Formal Fault Analysis of Branch Predictors: Attacking countermeasures of Asymmetric key ciphers

Sarani Bhattacharya and Debdeep Mukhopadhyay

Indian Institute of Technology Kharagpur

PROOFS 2016
August 20, 2016

Overview of the talk

- Introduction
- Motivation of the problem
- Exponentiation primitives for Public key cryptography
- Formalizing Differential of branch misses simulated from 2-bit predictor
- Developing the Attack Algorithm
- Experimental validation over Hardware Performance Counters
- Conclusion

Introduction

- Asymmetric key algorithm have been threatened via timing side channels due to the behavior of the underlying branch predictors.

Introduction

- Asymmetric key algorithm have been threatened via timing side channels due to the behavior of the underlying branch predictors.
- Effect of faults on such predictors and the consequences thereof on security of crypto-algorithms have not been studied.

Introduction

- Asymmetric key algorithm have been threatened via timing side channels due to the behavior of the underlying branch predictors.
- Effect of faults on such predictors and the consequences thereof on security of crypto-algorithms have not been studied.
- We develop a formal analysis of such a bimodal predictor under the effect of faults.

Introduction

- Asymmetric key algorithm have been threatened via timing side channels due to the behavior of the underlying branch predictors.
- Effect of faults on such predictors and the consequences thereof on security of crypto-algorithms have not been studied.
- We develop a formal analysis of such a bimodal predictor under the effect of faults.
- Analysis shows that differences of branch misses under the effect of bit faults can be exploited to attack implementations of RSA-like asymmetric key algorithms, based on square and multiplication operations.

Introduction

- Asymmetric key algorithm have been threatened via timing side channels due to the behavior of the underlying branch predictors.
- Effect of faults on such predictors and the consequences thereof on security of crypto-algorithms have not been studied.
- We develop a formal analysis of such a bimodal predictor under the effect of faults.
- Analysis shows that differences of branch misses under the effect of bit faults can be exploited to attack implementations of RSA-like asymmetric key algorithms, based on square and multiplication operations.
- The attack is also threatening against Montgomery ladder of CRT-RSA (RSA implemented using Chinese Remainder Theorem).

Contributions

- The difference of branch misses observed through HPCs between the correct and the faulty execution can be modeled efficiently to develop a key recovery attack.

Contributions

- The difference of branch misses observed through HPCs between the correct and the faulty execution can be modeled efficiently to develop a key recovery attack.
- We develop an iterative attack strategy, which simulates the branches corresponding to partially known exponent bits and observes the difference of branch misses from HPCs to reveal the next bit.

Contributions

- The difference of branch misses observed through HPCs between the correct and the faulty execution can be modeled efficiently to develop a key recovery attack.
- We develop an iterative attack strategy, which simulates the branches corresponding to partially known exponent bits and observes the difference of branch misses from HPCs to reveal the next bit.
- The theoretical simulations are validated on secret key-dependent modular exponentiation algorithms as well as on CRT-RSA implementation.

Vulnerability of system due to HPCs in presence of fault

- The scenario where the secret key gets flipped or corrupted can manifest as a fault.

Vulnerability of system due to HPCs in presence of fault

- The scenario where the secret key gets flipped or corrupted can manifest as a fault.
- However, fault can also be introduced by skipping some target instructions as well [1].

Vulnerability of system due to HPCs in presence of fault

- The scenario where the secret key gets flipped or corrupted can manifest as a fault.
- However, fault can also be introduced by skipping some target instructions as well [1].
- On platforms such as Xilinx Microblaze where the HPC accesses are provided [2], the instruction skip phenomenon can be exploited to reveal secret by monitoring events such as branching.

Vulnerability of system due to HPCs in presence of fault

- The scenario where the secret key gets flipped or corrupted can manifest as a fault.
- However, fault can also be introduced by skipping some target instructions as well [1].
- On platforms such as Xilinx Microblaze where the HPC accesses are provided [2], the instruction skip phenomenon can be exploited to reveal secret by monitoring events such as branching.
- In recent processors, Rowhammer is a term coined for disturbances observed in DRAM devices, where repeated row activation causes the DRAM cells to electrically interact within themselves [3, 4].

Vulnerability of system due to HPCs in presence of fault

- The scenario where the secret key gets flipped or corrupted can manifest as a fault.
- However, fault can also be introduced by skipping some target instructions as well [1].
- On platforms such as Xilinx Microblaze where the HPC accesses are provided [2], the instruction skip phenomenon can be exploited to reveal secret by monitoring events such as branching.
- In recent processors, Rowhammer is a term coined for disturbances observed in DRAM devices, where repeated row activation causes the DRAM cells to electrically interact within themselves [3, 4].
- Authors in [5] has exploited this Rowhammer vulnerability to flip secret exponent bits residing in the memory of a x86 system. This motivates the study of differential analysis of HPCs when there is a fault.

- In fault analysis attacks as well as their countermeasures, the adversary may be prevented in getting useful information but the hardware events reflects the systems internal state which may have a dependence on the secret.

