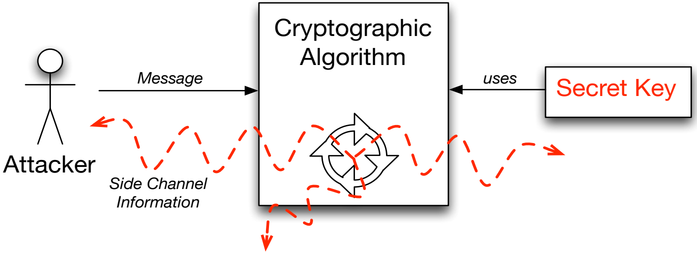
A Formal Model for Side-Channel Attacks

Boris Köpf
ETH Zurich

VERIMAG, Grenoble
12.11.2007

Side-Channel Attacks

- Attacks against implementations of cryptographic algorithms
- Physical characteristics of computations are exploited: timing, power, cache behavior, ...



- Increasingly effective
  - template attacks: key recovery from 1 power trace
  - remote timing attacks: 1024 bit RSA key in two hours
- Cryptographic security guarantees do not apply
- This talk: bounds on the information that can be extracted in an adaptive side-channel attack

Approaches to Countering Timing Attacks

- Ad-hoc countermeasures (randomization, blinding,...)
  - preferred in practice - render known attacks impossible
  - no formal security guarantees
  - more sophisticated attacks still possible?

Approaches to Countering Timing Attacks

- Ad-hoc countermeasures (randomization, blinding,...)
  - preferred in practice - render known attacks impossible
  - no formal security guarantees
  - more sophisticated attacks still possible?
- Formal approaches
  - physically observable cryptography (Micali & Reyzin ’03)
  - information-flow analysis
    - aims for proving implementations secure
    - based on formal system models and notions of security

Information-Flow Security and Timing Attacks

Limitations of today's approaches

- Abstract program models
  - timing behavior not adequately captured
- Security properties do not capture adaptive attackers
  - noninterference is very restrictive
  - quantitative properties only for passive observers

Information-Flow Security and Timing Attacks

Limitations of today's approaches

- Abstract program models
  - timing behavior not adequately captured
- Security properties do not capture adaptive attackers
  - noninterference is very restrictive
  - quantitative properties only for passive observers

This talk

- How to analyze security with respect to adaptive attackers
- Focus on special-purpose implementation in synchronous hardware

Problem Statement

Question

How much secret information can be extracted in an adaptive side-channel attack against a given implementation?

[1] joint work with David Basin

Problem Statement

Question

How much secret information can be extracted in an adaptive side-channel attack against a given implementation?

Contributions[1]

- A model that allows to express this quantity
- Algorithms and approximation techniques to compute it
- We apply our techniques to analyze implementations in synchronous hardware

[1] joint work with David Basin

Problem Statement

Question

How much secret information can be extracted in an adaptive side-channel attack against a given implementation?

Contributions[1]

- A model that allows to express this quantity
- Algorithms and approximation techniques to compute it
- We apply our techniques to analyze implementations in synchronous hardware

- Foundation for push-button tools for analyzing the vulnerability of systems to adaptive side-channel attacks

[1] joint work with David Basin

Outline

- Introduction
- Problem Statement
- Formalization of
  - side-channels
  - single attack steps
  - adaptive attacks
- Information-theoretic bounds
- Algorithms
- Experiments
- Conclusions
- Future work

A Simple Model of Side-Channels

$$f : \underbrace{K}_{\text{Keys}} \times \underbrace{M}_{\text{Messages}} \rightarrow \underbrace{O}_{\text{Observations}}$$

Assumptions

- Fixed unknown key
- Attacker knows f and can choose messages
- Observations are noiseless

Examples

- Number of clock ticks required for computation ($O = \mathbb{N}$)
- Bit toggles during each of n clock ticks ($O = \mathbb{N}^n$)

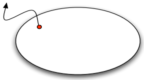
A Single Attack Step I

The attacker queries the system with $m \in M$, and

A Single Attack Step I

The attacker queries the system with $m \in M$, and

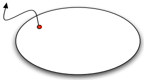
- observes $o = f(k, m)$



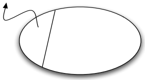
A Single Attack Step I

The attacker queries the system with $m \in M$, and

- observes $o = f(k, m)$



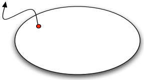
- narrows down the set of possible keys



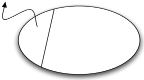
A Single Attack Step I

The attacker queries the system with $m \in M$, and

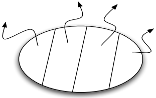
- observes $o = f(k, m)$



- narrows down the set of possible keys



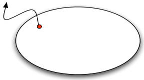
- Different observations correspond to disjoint subsets of K
  - k_1, k_2 are in the same subset iff $f(k_1, m) = f(k_2, m)$
  - every $m \in M$ induces a partition on K



