Modeling, Derivation, and Automated Analysis of Branch Predictor Security Vulnerabilities

Quancheng Wang, Ming Tang, Ke Xu, Han Wang

Wuhan University

2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)
Session 3B: Side-Channel & Microarchitecture
Edinburgh, March 2-6, 2024

Background: Evolution of BP Attacks

- **Attacks and CVEs against branch predictors are proliferating**
- **Manual search for branch predictor attacks is not exhaustive**

- 2018 — Spectre Attacks
- 2019 — NetSpectre, SMoTherSpectre
- 2020 — BranchScope, Bluethunder, Speculative Inteference
- … — …
- 2022 — unXpec, BHI, PACMAN, RetBleed
- …


Wednesday, January 3, 2018

Reading privileged memory with a side-channel

Posted by Jann Horn, P

We have discovered tha
speculated execution, le
boundaries in various co

BRANCH HISTORY INJECTION

On the Effectiveness of Hardware Mitigations Against Cross-Privilege Spectre-v2 Attacks

modern systems
leaking arbitrary

Retbleed: Arbitrary Speculative Code Execution with Return Instructions

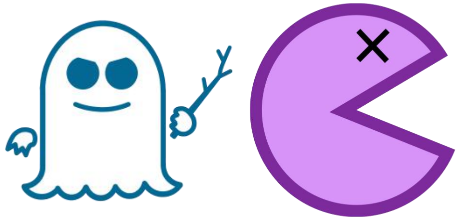
Retbleed (CVE-2022-29900 and CVE-2022-29
speculative execution attacks that exploit bra
we call Spectre-BTI. Unlike its siblings, who tri
exploiting indirect jumps or calls, Retbleed ex
deal, since it undermines some of our current

MIT News
ON CAMPUS AND AROUND THE WORLD

Researchers discover a new hardware vulnerability in the Apple M1 chip

Background: Evolution of BP Attacks

- Attacks and CVEs against branch predictors are proliferating
- Manual search for branch predictor attacks is not exhaustive



- 2018 — Spectre Attacks
- 2019 — NetSpectre, SMoTherSpectre
- Speculative Interference
- … — …
- 2022 — unXpec, BHI, PACMAN, RetBleed
- …

A trustworthy tool is essential for exploring all branch predictor attacks!

Background: Insufficient Security Evaluation

WUHAN UNIVERSITY

➢ Weak security evaluation of many defenses

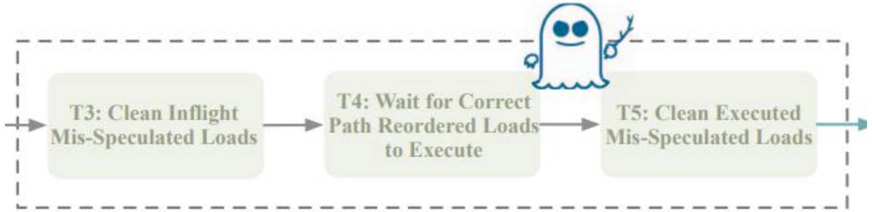
defense

gadgets

gem5

simulator

The defense is **secure** because it can prevent against all attack gadgets



unXpec (HPCA'22) bypasses CleanupSpec (MICRO'19)

➢ Existing principled approaches are not comprehensive

Software-Level

- KLEESpectre
- Spectector
- SpecuSym
- …

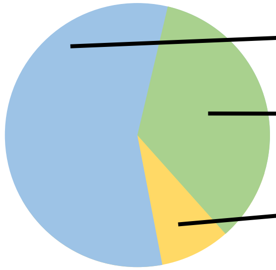
Failing to analyze the root cause

Only targeting existing attacks

Cannot evaluate hardware defenses

Weak Security Guarantee

Microarchitecture-Level



Formal-Based → Cannot assess timing attacks and secure BPs

Symbol-Based → Cannot evaluate branch predictor vulnerabilities

Graph-Based → Only targeting existing speculative attack types

Background: Insufficient Security Evaluation

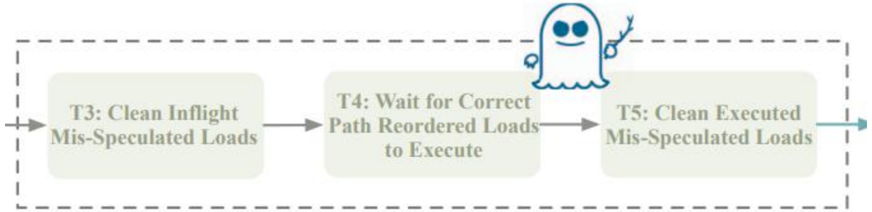
- ➢ **Weak security evaluation of many defenses**

defense

gadgets

gem5 simulator

The defense is **secure** because it can prevent against all attack gadgets



unXpec (HPCA'22) bypasses CleanupSpec (MICRO'19)

A comprehensive security evaluation is imperative for defense solutions!

Software-Level

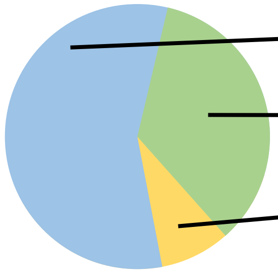
- Spectector
- SpecuSym
- ...