- In fault analysis attacks as well as their countermeasures, the adversary may be prevented in getting useful information but the hardware events reflects the systems internal state which may have a dependence on the secret.
- HPCs can be of potential threat with respect to fault analysis attacks and more notably against their countermeasures.

Exponentiation and Underlying Multiplication Primitive

- Inputs(M) are encrypted and decrypted by performing modular exponentiation with modulus N on public or private keys represented as n bit binary string.

Square and Multiply Exponentiation

Algorithm 1: **Binary version of Square and Multiply Exponentiation Algorithm**

$S \leftarrow M$;
for i **from** 1 **to** $n-1$ **do**
    $S \leftarrow S * S \ mod \ N$;
    if $d_i = 1$ **then**
        $S \leftarrow S * M \ mod \ N$;
    end
end
return S ;

- Conditional execution of instruction and their dependence on secret exponent is exploited by the simple power and timing side-channels [6].

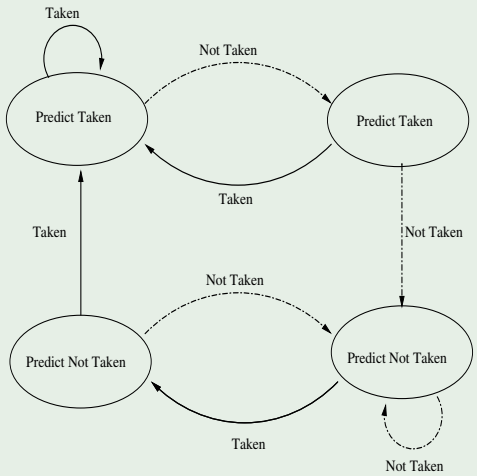
Montgomery Ladder Exponentiation Algorithm

- A naïve modification is to have a balanced ladder structure having equal number of squarings and multiplications.
- Most popular exponentiation primitive for Asymmetric-key cryptographic implementations.

Algorithm 2: Montgomery Ladder Algorithm

$R_0 \leftarrow 1$;
$R_1 \leftarrow M$;
for i **from** 0 **to** $n-1$ **do**
  if $d_i = 0$ **then**
    $R_1 \leftarrow (R_0 * R_1) \mod N$;
    $R_0 \leftarrow (R_0 * R_0) \mod N$;
  end
  else
    $R_0 \leftarrow (R_0 * R_1) \mod N$;
    $R_1 \leftarrow (R_1 * R_1) \mod N$;
  end
end
return R_0 ;

Approximating the System predictor with 2-bit branch predictor [7]



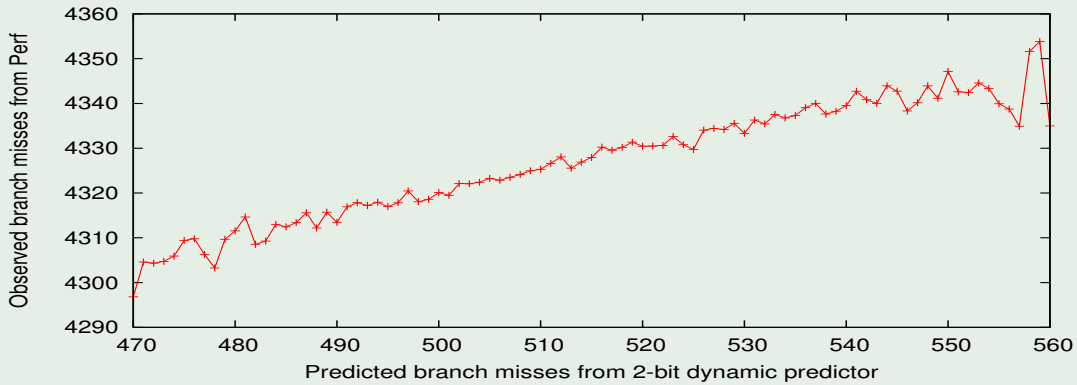


Figure: Variation of branch-misses from performance counters with increase in branch miss from 2-bit predictor algorithm

- Direct correlation observed for the branch misses from HPCs and from the simulated 2-bit dynamic predictor over a sample of exponent bitstream.
- This confirms assumption of 2-bit dynamic predictor being an approximation to the underlying system branch predictor.

Formalizing the differential of 2-bit predictor in fault attack setup

- We model the strong effect of the bimodal predictor to exploit the side-channel leakage of branch misses from the performance counters.
- Also we characterize the differential of branch misses from correct and faulty branching sequences based on the behavior of 2-bit predictor.

Various parameters used during the analysis are defined as follows:

- There is a sequence of n branches denoted as $(b_0, b_1, \cdots, b_{n-1})$ generated from execution of the algorithm under attack.
- A fault at the i^{th} execution of the algorithm changes the branching decision for the i^{th} instance.
- Difference in branch misses (Δ_i) between the correct branching sequence $(b_0, b_1, \cdots, b_i, \cdots, b_{n-1})$ and the faulty sequence $(b_0, b_1, \cdots, \overline{b_i}, \cdots, b_{n-1})$ simulated theoretically over a 2-bit predictor algorithm can be atleast -3 and atmost 3.

