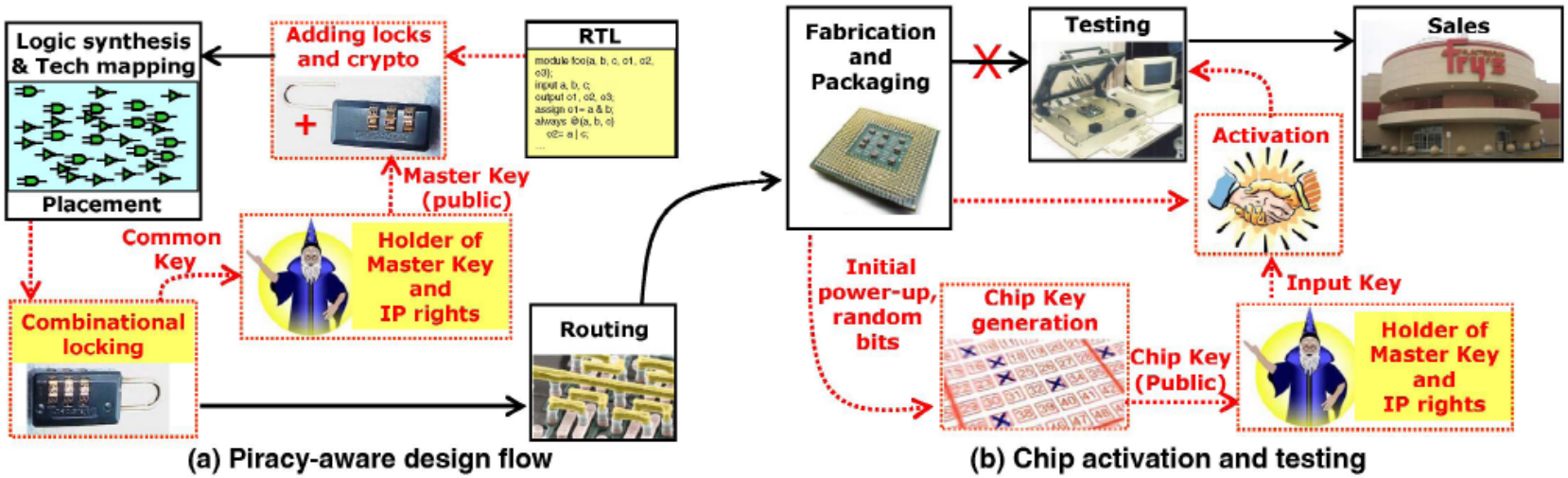
** Koushanfar. Provably secure active metering. IEEE TIFS, to appear'11

Active hardware metering

How else can it be done?

Active metering based on traditional cryptography a.k.a EPIC

- Asymmetric cryptography interfaced with the chip functionality

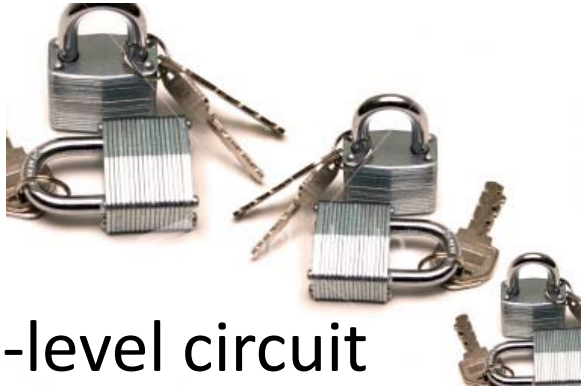


* Roy, Koushanfar, Markov. Protecting bus-based hardware IP by secret sharing, DAC'08

** Roy, Koushanfar, Markov. EPIC: Ending Piracy of ICs, DATE'08, IEEE Computer'11

EPIC: Ending Piracy of ICs

- Additional hardware
  - A novel lightweight **locking system**
  - **Public-key crypto with random key generation** (available on Niagara2)
  - Additional pins for encrypted keys



- Keys
  - **Common key** (CK) – built into gate-level circuit
  - **Master keys** (MK) – owned by holder of IP rights: private key never transmitted, cannot be deduced
  - **Random chip keys** (RCK) – public/private keys
  - **Input key** (IK) – key entered to unlock the chip

Spurious common keys ?

