A Formal Security Analysis of Even-Odd Sequential Prefetching in Profiled Cache-Timing Attacks

Sarani Bhattacharya, Chester Rebeiro, Debdeep Mukhopadhyay

Indian Institute of Technology Kharagpur

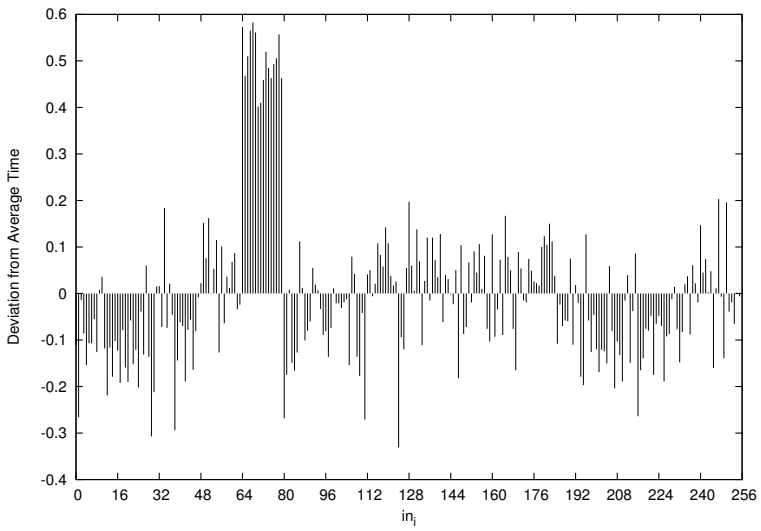
HASP 2016
June 18, 2016

Contributions of the paper

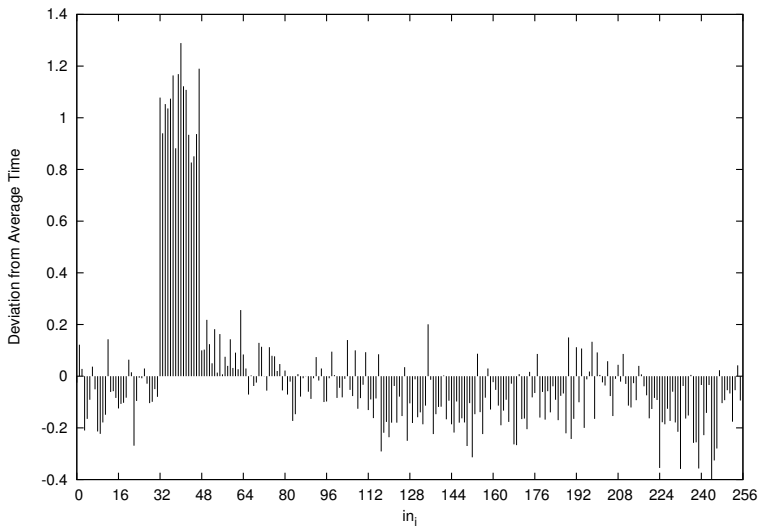
- In this paper we analyze the leakage for a variant of sequential hardware prefetching algorithm termed as *even-odd sequential* (EOS).
- We formally analyze this prefetching algorithm using combinatorial and provable techniques and a method is developed which quantifies the leakage in profiled cache timing attacks.
- We show that leakage due to the EOS prefetcher depends on the size and alignment of the tables used in the cipher.
- The results were verified with cachegrind [a].
- Further, we show that for a particular table alignment the leakage is always zero and for other alignments leakage reduces for large tables.

[a] http://valgrind.org/docs/manual/cg-manual.html

Profiled Cache Timing Attacks



(a) Known Key Profile



(b) Unknown Key Profile

Figure: Timing Profile for OpenSSL AES on Intel Core 2 Duo

Motivation for the work

- A formal treatment to quantify leakage for profiled cache-timing attacks was introduced in [1].
- Hardware cache prefetching [2], a common feature in most modern microprocessors, resulted in non-uniform encryption time and therefore a cause of leakage in profiled cache-timing attacks.
- In [3], block cipher CLEFIA and its vulnerability due to sequential prefetching is demonstrated using a metric Timing SVF in the context of profiled cache-timing attacks.