Some more parameters

Table: Tabular Representation of Symbols

Symbols	Meanings with respect to their analysis
$(b_0, b_1, \cdots, b_{i-1})$	Sequence of taken or not-taken known branches
St_j^K	State of 2-bit predictor after j conditional branches with respect to the Correct Sequence
$St_j^{F_i}$	State of 2-bit predictor after j conditional branches with respect to the Faulty Sequence
P_{j+1}^K	Branch predicted by 2-bit predictor for branch statement corresponding to $(j+1)^{th}$ bit of Correct Sequence
$P_{j+1}^{F_i}$	Branch predicted by 2-bit predictor for branch statement corresponding to $(j+1)^{th}$ bit of Faulty Sequence

Formalizing 2-bit predictor behavior

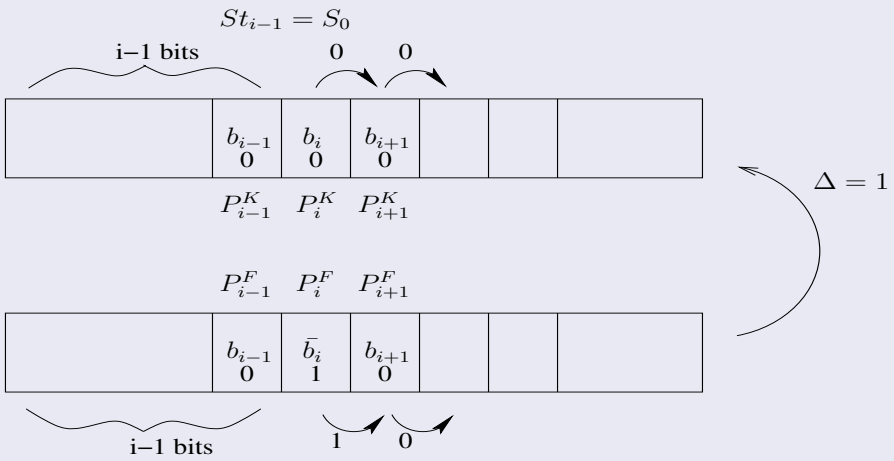
Properties

- Property 1:
  If $St^K_{i-1} = S_0$ or $St^K_{i-1} = S_2$, then $P^K_i = P^F_i = b_{i-1}$.

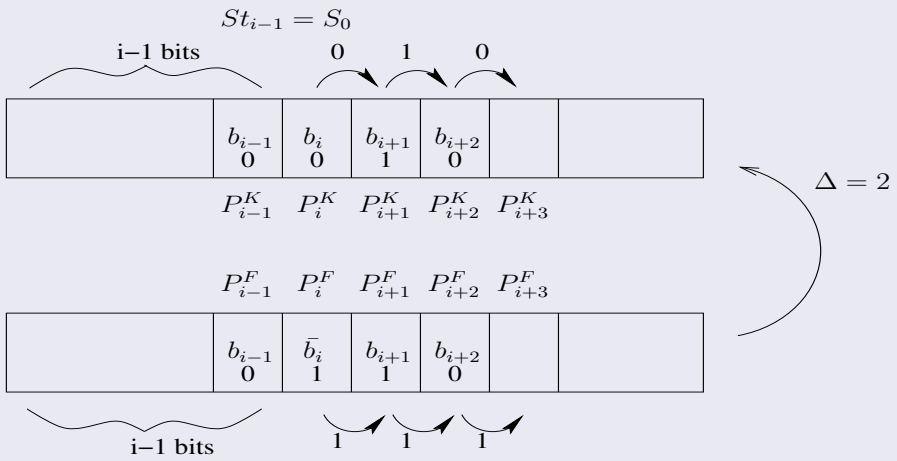
- Property 2:
  If $St^K_{i-1} = S_0$ or $St^K_{i-1} = S_2$, then there are guaranteed mispredictions for branch statement at the i^{th} instance for either K or F_i. If the branch statement corresponding to $(i+1)^{th}$ instance is not same as the predicted P^K_i, then there is a mismatch between the correct and the faulty sequence in the predictor's output for the $(i+2)^{th}$ position as $P^K_{i+2} \neq P^{F_i}_{i+2}$.