- Consider circuit C(x) and a locked variant C(x,y) such that for a designated key y_0
  $\forall x\ C(x, y_0)=C(x)$
- To find a working common key, must solve this Boolean equation
  $\exists\ y_0\ \forall x\ C(x, y_0)=C(x)$
  - Our locking construction guarantees solution
  - Note that this problem is beyond NP
- Can there be multiple solutions ? - Yes
  - Consider initial circuit $c=XOR(x_1,x_2)$
  - Locked variant $c=XOR(XOR(x_1,y_1),XOR(x_2,y_2))$
  - Common keys: (0,0) and (1,1)

Unique common keys

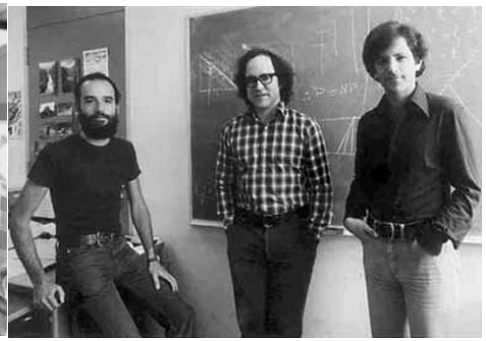
- Ideally we have $\exists!\ \mathbf{y_0}\ \forall \mathbf{x}\ \mathbf{C(x, y_0)=C(x)}$
- This can be checked for a given circuit
  - Build BDDs of C(x) and C(x, y)
  - Build BDD of the miter C(x,y)=C(x)
  - *Quantify out* ($\forall$) the variable x
  - *Count paths* in the resulting BDD (linear time)
  - Expected result: a single path
- To ensure unique common keys
  - Each wire should affect an output not affected by other wires ($\Rightarrow$ no cancellations)

EPIC: vulnerability assessment

- **Main scenarios:**
  - **Fab selling excess chips**
  - **Forgers stealing masks & using fabs**
- Additional cases, when forgers can
  - Reverse-engineer and modify masks
  - Modify chips in large quantities (FIB required)
  - Observe individual transient signals on chip
- Also must consider
  - Stolen RTL, gate-level netlist
  - Stolen layouts (placed & routed)
  - Stolen test vectors & correct responses

* Roy, Koushanfar, Markov. Protecting bus-based hardware IP by secret sharing, DAC'08
** Roy, Koushanfar, Markov. EPIC: Ending Piracy of ICs, DATE'08, IEEE Computer'11

Technology co



- **Operational assumptions**
  - *Public-key crypto cannot be broken or reversed*
  - RCK is random (available in Sun's Niagara 2)
  - RCK is generated once per chip (burned into fuses)
  - Common Key is unique (or has very few variants)
    - By construction + empirically checked
- **Multiple levels of protection**
  - Some keys are never transmitted (e.g., MK-Private)
  - Some keys are not in RTL (CK), or layout (RCK)
  - To break EPIC, must have both Master Keys (MK), Common Key (CK) and RCK-Public for each chip

[*] Roy, Koushanfar, Markov. Protecting bus-based hardware IP by secret sharing, DAC'08
[**] Roy, Koushanfar, Markov. EPIC: Ending Piracy of ICs, DATE'08, IEEE Computer'11

Conclusions

- **Hardware piracy a growing threat**
  - Current efforts barely go beyond serial numbers
- Metering offers a suit of **robust mechanisms to protect against piracy of ICs**
  - Passive metering based on unique identification, can be even done on legacy ICs
  - Active metering based on automatic locking of the chip based on unclonable identification
  - Both internal design methods and cryptographic methods
- **Overhead, security and attacks analyzed**

Metering: future directions

- Combinations of the two employed security mechanisms, provide basis for many new security and DRM protocols
  - Variability-based unclonable uniqueness of each IC
  - Functionality preserving structural manipulation of functionality
- Need more security and attack analysis
- Third party IC and IP integration
- Potential for broad impact on IC industry and military use
  - E.g., new royalty enforcement system

Thank you!

- Questions?
- farinaz@rice.edu