Even-Odd Sequential Prefetcher

Even-Odd Sequential Prefetcher is:

- a variant of sequential prefetcher.
- It prefetches the adjacent memory block whose location is determined by the address of the current access.
- If memory block is even then the next block is prefetched.
- If the memory block accessed is odd then the previous block is prefetched.

Algorithm for Even-Odd Sequential Prefetcher

- Input: Address of the memory block accessed (t_i)
- begin
- If *(t_i is not present in cache) or (t_i was prefetched and this is the first access to t_i)* then
  - If *t_i is even and t_{i+1} is not in cache* then *prefetch t_{i+1}*
  - If *t_i is odd and t_{i-1} not in cache* then *prefetch t_{i-1}*
- end

Mathematical Model for Cache Memory Accesses

- Let a cipher is implemented with a lookup table of l blocks.
- During execution table is accessed n_{max} number of times at random locations.

The steps involved in formal analysis are:-

- Obtain the probability of a cache hit in the n^{th} access to the lookup-table, where $1 \leq n \leq n_{max}$.
- Obtain the conditional probability of a cache hit in the n^{th} memory access to the lookup-tables, with the EOS prefetcher.
- The distribution of cache misses is Gaussian and therefore can be characterized by its mean and variance.
- Apply the Kullback-leibler divergence to quantify the information leakage.

Probability of a Cache hit in the n^{th} access in a classical cache

Let $A^C_{I,n}$ be a random variable that denotes the result of the n^{th} memory access to the table of size I in a system having a classical cache.

- $A^C_{I,n}$ can take values of either H or M respectively corresponding to a cache hit and a cache miss in the n^{th} memory access.

- The probability of obtaining a cache hit in the n^{th} access is given by [4],

$$\Pr[A^C_{I,n} = H] = \frac{1}{I^{n-1}} \sum_{i=0}^{n-2} \binom{n-1}{i} (I-1)^i \tag{1}$$

Probability of a Cache miss in the n^{th} access in a classical cache

The probability of obtaining a cache miss in the n^{th} memory access is

$$\Pr[A^C_{l,n} = M] = 1 - \Pr[A^C_{l,n} = H] \tag{2}$$

Probability of a Cache hit in the n^{th} access in a cache supporting prefetching

- P denote the given prefetching strategy
- $A_{l,n}^{C,P}$ the random variable denoting the result of the n^{th} access to the table of size l.
- The probability of obtaining a cache hit in the n^{th} access is

$$\Pr[A_{l,n}^{C,P} = H] = \Pr[A_{l,n}^{C,P} = H \mid collision] \cdot \Pr[collision] + \Pr[A_{l,n}^{C,P} = H \mid \overline{collision}] \cdot (1 - \Pr[collision])]$$

Two cases:-

- On collision, probability of a cache hit is exactly equal to cache hit in classical case.
- When no collision has occured, a cache hit is obtained if data has been prefetched.

The equation is rewritten as

$$\Pr[A_{l,n}^{C,P} = H] = \Pr[A_{l,n}^{C} = H] + \Pr[A_{l,n}^{P} = H] \cdot (1 - \Pr[A_{l,n}^{C} = H])]$$