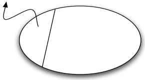
A Single Attack Step I

The attacker queries the system with $m \in M$, and

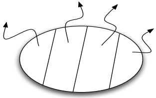
- observes $o = f(k, m)$



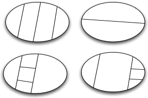
- narrows down the set of possible keys



- Different observations correspond to disjoint subsets of K
  - k_1, k_2 are in the same subset iff $f(k_1, m) = f(k_2, m)$
  - every $m \in M$ induces a partition on K

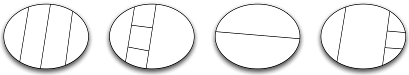


- f corresponds to a set of partitions $\{P_{m_1}, P_{m_2}, \ldots\}$ of K.



A Single Attack Step II

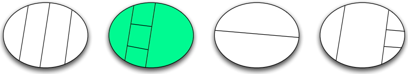
Abstraction of attack step:

- pick a partition P from 

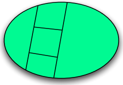
A Single Attack Step II

Abstraction of attack step:

- pick a partition P from



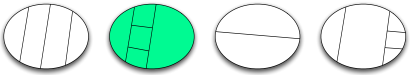
- obtain the block $B \in P$ that contains k



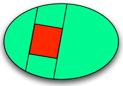
A Single Attack Step II

Abstraction of attack step:

- pick a partition P from

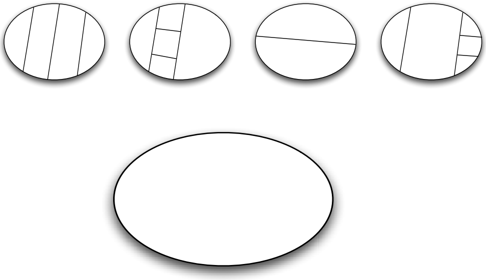


- obtain the block $B \in P$ that contains k



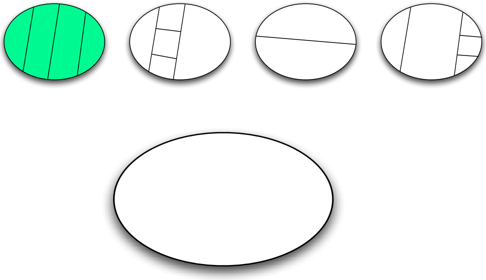
Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



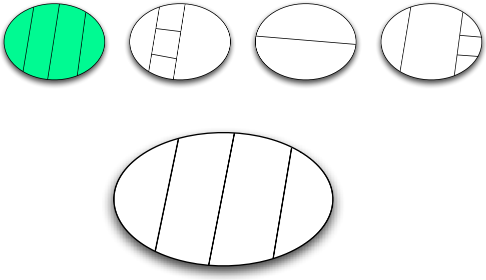
Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



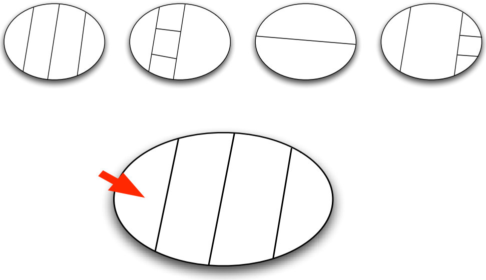
Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



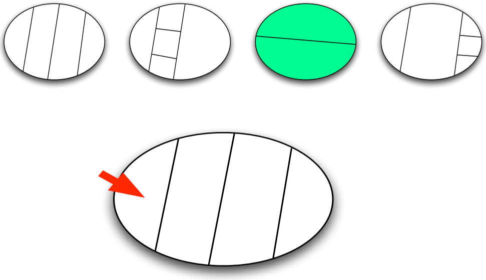
Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