Only targeting existing attacks

Cannot evaluate hardware defenses

Weak Security Guarantee

Microarchitecture-Level



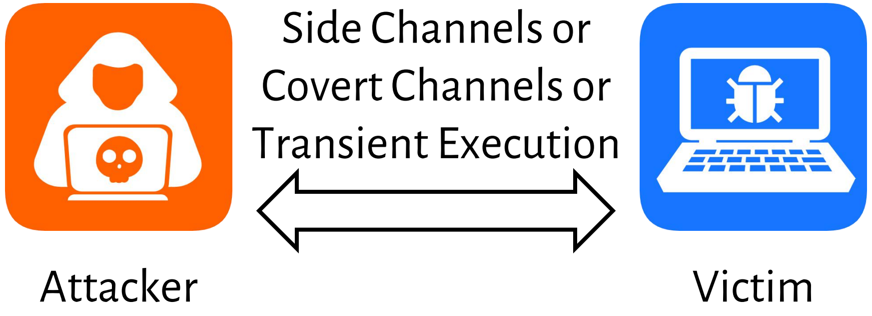
Formal-Based → Cannot assess timing attacks and secure BPs

Symbol-Based → Cannot evaluate branch predictor vulnerabilities

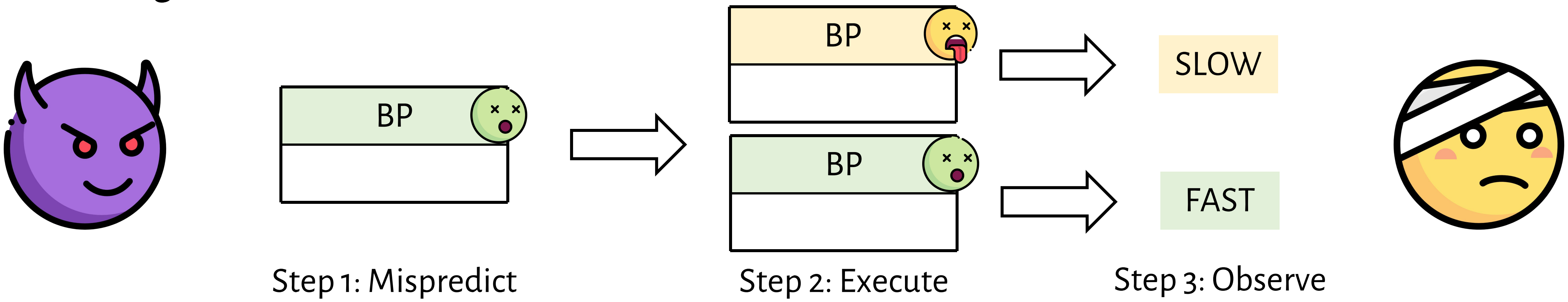
Graph-Based → Only targeting existing speculative attack types

Threat Model

- **Attacker and victim**
  - Attacker: App, OS, VM, etc.
  - Victim: App, OS, VM, TEE, etc.



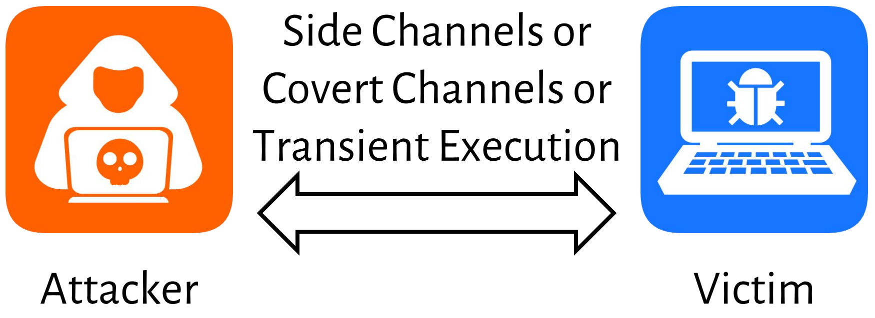
- **Attacker's goal**
  - Inferring secret data based on branch instruction execution time differences or transient execution due to misprediction
- **Attack types**
  - Timing-based attacks: side channels, covert channels



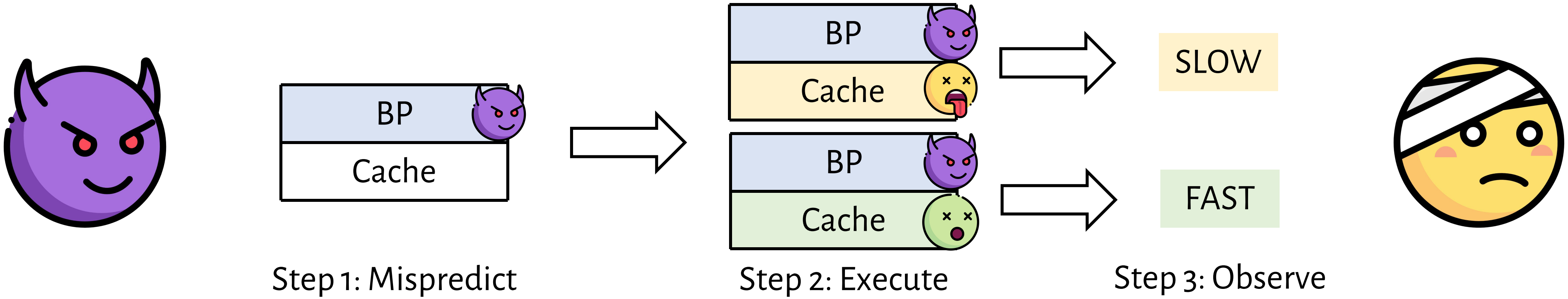
Threat Model

WUHAN UNIVERSITY

- **Attacker and victim**
  - Attacker: App, OS, VM, etc.
  - Victim: App, OS, VM, TEE, etc.