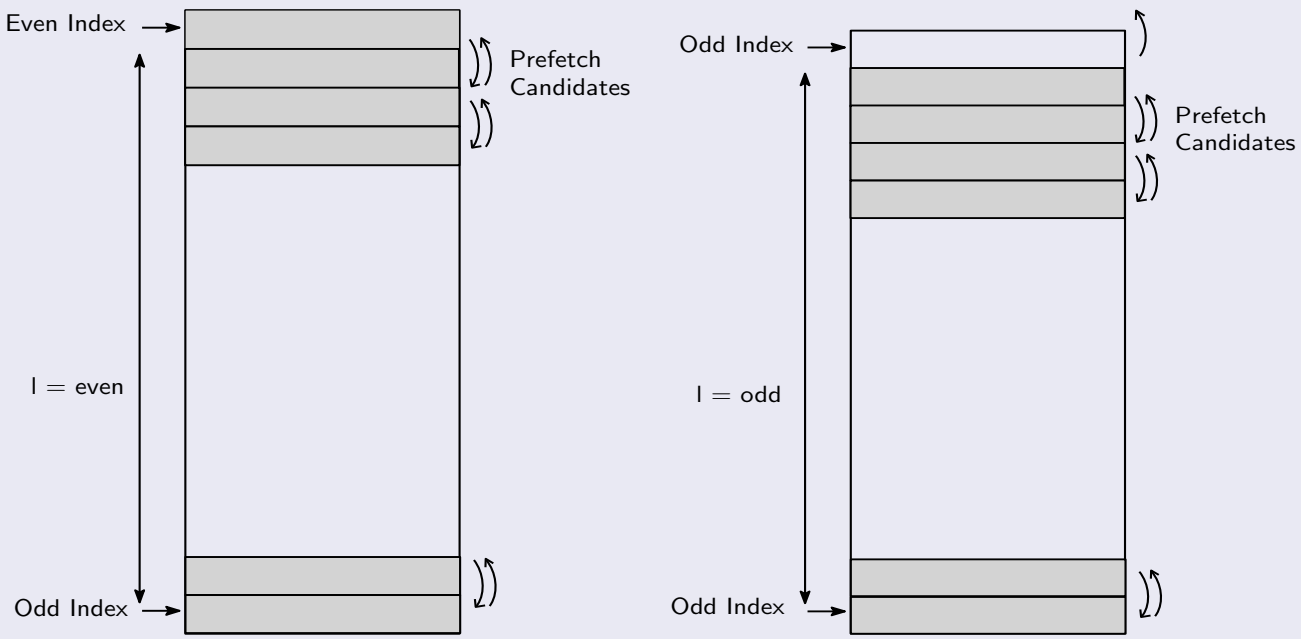
Conditional probability of the even-odd sequential prefetcher

Probability of cache hit is altered if one of previous accesses is known

Here we analyze the conditional probability of obataining a cache hit conditioning on the previous occurances of plain text.

- T_m is the random variable denoting the block in the table accessed in the m^{th} access.
- We assume $m = 1$, thus conditioning on the first access.
- To determine $\Pr[A_{l,n}^{EOSP} = H \mid T_m]$, where $A_{l,n}^{EOSP}$ is the random variable denoting the effect of the Even-Odd Sequential prefetcher in the n^{th} access.