Differentials over 2-bit predictor

If $St^K_{i-1} = S_0$ and $b_i = 0$ then $\Delta_i \in \{0, 1, 2, 3\}$



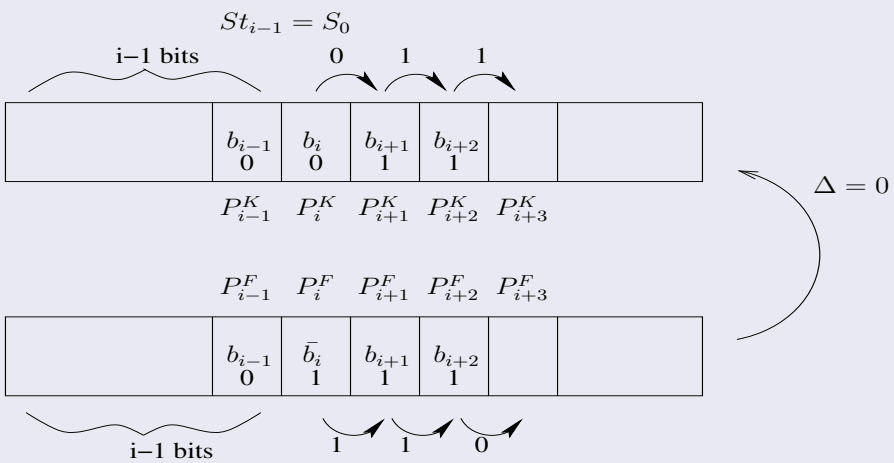
(a) $\Delta_i = 1$



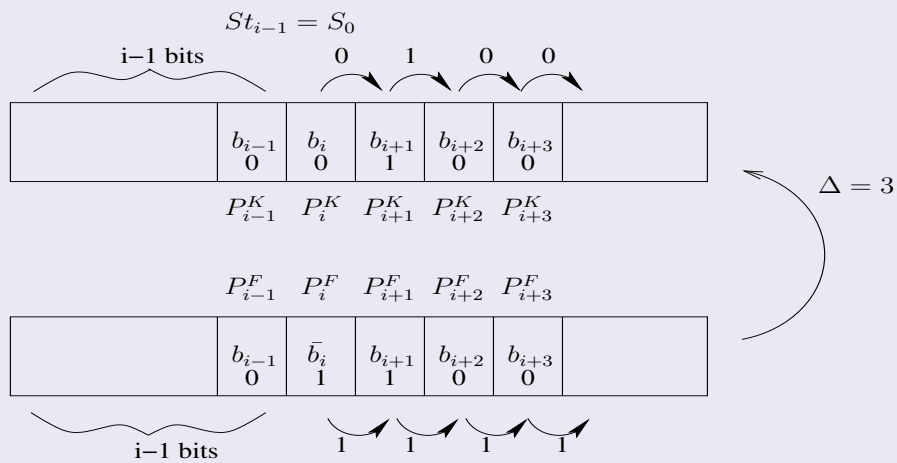
(b) $\Delta_i = 2$

Figure: Variation of simulated branch-misses on the i^{th} branching decision having $St_{i-1} = S_0$

If $St^K_{i-1} = S_0$ and $b_i = 0$ then $\Delta_i \in \{0, 1, 2, 3\}$



(a) $\Delta_i = 0$



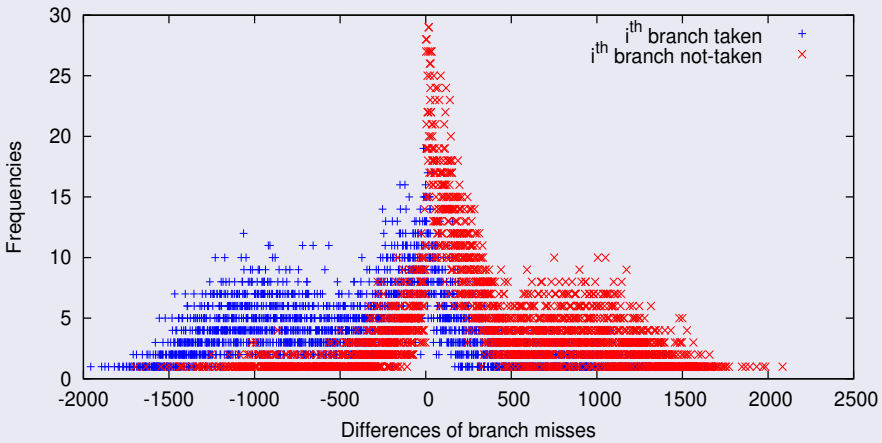
(b) **Maximum Variation**

Figure: Variation of simulated branch-misses on the i^{th} branching decision having $St_{i-1} = S_0$

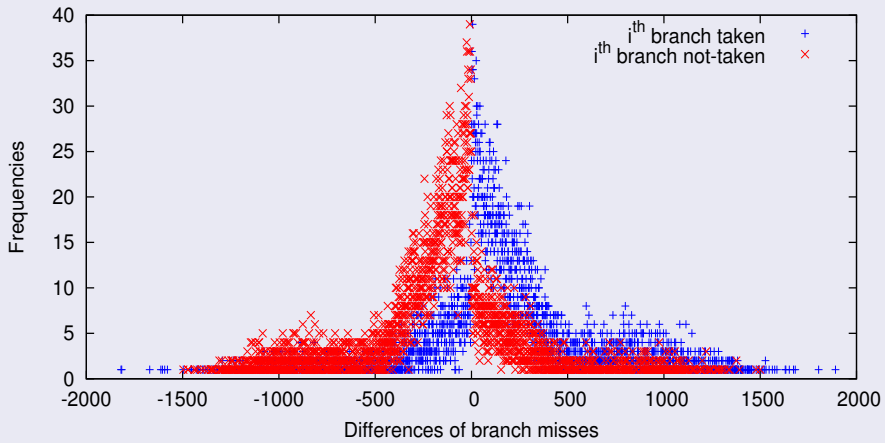
1. **If** $St_{i-1}^{K} = S_0$ **and** $b_i = 0$ **then** $\Delta_i \in \{0, 1, 2, 3\}$
2. **If** $St_{i-1}^{K} = S_0$ **and** $b_i = 1$ **then** $\Delta_i \in \{0, -1, -2, -3\}$
3. **If** $St_{i-1}^{K} = S_2$ **and** $b_i = 0$ **then** $\Delta_i \in \{0, -1, -2, -3\}$, and
4. **If** $St_{i-1}^{K} = S_2$ **and** $b_i = 1$ **then** $\Delta_i \in \{0, 1, 2, 3\}$