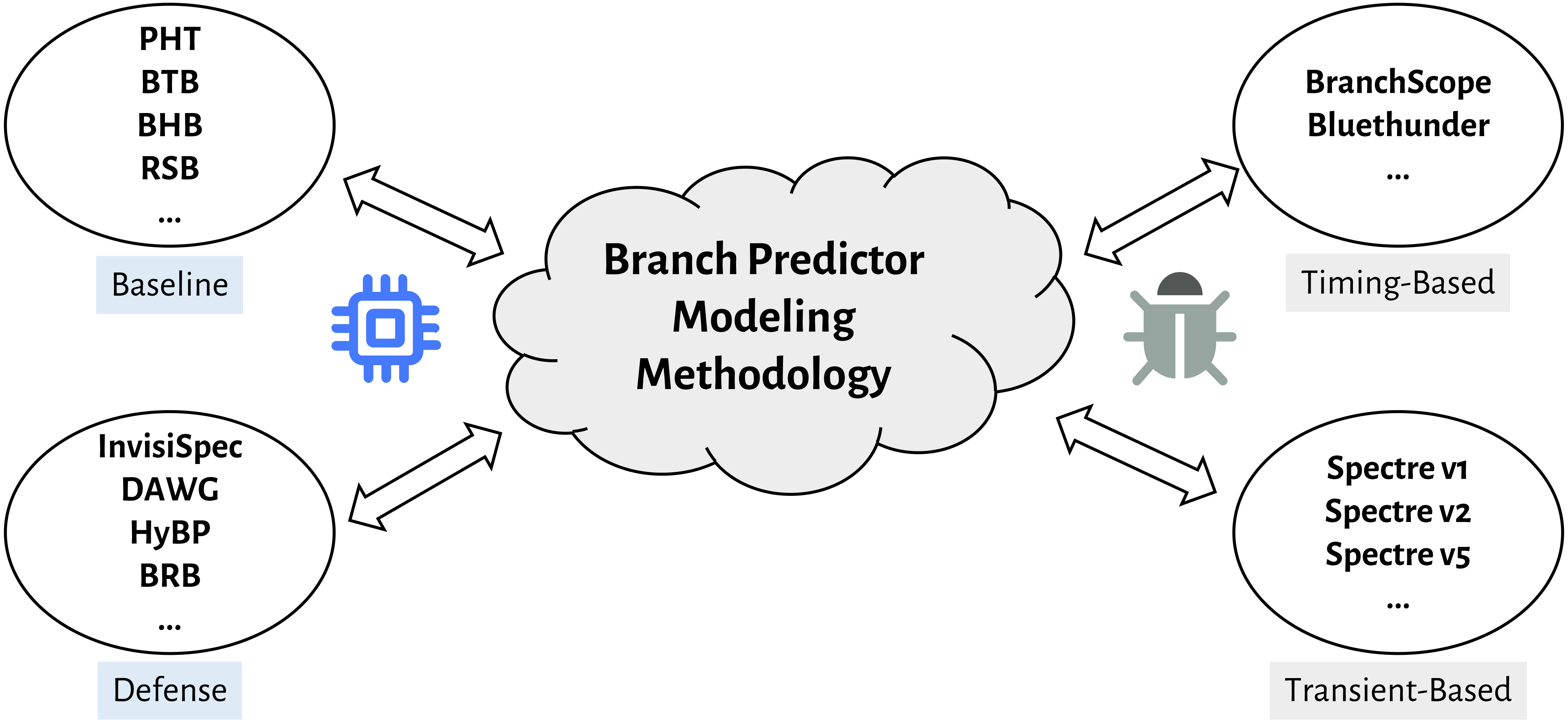


- **Attacker's goal**
  - Inferring secret data based on branch instruction execution time differences or transient execution due to misprediction
- **Attack types**
  - Transient-based attacks: speculative attacks



Challenge

➢ **How to model branch predictors for security evaluation**



Modeling: Three-Step Attack Model

- **Insights from microarchitectural attacks against branch predictors**
  - All existing branch predictor attacks include three steps

Set the state of the branch predictor to a known **initial state**	→	The state of the branch predictor should be **changed** to leak secret information	→	The attacker infers the state of the branch predictor through **timing** observation
Step 1: BR		Step 2: BR		Step 3: BR or CC