Even-Odd prefetching in various table alignments

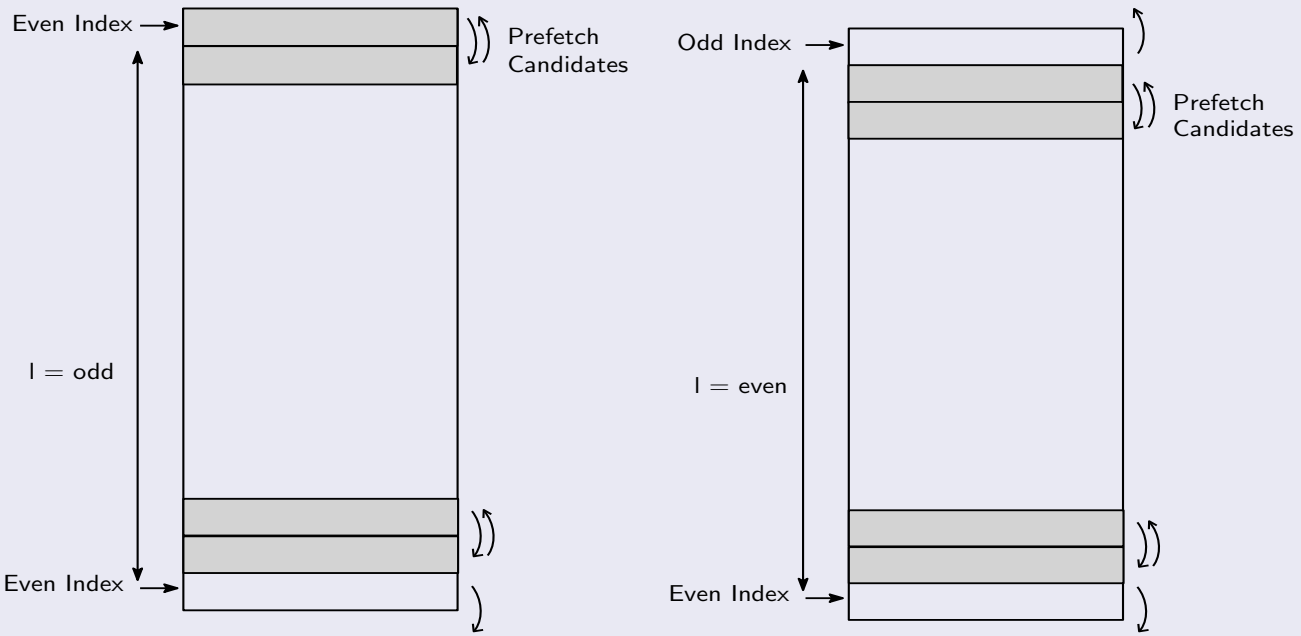


(a) EOSP in even-odd alignment

(b) EOSP in odd-odd alignment

Figure: Effect of EOSP on cache misses in various table alignments

Even-Odd prefetching in various table alignments



(a) EOSP in even-even alignment

(b) EOSP in odd-even alignment

Figure: Effect of EOSP on cache missses in various table alignments

Even-Odd Alignment

- The table starts from an even location, ends in a odd location
- Length of the table is even.

- *EOSP* be a function returning the prefetched memory block ie. $EOSP(t_b) = t_{b+1}$ if t_b is even and $EOSP(t_b) = t_{b-1}$ if t_b is odd.
- To determine the probability of hit in the n^{th} access given the first access. So two cases arises.
  - The probability that the n^{th} access is a hit due to the block being prefetched by the first access T_1
  - The block being prefetched by any other $n-2$ accesses other than the known first access T_1 and T_n

$$Pr[A_{l,n}^{EOSP} = \mathcal{H} | T_1] = Pr[A_{l,n}^{EOSP} = \mathcal{H} | T_n = EOSP(T_1)] \cdot Pr[T_n = EOSP(T_1)] + Pr[A_{l,n}^{EOSP} = \mathcal{H} | T_n \neq EOSP(T_1)] \cdot Pr[T_n \neq EOSP(T_1)]$$

There are two components in this equation

- *When* $T_n = EOSP(T_1)$, it would certainly cause a cache hit.
- Since T_n cannot have a collision with T_1, it can only take $l - 1$ different values and not l. Thus,

$$Pr[T_{l,n} = EOSP(T_1)] = \frac{1}{l-1}$$

$T_n \neq EOSP(T_1)$

- Happens with probability $1 - 1/(l-1)$.
- Hit in the n^{th} access occurs iff $T_n = EOSP(T_i)$ and $2 \leq i \leq n-1$.
- The probability of occurance is given by:

$$\Pr[A_{l,n}^{EOSP} = H \mid T_n \neq EOSP(T_1)] = \frac{\alpha}{(l-2)(l-1)^{n-2}} \cdot \sum_{i=1}^{n-2} \binom{n-2}{i} (l-2)^{n-2-i}$$

- where α is the number of prefetchable blocks.
  - Thus, $\alpha = l - 2$ as $T_n \neq T_1$ and $T_n \neq EOSP(T_1)$

Thus combining two parts overall equation is written as,

$$\Pr[A_{l,n}^{EOS} = H \mid T_1] =$$

$$\frac{1}{(l-1)^{n-1}}[(l-1)^{n-2} + (l-2) \cdot \sum_{i=1}^{n-2} \binom{n-2}{i} (l-2)^{n-2-i}]$$

Odd-Odd Table Alignment

With the same analysis as before we here have two components as

- *When* $T_n = EOSP(T_1)$, it would certainly cause a cache hit.
- T_n cannot have a collision with T_1
- It can only take $l - 1$ different values and not l. Thus,

$$Pr[T_{l,n} = EOSP(T_1)] = \frac{1}{l-1}$$

When $T_n \neq EOSP(T_1)$

- This happens with probability $1 - 1/(l-1)$.
- Hit in the n^{th} access occurs iff $T_n = EOSP(T_i)$ and $2 \leq i \leq n-1$.
- The probability with which this happens is given as:

$$\Pr[A_{l,n}^{EOSP} = H \mid T_n \neq EOSP(T_1)] = \frac{\alpha}{(l-2)(l-1)^{n-2}} \cdot \sum_{i=1}^{n-2} \binom{n-2}{i} (l-2)^{n-2-i} \quad (3)$$

,where α is the number of prefetchable blocks.

- If $T_1 = t_1$ then $T_n \neq t_1$. Thus, $\alpha = l - 1$.
- If $T_1 \neq t_1$ then $T_n \neq T_1$, $T_n \neq EOSP(T_1)$ and $T_n \neq t_1$. Thus, $\alpha = l - 3$.