Differential behavior of HPC due to an i^{th} bit fault

- The secret and faulty sequences only differ at the i^{th} bit, the previous 0^{th} to $(i-1)^{th}$ bits being same for both the exponents, the branch sequences corresponding to secret and its faulty counterpart varies only at the i^{th} bit.
- Initially the adversary observes the number of branch misses for exponentiation operation using the secret exponent from HPCs.
- In the next step, a fault induced at the target bit of secret key, simultaneously observing the number of branch misses from HPCs for exponentiation using the faulty exponent.
- The difference of branch misses obtained through HPCs is denoted as δ_i.



(a) **when** $St^K_{i-1} = S_0$



(b) $St^K_{i-1} = S_2$

Figure: **Variation of branch-misses from performance counters based on the i^{th} branching decision**

If $St^K_{i-1} = S_0$,

- If $b_i = 0$, then $\delta_i > 0$
- Else if $b_i = 1$, then $\delta_i < 0$

If $St^K_{i-1} = S_2$,

- If $b_i = 0$, then $\delta_i < 0$
- Else if $b_i = 1$, then $\delta_i > 0$

Developing the Attack Algorithm

Let δ_i be the differences of branch misses over the secret and faulty exponent observed from the HPCs. We determine the next bit nb_i as,

If $St_{i-1}^K = S_0/S_2$:

- If $\delta_i < 0$,
  - $nb_i = 0$, if $St_{i-1}^K = S_2$ and
  - $nb_i = 1$, when $St_{i-1}^K = S_0$.
- Else if $\delta_i > 0$
  - $nb_i = 0$, if $St_{i-1}^K = S_0$ and
  - $nb_i = 1$, when $St_{i-1}^K = S_2$.

Else if, $St^K_{i-1} = S_1/S_3$:
If we flip the $(i-1)^{th}$ bit, the state upto $(i-1)^{th}$ bit changes to S_0 or S_2.

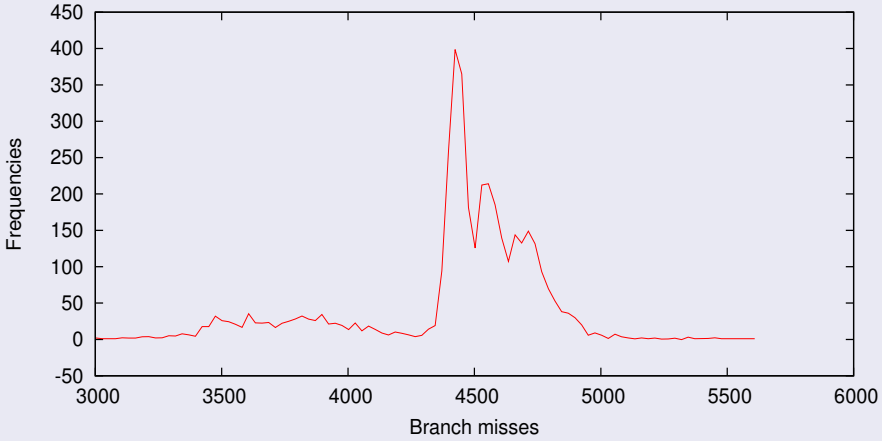
- the characteristic property for $St_{i-1} = S_1/S_3$ is such that $b_{i-2} = P_{i-1} = P_i \neq b_{i-1}$.

If we inject a fault at $(i-1)^{th}$ position then branching decision b_{i-1} gets complemented. Effectively, if $St^K_{i-1} = S_1$ previously then after fault $St^{F_{i-1}}_{i-1}$ becomes S_0. Similarly, if $St^K_{i-1} = S_3$ previously then after fault $St^{F_{i-1}}_{i-1}$ becomes S_2.