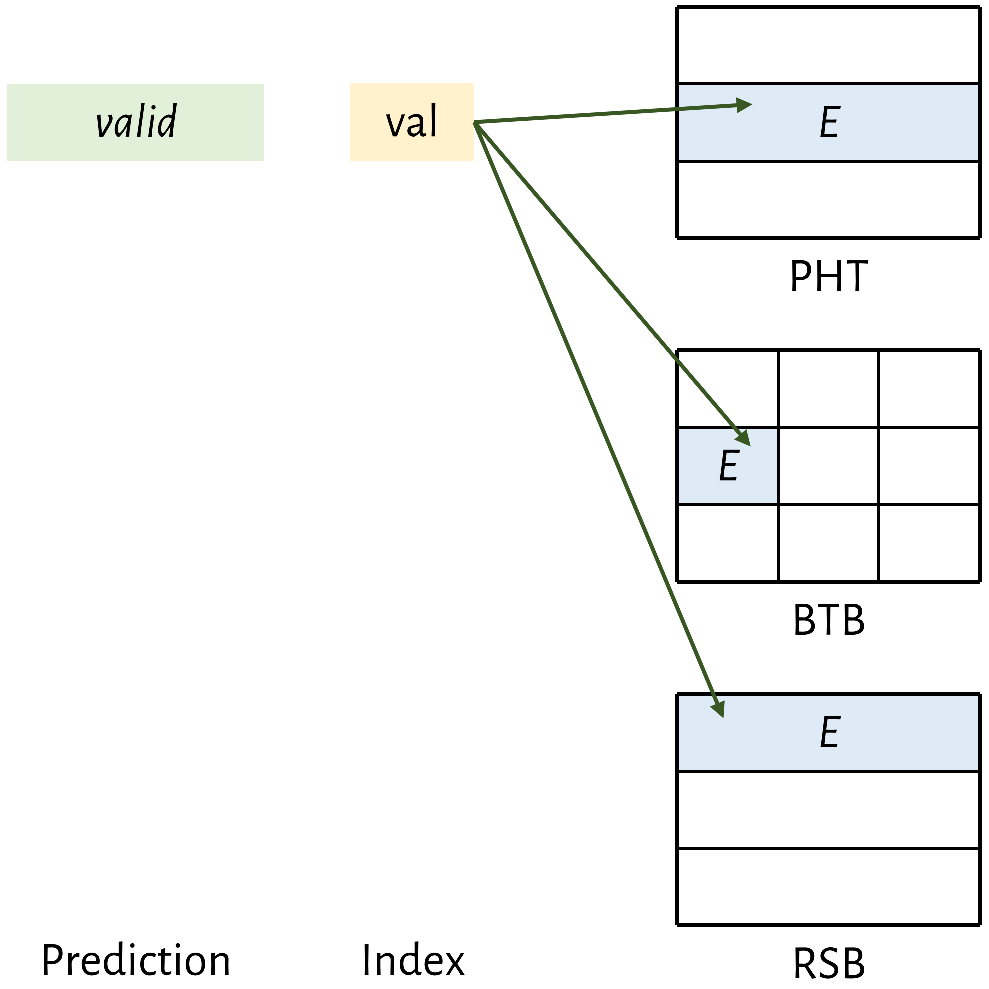
BR: branch operation
CC: covert channel

- For each branch predictor entry
  - The state of each entry is independent of other entries
  - The attacker only focus on a single entry during attacks
  - The update logic is the same for each entry

⇒ **Consider only a single entry for each branch predictor unit**

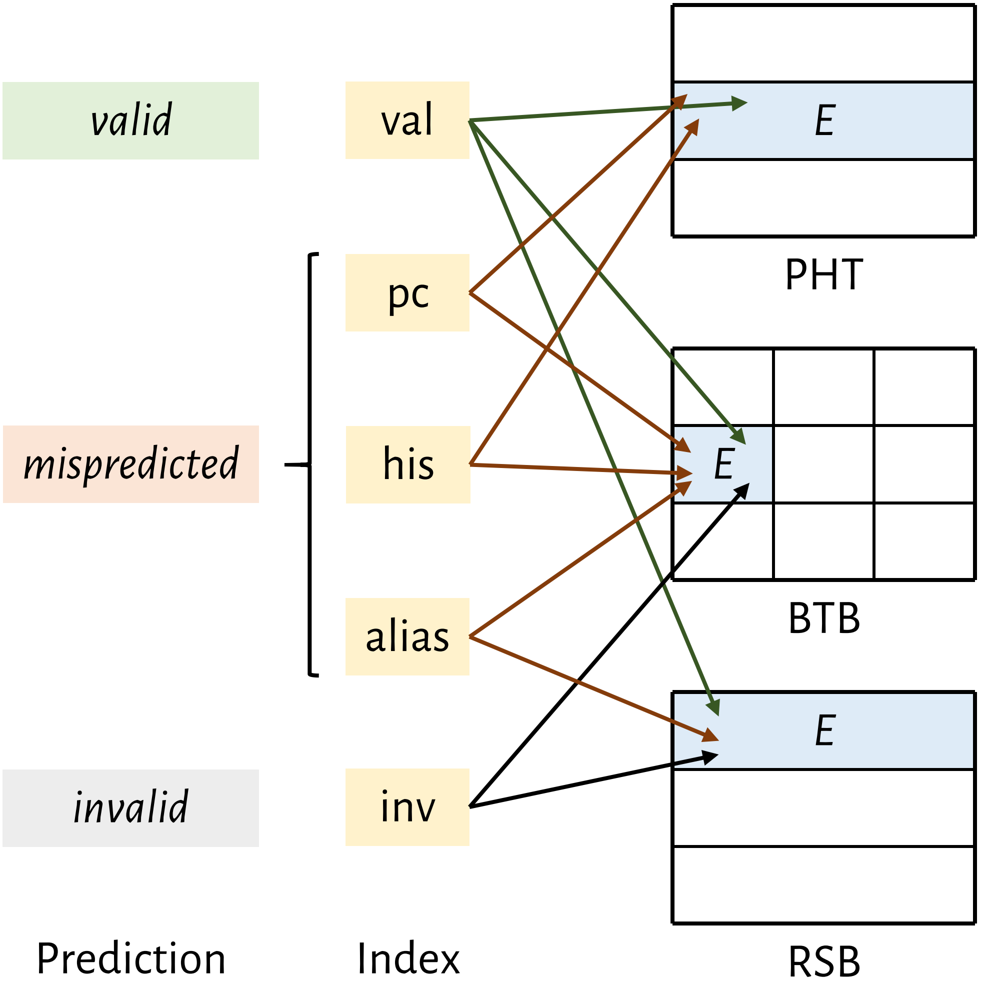
Modeling: Possible Branch Predictor States