Odd-Even Alignment

- Table start from an odd location.
- Has an even length and thus ends in an even location.
- First block and the last block cannot be prefetched.
- Both of them prefetches a block outside the table.
- Probability equation can be written as

$$\begin{aligned} &\Pr[A_{l,n}^{EOSP} = H \mid T_1 = t_l \text{ or } T_1 = t_1] \\ &= \frac{(l-2)}{(l-1)^{n-1}} \cdot \sum_{i=1}^{n-2} \binom{n-2}{i} (l-2)^{n-2-i} \end{aligned} \tag{4}$$

When $T_1 \neq t_l$ and $T_1 \neq t_1$

There are two components in this equation.

- *When* $T_n = EOSP(T_1)$, it would certainly cause a cache hit. Also, since T_n cannot have a collision with T_1, it can only take $l-1$ different values and not l. Thus,

$$Pr[T_{l,n} = EOSP(T_1)] = \frac{1}{l-1}$$

- *When* $T_n \neq EOSP(T_1)$. This happens with probability $1 - 1/(l-1)$. A hit in the n^{th} access occurs if and only if $T_n = EOSP(T_i)$ and $2 \leq i \leq n-1$. The probability with which this happens is given by the following Equation.

$$\Pr[A_{l,n}^{EOSP} = H \mid T_n \neq EOSP(T_1), T_1 \neq t_l, T_1 \neq t_1] = \frac{\alpha}{(l-2)(l-1)^{n-2}} \cdot \sum_{i=1}^{n-2} \binom{n-2}{i} (l-2)^{n-2-i} \quad (5)$$

α is the number of prefetchable blocks.

- $T_n \neq T_1$, $T_n \neq EOSP(T_1)$, $T_n \neq t_l$(cannot be prefetched) and $T_n \neq t_1$(cannot be prefetched). Thus, $\alpha = l - 4$.

Even-Even Alignment

- The table starts from an even location
- Ends in an even location.
- All the blocks in the table can be prefetched.
- Last block prefetches a block outside the table.
- Probability equation can be given as

$$\Pr[A_{l,n}^{EOSP} = H \mid T_1 = t_l] = \frac{1}{(l-1)^{n-2}} \cdot \sum_{i=1}^{n-2} \binom{n-2}{i} (l-2)^{n-2-i} \quad (6)$$

When $T_1 \neq t_l$

- *When* $T_n = EOSP(T_1)$, it would certainly cause a cache hit. Also, since T_n cannot have a collision with T_1, it can only take $l-1$ different values and not l. Thus,

$$Pr[T_{l,n} = EOSP(T_1)] = \frac{1}{l-1}$$

- *When* $T_n \neq EOSP(T_1)$. This happens with probability $1 - 1/(l-1)$. A hit in the n^{th} access occurs if and only if $T_n = EOSP(T_i)$ and $2 \leq i \leq n-1$. The probability with which this happens is given by the following equation.

$$\Pr[A_{l,n}^{EOSP} = H \mid T_n \neq EOSP(T_1), T_1 \neq t_l] = \frac{\alpha}{(l-2)(l-1)^{n-2}} \cdot \sum_{i=1}^{n-2} \binom{n-2}{i} (l-2)^{n-2-i} \quad (7)$$

α is the number of prefetchable blocks.

- since $T_n \neq T_1$, $T_n \neq EOSP(T_1)$ and $T_n \neq t_l$(cannot be prefetched) . Thus, $\alpha = l - 3$.

Cache Miss Distribution

This is a normal distribution having mean and variance

- The expected number of cache misses in the n^{th} access is given by ,

$$\begin{aligned} E(A_{l,n}) &= 0 \cdot (Pr[A_{l,n} = \mathcal{H}]) + 1 \cdot (Pr[A_{l,n} = \mathcal{M}]) \\ &= 1 - Pr[A_{l,n} = \mathcal{H}] \end{aligned}$$

- The variance of cache misses in the n^{th} access is

$$\begin{aligned} V(A_{l,n}) &= (1 - Pr[A_{l,n} = \mathcal{H}]) - (1 - Pr[A_{l,n} = \mathcal{H}])^2 \\ &= Pr[A_{l,n} = \mathcal{H}]^2 + Pr[A_{l,n} = \mathcal{H}] \end{aligned}$$