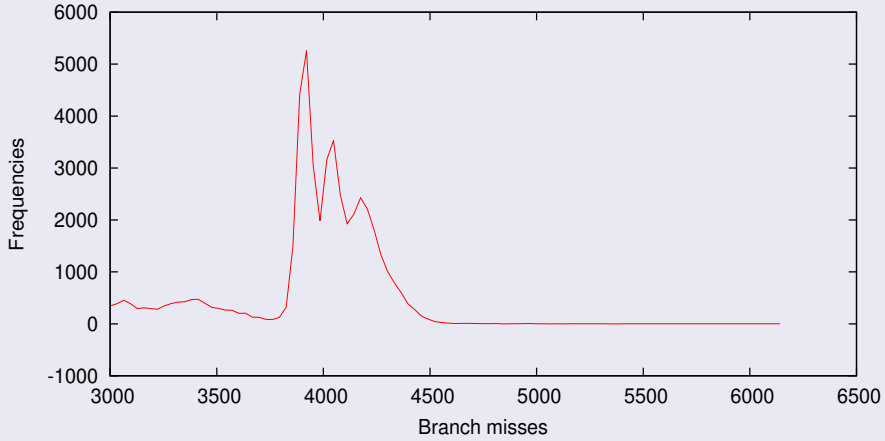
Let $\delta_{i-1,i}$ be the differences of branch misses over the faulty exponents observed from the HPCs. We determine the next bit nb_i as,

- If $\delta_{i-1,i} < 0$,
  - $nb_i = 0$, if $St^K_{i-1} = S_3$ and
  - $nb_i = 1$, when $St^K_{i-1} = S_1$.
- Else if $\delta_{i-1,i} > 0$
  - $nb_i = 0$, if $St^K_{i-1} = S_1$ and
  - $nb_i = 1$, when $St^K_{i-1} = S_3$.

Modelling the System Noise



(a) **Due to exponentiation on secret exponent**



(b) **Due to environmental processes running in the system**

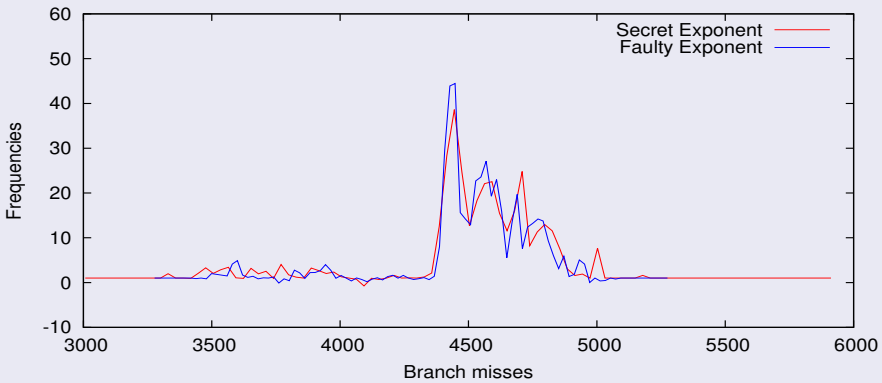
Figure: Distribution of branch-misses of secret and faulty exponent on square and multiply implementation from HPCs having $St_{i-1} = S_0$

Fig.(a) has similar nature to this noise distribution in Fig.(b) with a shift in the respective statistics with an increase in branch misprediction due to the conditional statements from the secret exponents.

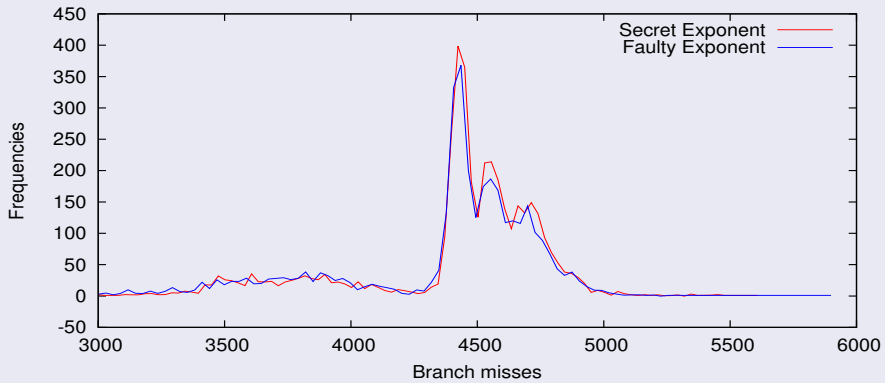
Validation of the Attack Algorithm

- We present the validation of previous discussion through experiments on 1024 bits of RSA.
- The fault model is simulated in software.
- Experiments are performed on various platforms as Core-2 Duo E7400, Intel Core i3 M350 and Intel Core i5-3470.