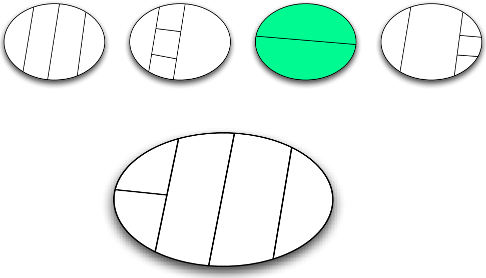
Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



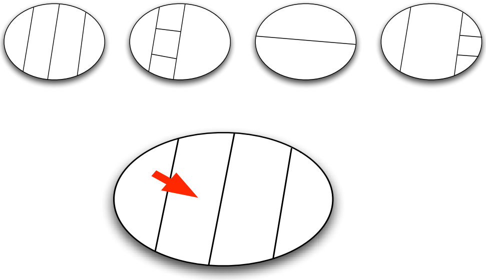
Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



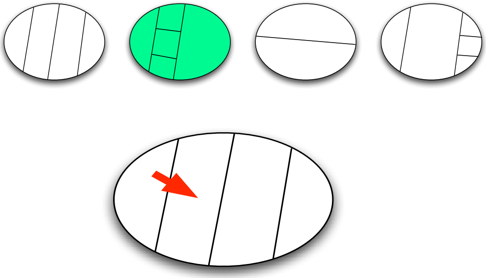
Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



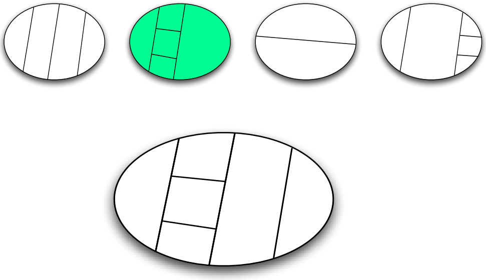
Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



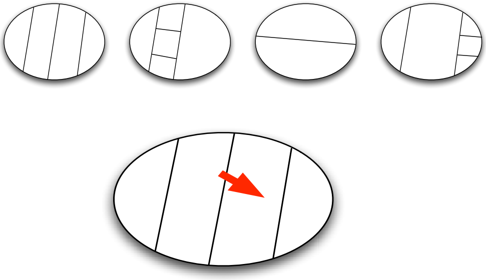
Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



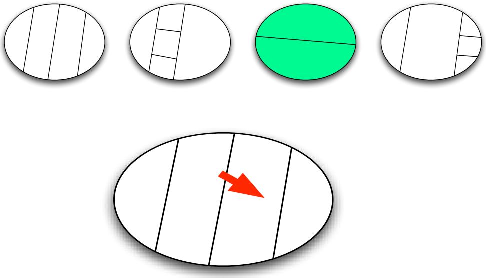
Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



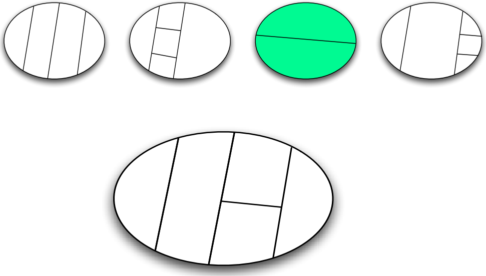
Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



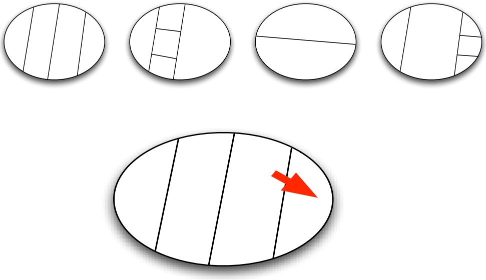
Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



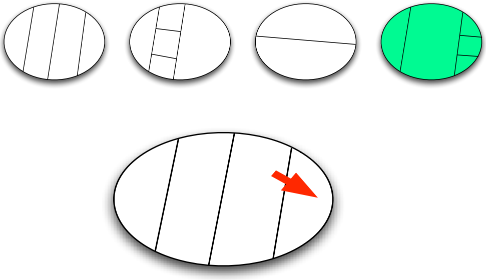
Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



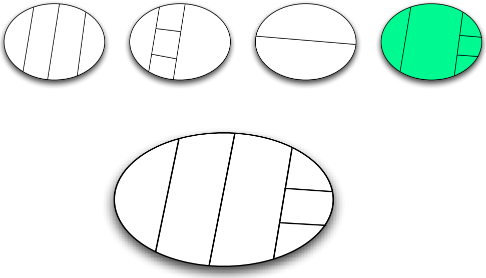
Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