Expectation of cache Misses

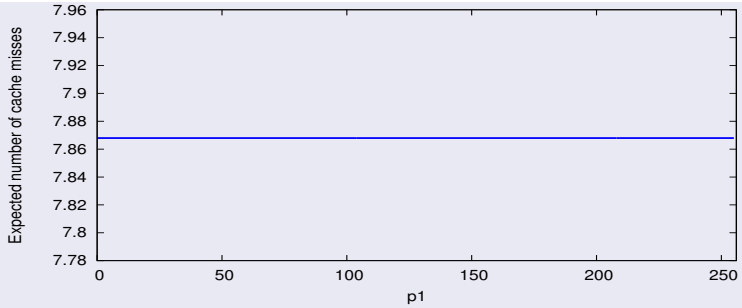
- The expectation of the number of cache misses after n memory accesses are given by the recurrence equation as,

$$E(M_n) = E(M_{n-1}) + E(A_{l,n})$$

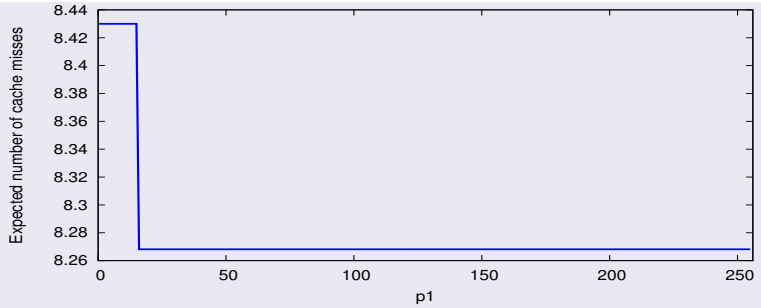
- Suppose,

$$E(M_1) = 1$$

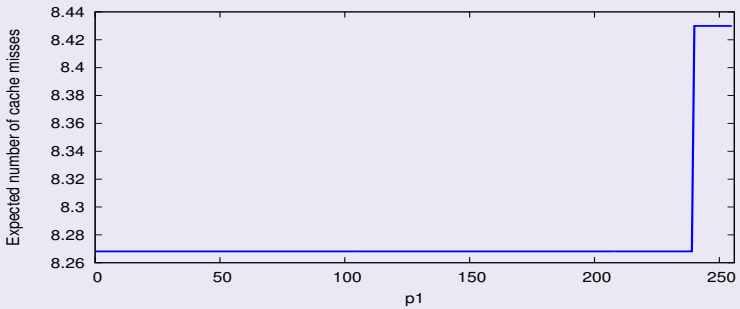
Cache profiles with Even-Odd Sequential Prefetcher for different Table Alignments