- **Modeling 19 states of security-critical branch predictor entry *E***



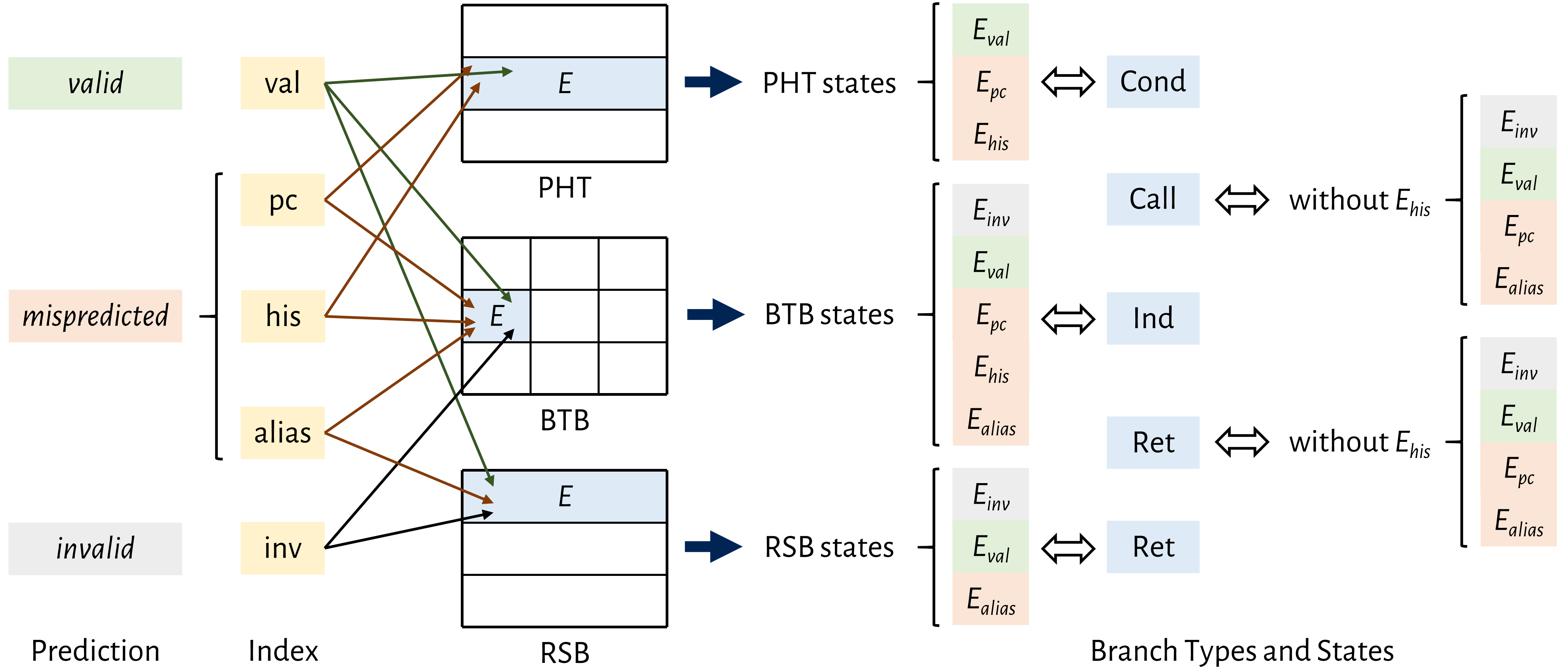
Modeling: Possible Branch Predictor States