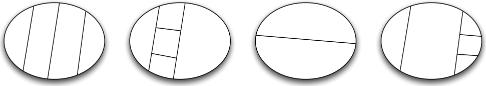
Adaptive Attacks (Intuition)

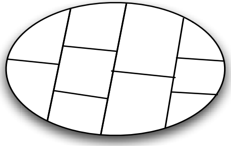
- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge



Adaptive Attacks (Intuition)

- Attacker's knowledge represented as set of possible keys
- In an adaptive attack, he can choose queries with respect to this knowledge





- Block-dependent choice of queries: attack strategy
- Attack strategies induce partitions on K

Adaptive Attacks (Formally)

Definition

An attack strategy α is a tree $T = (V, E)$ with vertex labeling $L : V \to 2^K$ with

1. $L(root) = K$, and
2. for every $v \in V$, there is a $m \in M$ such that $L(v) \cap P_m = \{L(w) \mid w \in succ(v)\}$

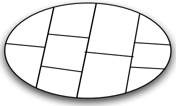
Adaptive Attacks (Formally)

Definition

An attack strategy α is a tree $T = (V, E)$ with vertex labeling $L : V \rightarrow 2^K$ with

1. $L(root) = K$, and
2. for every $v \in V$, there is a $m \in M$ such that $L(v) \cap P_m = \{L(w) \mid w \in succ(v)\}$

- Attack corresponds to a path in α
- Labels of the leaves induce partition P_α of K
- If the attacker follows strategy α, he learns the key's enclosing block in P_α



Quantitative Evaluation of Attack Strategies

Question

How much information can an attacker gain if he follows a given strategy?

Quantitative Evaluation of Attack Strategies

Question

How much information can an attacker gain if he follows a given strategy?

- Shannon entropy $H(X)$ of random variable X
  - Measure for the uncertainty about the outcome of X
  - Example: uniformly distributed 100-bit keys
    - no knowledge: $H(X) = 100$
    - known key: $H(X|X = k) = 0$
    - known Hamming weight: $H(X|hw(X)) = 95.6$
  - Alternative entropy measures can be used

Quantitative Evaluation of Attack Strategies

Question

How much information can an attacker gain if he follows a given strategy?

- Shannon entropy $H(X)$ of random variable X
  - Measure for the uncertainty about the outcome of X
  - Example: uniformly distributed 100-bit keys
    - no knowledge: $H(X) = 100$
    - known key: $H(X|X = k) = 0$
    - known Hamming weight: $H(X|hw(X)) = 95.6$
  - Alternative entropy measures can be used
- U: choice of a key from K (assume fixed distribution)
- V_P: choice of a block in a partition P of K
- $H(U|V_P) = E_{B \in P}(H(U|V_P = B))$: expected uncertainty about the key if the enclosing block is known

Quantitative Evaluation of Attack Strategies

Question

How much information can an attacker gain if he follows a given strategy?

- Shannon entropy $H(X)$ of random variable X
  - Measure for the uncertainty about the outcome of X
  - Example: uniformly distributed 100-bit keys
    - no knowledge: $H(X) = 100$
    - known key: $H(X|X = k) = 0$
    - known Hamming weight: $H(X|hw(X)) = 95.6$
  - Alternative entropy measures can be used
- U: choice of a key from K (assume fixed distribution)
- V_P: choice of a block in a partition P of K
- $H(U|V_{P_\alpha}) = E_{B \in P_\alpha}(H(U|V_{P_\alpha} = B))$: expected uncertainty about the key after an attack with strategy α

Resistance to Attacks

Desirable for evaluating implementations:

Bounds that hold against all attack strategies

Resistance to Attacks

Desirable for evaluating implementations:

Bounds that hold against all attack strategies

Resistance to attacks

$$\Phi(n) = \min\{H(U|V_{P_\alpha}) \mid \alpha \text{ attack strategy of length } n\}$$

Resistance to Attacks

Desirable for evaluating implementations:

Bounds that hold against all attack strategies

Resistance to attacks

$$\Phi(n) = \min\{H(U|V_{P_\alpha}) \mid \alpha \text{ attack strategy of length } n\}$$

- $\Phi(n)$: lower bound on the expected uncertainty about the secret after n steps of an adaptive attack against f
- Relates information gain and number of attack steps
- How do we compute Φ for a given implementation