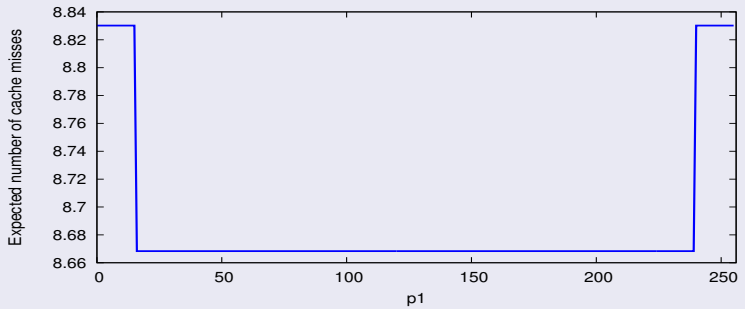
(a) EOSP with even-odd



(b) EOSP with odd-odd



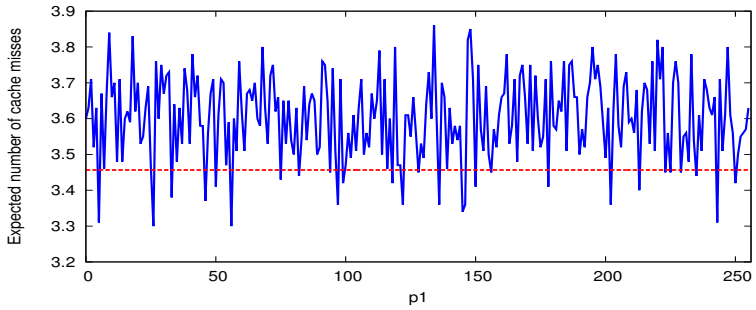
(c) EOSP with even-even



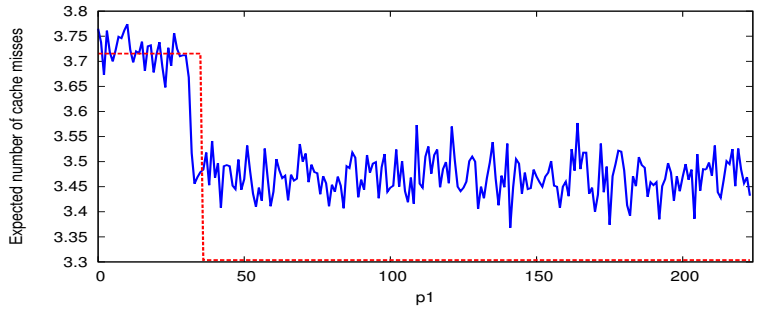
(d) EOSP with odd-even

Figure: Cache Profiles for k_1 with Even-Odd Sequential Prefetching Styles for Different Table Alignments and $l = 16$, $n_{max} = 36$, $\delta = 16$ with $x-$axis having the conditioned value and y-axis the number of cache misses

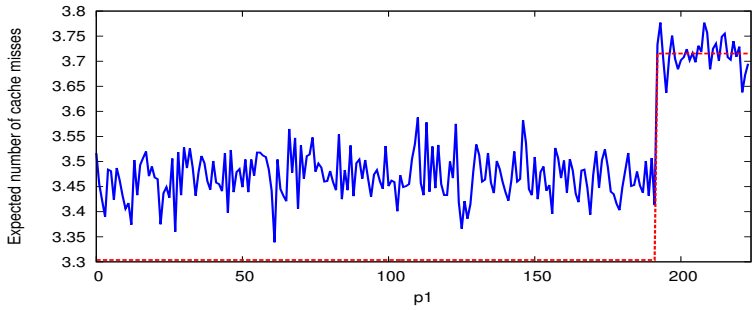
Predicted and Emperical Cache Profiles



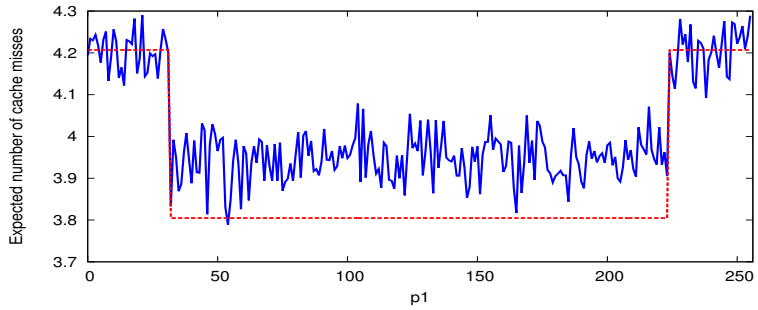
(a) Cache Misses with even-odd



(b) Cache Misses with odd-odd



(c) Cache Misses with even-even



(d) Cache Misses with odd-even

Figure: Predicted and Empirical Cache Profiles for k_1 for Cipher Model $\Gamma = 1$, $n_{max} = 8$

Information Leakage using Kullback Leibler Divergence

Kullback-Leibler Divergence

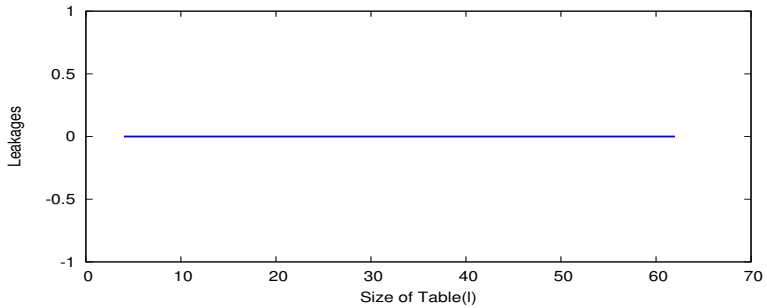
- To quantify the deviations in timing profiles Kullback-Leibler Divergence is used
- Symmetric KL divergence between two distribution $F_{k_{1,i}}$ and $F_{k_{1,i'}}$ is computed
- The metric is defined as:

$$D(F_{k_{1,i}}, F_{k_{1,i'}}) = D(F_{k_{1,i}} || F_{k_{1,i'}}) + D(F_{k_{1,i'}} || F_{k_{1,i}}) \tag{8}$$