➢ **Modeling 19 states of security-critical branch predictor entry *E***



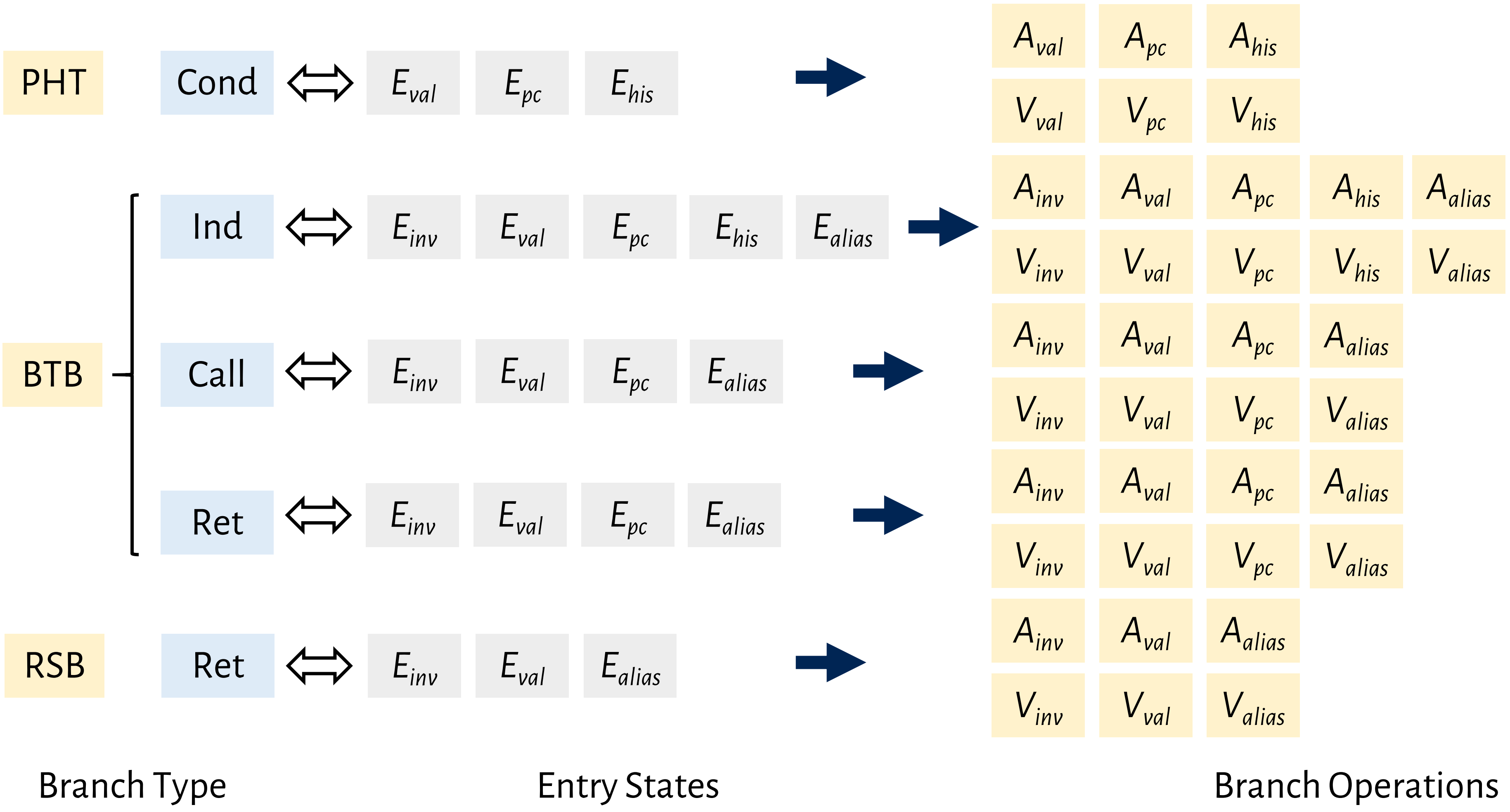
Modeling: Possible Branch Predictor States