Experiments on Square and Multiply Algorithm



(a) $b_i = 0$ **and** $\delta_i = 14.014$

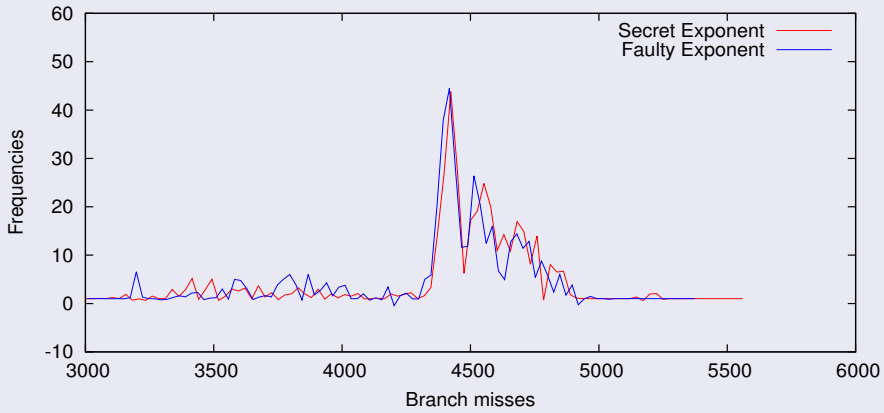


(b) $b_i = 1$ **and** $\delta_i = -35.79$

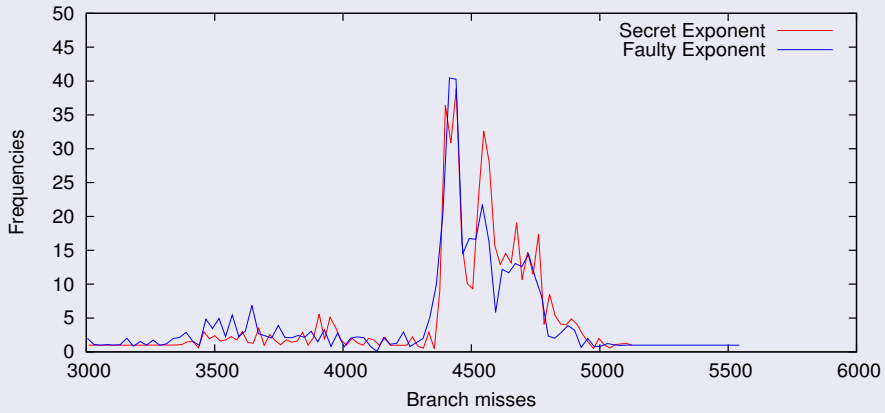
Figure: Distribution of branch-misses of secret and faulty exponent on square and multiply implementation from HPCs having $St_{i-1} = S_0$

- Fig.(a) show distribution of branch misses from the square and multiply exponentiation having $St_{i-1} = S_0$ for $b_i = 0$ and the fault being introduced at $i = 1019^{th}$ position.
- $\delta_i = 14.014$ and since $St_{i-1} = S_0$, and with positive value of δ_i, the next branch is decided as $nb_i = 0$ and $k_i = \overline{b_i}$.
- Similarly, Fig.(b) $i = 548^{th}$ location having $b_i = 1$ and $St_{i-1} = S_0$, we observed $\delta_i = -35.79$ which correctly decides the i^{th} branch as 1.

Experiments on Montgomery Ladder



(a) $b_i = 0$ **and** $\delta_i = 9.828$

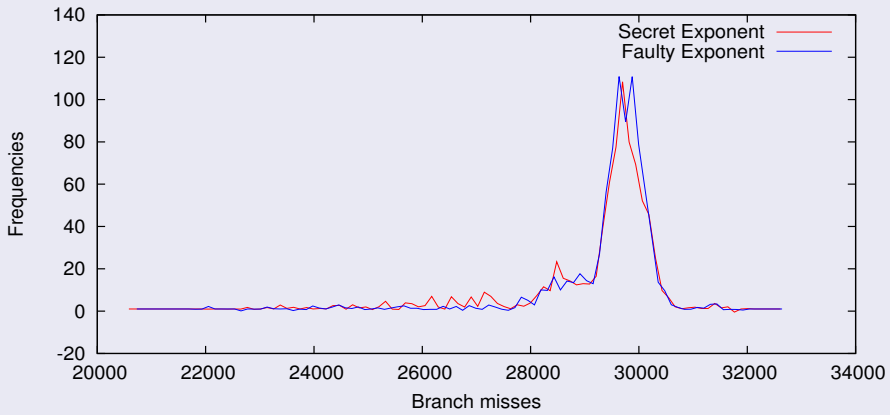


(b) $b_i = 1$ **and** $\delta_i = -139.086$

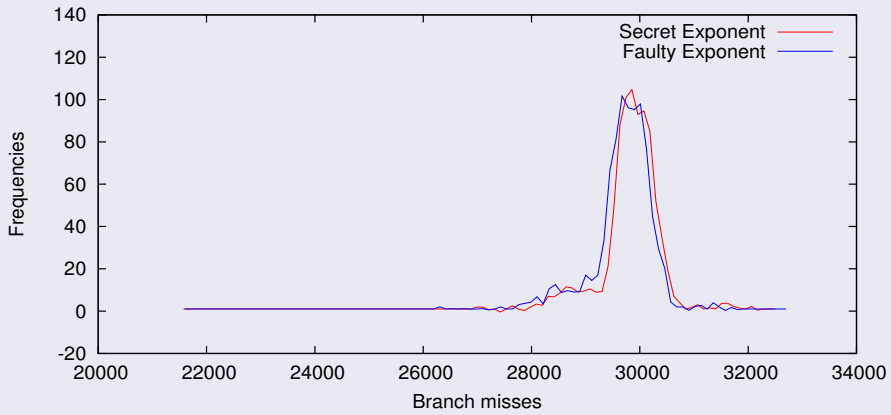
Figure: Distribution of branch-misses of secret and faulty exponent on Montgomery Ladder implementation from HPCs having $St_{i-1} = S_0$