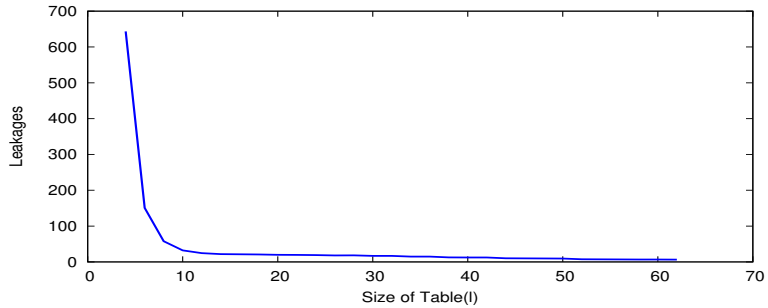
where

$$D(F_x || F_y) = \sum_j F_x(j) \log \frac{F_x(j)}{F_y(j)}$$

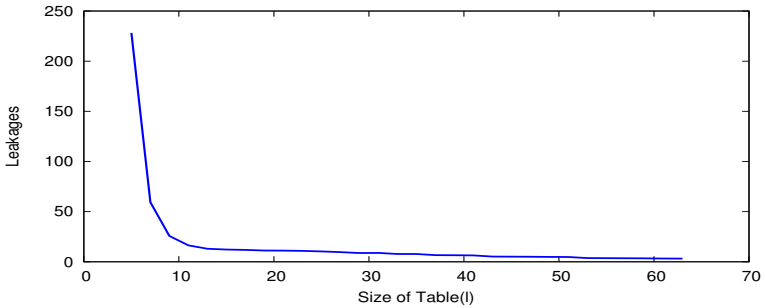
Variation of Leakage with Table Sizes



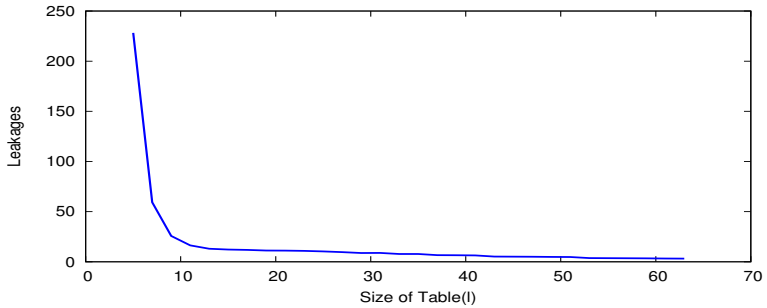
(a) Even-Odd Table alignment ($\Gamma = 1$)



(b) Odd-Even Table alignment ($\Gamma = 1$)



(c) Odd-Odd Table alignment ($\Gamma = 1$)



(d) Even-Even Table alignment ($\Gamma = 1$)

Figure: Leakage for k_1 with Various Table Alignments as the Table Size Increases ($n_{max} = 36$)

Conclusion

- Information leakage due to the Even-Odd Sequential prefetcher is formally analyzed for profiled cache-timing attacks.
- The analysis shows that the alignment of tables i.e., their starting and ending locations do have a great impact on the leakage of information.
- Also for a particular table alignment namely when the table start at an even memory block and ends at an odd memory block, there is no information leaked.

Thank You


Chester Rebeiro and Debdeep Mukhopadhyay.
A formal analysis of prefetching in profiled cache-timing attacks on block ciphers.
IACR Cryptology ePrint Archive, 2015:1191, 2015.


John L. Hennessy and David A. Patterson.
Computer Architecture: A Quantitative Approach, 4th Edition.
Morgan Kaufmann, 2006.


Sarani Bhattacharya, Chester Rebeiro, and Debdeep Mukhopadhyay.
Hardware prefetchers leak: A revisit of SVF for cache-timing attacks.
In *45th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 2012, Workshops Proceedings, Vancouver, BC, Canada, December 1-5, 2012*, pages 17–23. IEEE Computer Society, 2012.


A Formal Analysis of Prefetching in Profiled Cache-Timing Attacks.
Technical report, Communicated to the Journal of Cryptology.