- **Modeling 19 states of security-critical branch predictor entry *E***



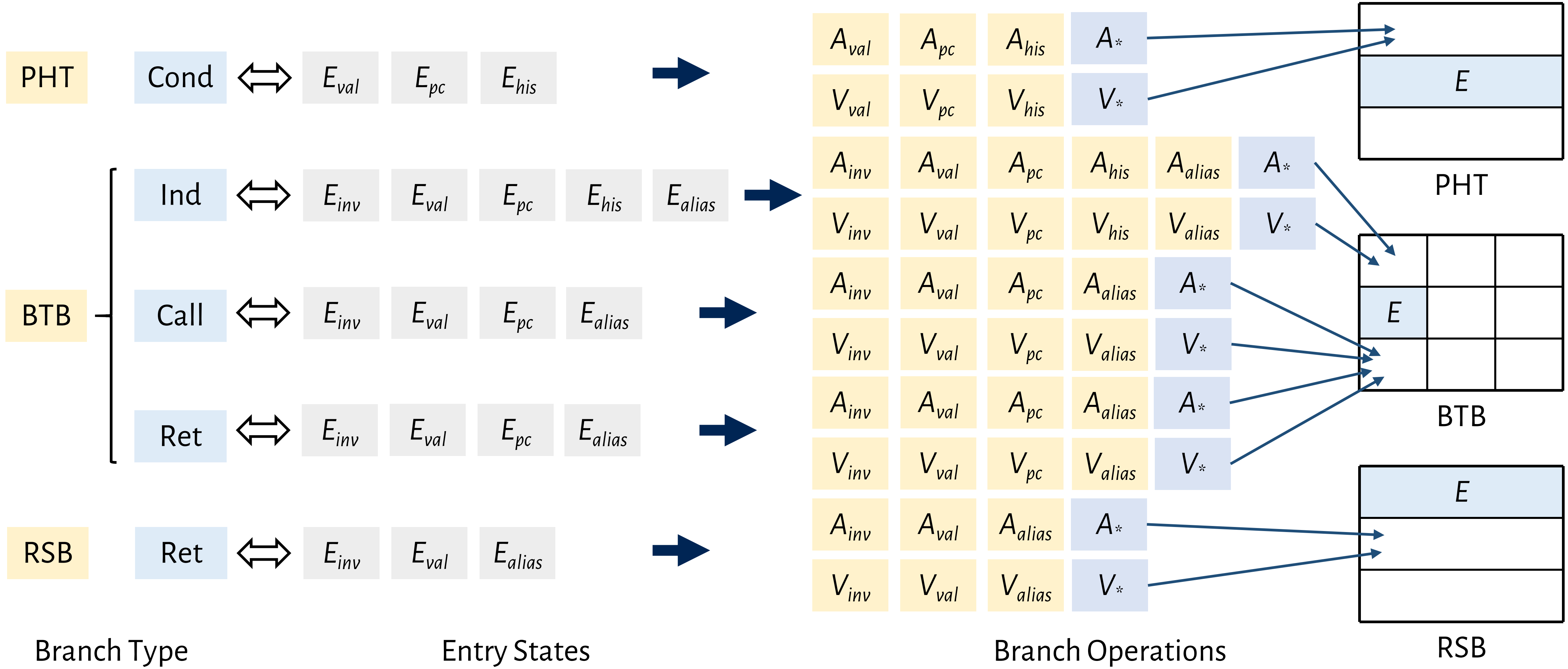
Modeling: Attacker's and Victim's Operations

➢ **Possible branch operations related to prior 19 target entry states**



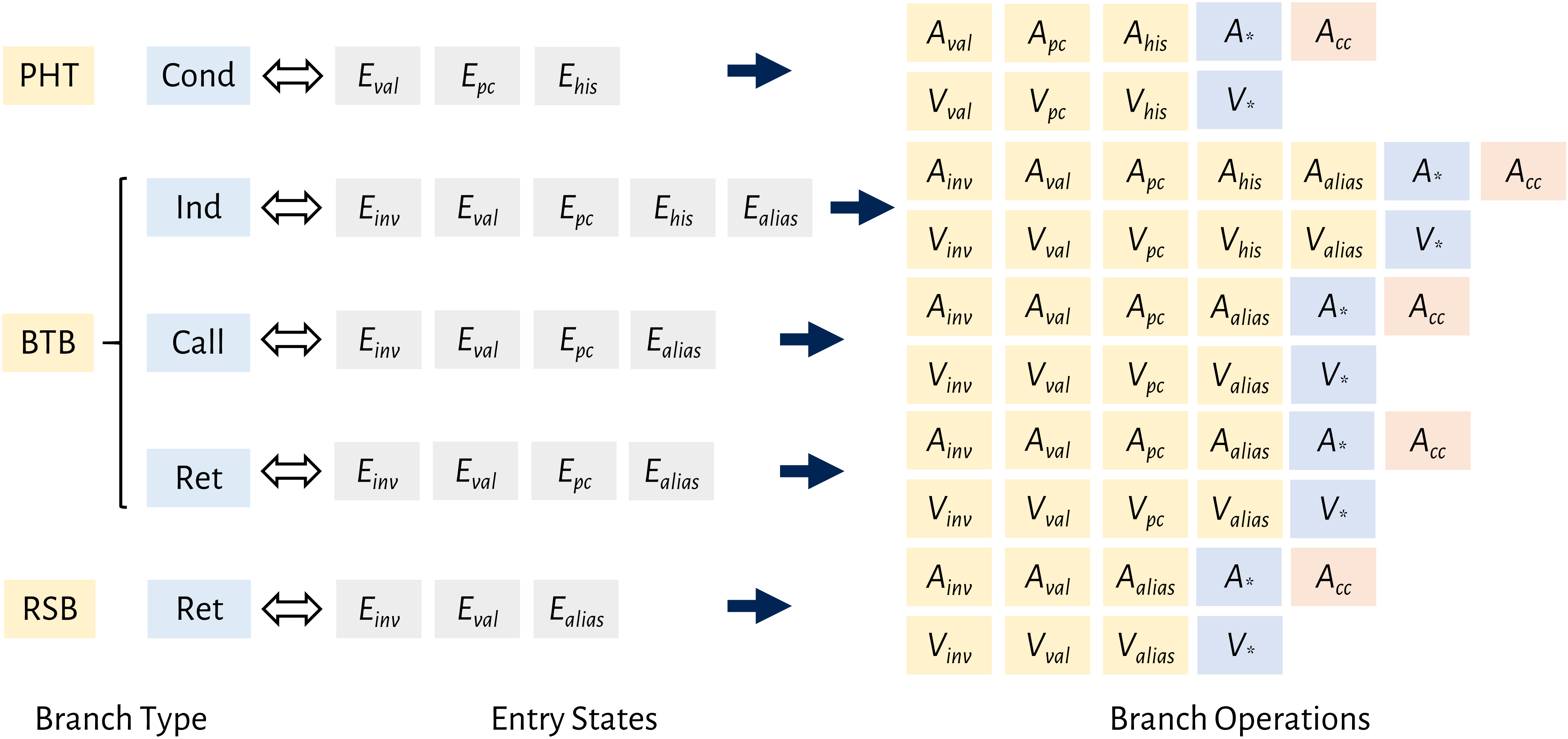
Modeling: Attacker's and Victim's Operations