Computing $\Phi(n)$

- Brute-force approach
  - treat f as a black box
  - enumerate all attack strategies
    - compute induced partitions
    - pick partition with minimal $H(U|V_P)$
  - requires time $\mathcal{O}(n \cdot |M|^{r^n} \cdot |K| \cdot \log |K|)$

Computing $\Phi(n)$

- Brute-force approach
  - treat f as a black box
  - enumerate all attack strategies
    - compute induced partitions
    - pick partition with minimal $H(U|V_P)$
  - requires time $\mathcal{O}(n \cdot |M|^{r^n} \cdot |K| \cdot \log |K|)$
- Approximation techniques
  - use a greedy strategy instead of enumerating all strategies
    - requires time $\mathcal{O}(n \cdot r \cdot |M| \cdot |K|^2)$
    - greedy is not optimal (in general)
  - compute ϕ for small bit-widths. Use regularity to extrapolate

Core Implementation in Haskell

```
greedy :: [Part k] -> Int -> [k] -> Part k
greedy f n keys = app n (greedystep f) [keys]

greedystep :: [Part k] -> Part k -> Part k
greedystep f pt = concat (map refine pt)
  where refine b = minimumBy entropy (restrict b f)
```

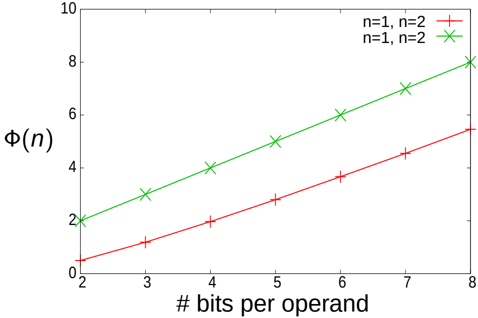
- *f* is given by the simulation environment of the HDL GEZEL
- Features of GEZEL:
  - specification of circuits as automata
  - cycle-true translation to VHDL

Experimental Results - Timing I

Timing analysis of shift-and-add integer multiplication

$$(\ldots((k_{w-1} \cdot m) \cdot 2 + k_{w-2} \cdot m) \cdot 2 + \ldots) \cdot 2 + k_0 \cdot m$$

- Two versions: unpadded and padded
- Operand k is the secret

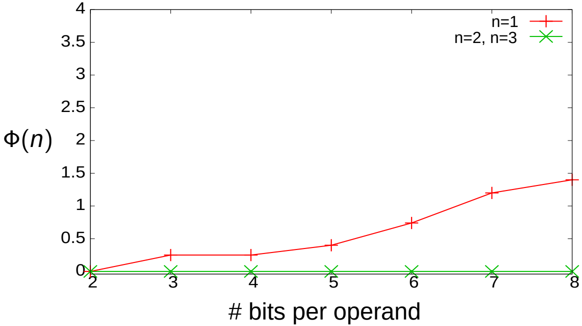


- Unpadded version leaks Hamming weight
- Padded version is secure

Experimental Results - Timing II

Timing behavior of finite field exponentiation

- Three nested loops
- Exponent is the secret

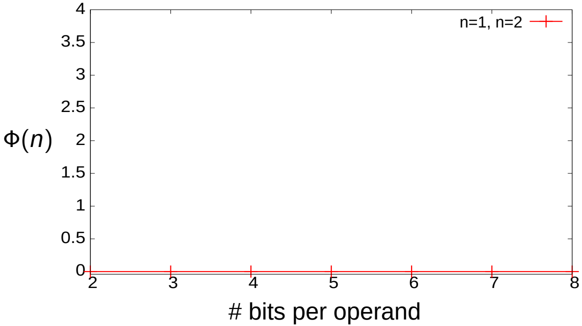


- After two attack steps the key is revealed
- Implementation is highly vulnerable to timing attacks

Experimental Results - Power

Power analysis of finite field multiplication

- Computes in constant time
- Bit-flips per clock cycle approximate power consumption



- After one attack step the key is revealed
- Template attacks show that this is possible even with noise

Conclusions

"How much secret information can be extracted in an adaptive side-channel attack?"

- Presented a simple model to express this quantity
- Showed how it can be computed
- Applied it to analyze implementations in synchronous hardware

Future Work

- Extend model to probabilistic systems
  - measurements with noise
  - evaluation of countermeasures
- Scaling-up
  - white-box analysis
  - entropy estimation – encouraging experimental results