- Fig.(a) shows for $k_i = 1$ for $i = 248$ where $St_{i-1} = S_0$, $b_i = 0$ and the branch misses from HPCs $\delta_i = 9.828$ reveals a positive difference correctly identifying $nb_i = 0$.
- While Fig.(b) shows a negative difference $\delta_i = -139.086$ correctly identifying $k_1 = 0$ for $i = 337$.

Attacks on CRT-RSA implementation



(a) $d_{p_i} = 0$ **and** $\delta_i = 243.212$



(b) $d_{p_i} = 1$ **and** $\delta_i = -136.029$

Figure: Distribution of branch-misses of secret and faulty exponent on CRT-RSA implementation from HPCs having $St_{i-1} = S_0$

- Fig.(a),(b) show two instances of the CRT-RSA implementation with square and multiply and simulated fault induced in d_p, while exponentiation for d_q is computed unaffected.
- In both situation, the target exponent bits of d_p are shown to be retrieved correctly and uniquely.

Conclusion

- HPCs used as performance monitors in modern systems can be observed by adversaries to determine critical information of secret key bits.

Conclusion

- HPCs used as performance monitors in modern systems can be observed by adversaries to determine critical information of secret key bits.
- The attack we illustrate exploit strong correlation of the 2-bit dynamic predictor to unknown underlying branch predictor of the system.

Conclusion

- HPCs used as performance monitors in modern systems can be observed by adversaries to determine critical information of secret key bits.
- The attack we illustrate exploit strong correlation of the 2-bit dynamic predictor to unknown underlying branch predictor of the system.
- We present a differential fault analysis to show that difference of branch misses for a 2-bit predictor can be utilized to reveal information of key bits.

Conclusion

- HPCs used as performance monitors in modern systems can be observed by adversaries to determine critical information of secret key bits.
- The attack we illustrate exploit strong correlation of the 2-bit dynamic predictor to unknown underlying branch predictor of the system.
- We present a differential fault analysis to show that difference of branch misses for a 2-bit predictor can be utilized to reveal information of key bits.
- The attacks can be adapted to embedded soft-core processors with practical faults being introduced by instruction skips.

Conclusion

- HPCs used as performance monitors in modern systems can be observed by adversaries to determine critical information of secret key bits.
- The attack we illustrate exploit strong correlation of the 2-bit dynamic predictor to unknown underlying branch predictor of the system.
- We present a differential fault analysis to show that difference of branch misses for a 2-bit predictor can be utilized to reveal information of key bits.
- The attacks can be adapted to embedded soft-core processors with practical faults being introduced by instruction skips.
- Interestingly, fault attack countermeasures which stop or randomize the output when a fault occurs can still be attacked using these techniques.

Conclusion

- HPCs used as performance monitors in modern systems can be observed by adversaries to determine critical information of secret key bits.
- The attack we illustrate exploit strong correlation of the 2-bit dynamic predictor to unknown underlying branch predictor of the system.
- We present a differential fault analysis to show that difference of branch misses for a 2-bit predictor can be utilized to reveal information of key bits.
- The attacks can be adapted to embedded soft-core processors with practical faults being introduced by instruction skips.
- Interestingly, fault attack countermeasures which stop or randomize the output when a fault occurs can still be attacked using these techniques.
- The work raises the question of secured implementation of ciphers in presence of HPCs in modern processors where fault inductions are feasible.

AVR Freaks.

Instruction skipping after spurious interrupt,
http://www.avrfreaks.net/forum/solved-instruction-skipping-after-spurious-interrupt, 2015.

mp-fpga.

Performance Counter for Microblaze, http://mp-fpga.blogspot.in/2007/10/performance-counter-for-microblaze.html, 2007.

Wikipedia.

Rowhammer wikipedia page, https://en.wikipedia.org/wiki/Row-hammer, 2016.

Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji-Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu.
Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors.
In *ACM/IEEE 41st International Symposium on Computer Architecture, ISCA 2014, Minneapolis, MN, USA, June 14-18, 2014*, pages 361–372. IEEE Computer Society, 2014.

Sarani Bhattacharya and Debdeep Mukhopadhyay.
Curious case of rowhammer: Flipping secret exponent bits using timing analysis.
In *CHES*, 2015.

Paul C. Kocher.
Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems.
In Neal Koblitz, editor, *CRYPTO '96: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113, London, UK, 1996. Springer-Verlag.

Sarani Bhattacharya and Debdeep Mukhopadhyay.
Who watches the watchmen?: Utilizing performance monitors for compromising keys of RSA on intel platforms.
In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 248–266. Springer, 2015.