- ➢ A_* or V_* indicates no operation on the target branch predictor entry



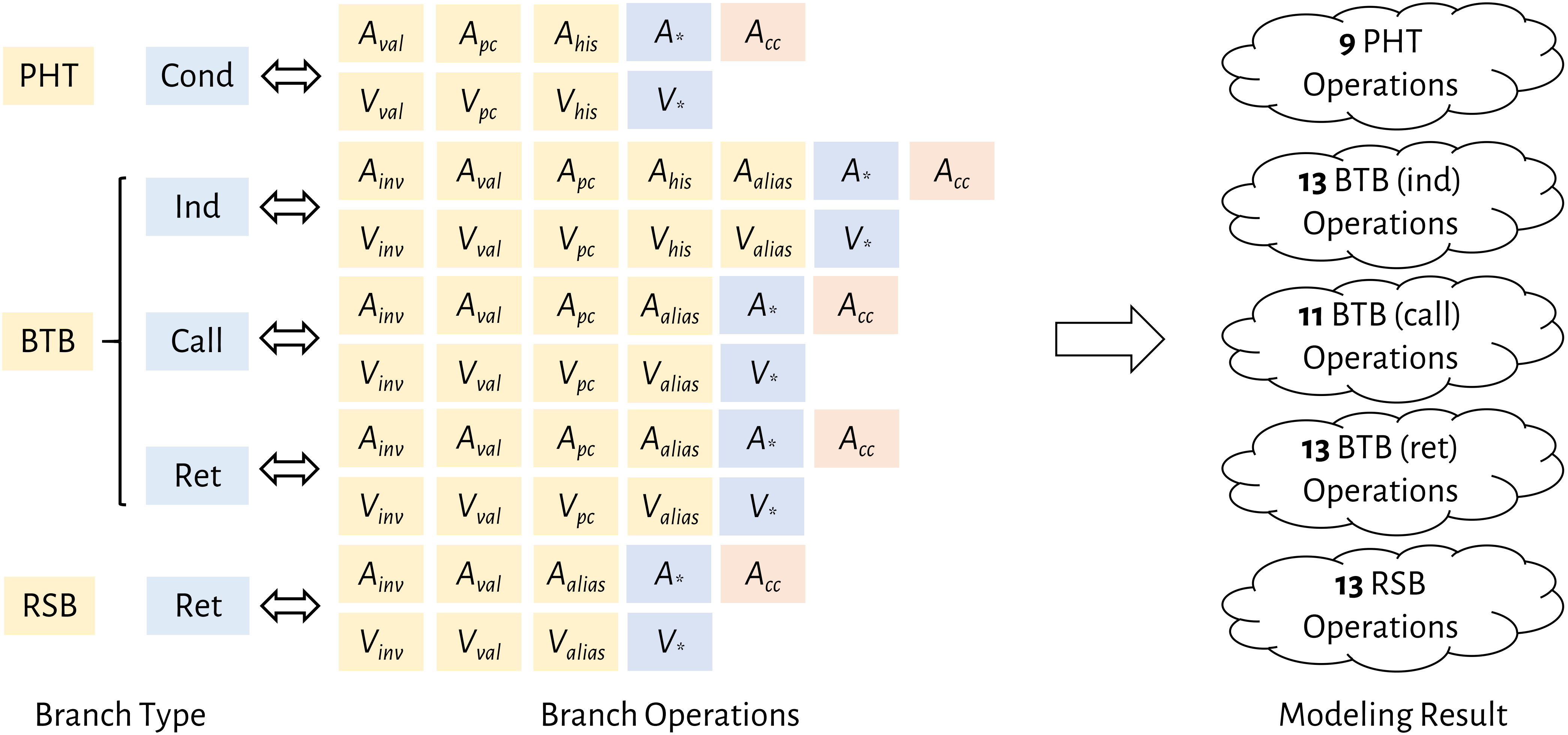
Modeling: Attacker's and Victim's Operations

➢ A_{cc} denotes the observation of the covert channel in transient attacks



Modeling: Attacker's and Victim's Operations

➤ **We finally model 53 possible operations in the three-step attack model**



Framework: Branch Predictor Simulator

WUHAN UNIVERSITY

➢ **We implement a branch predictor simulator to explore all attacks**

9 PHT Ops

13 BTB (ind) Ops

11 BTB (call) Ops

11 BTB (ret) Ops

9 RSB Ops

Input

Three-Step BP Simulator

$A_{pc} \rightarrow V_{val} \rightarrow V_{val}$

$V_{pc} \rightarrow V_{val} \rightarrow V_{val}$

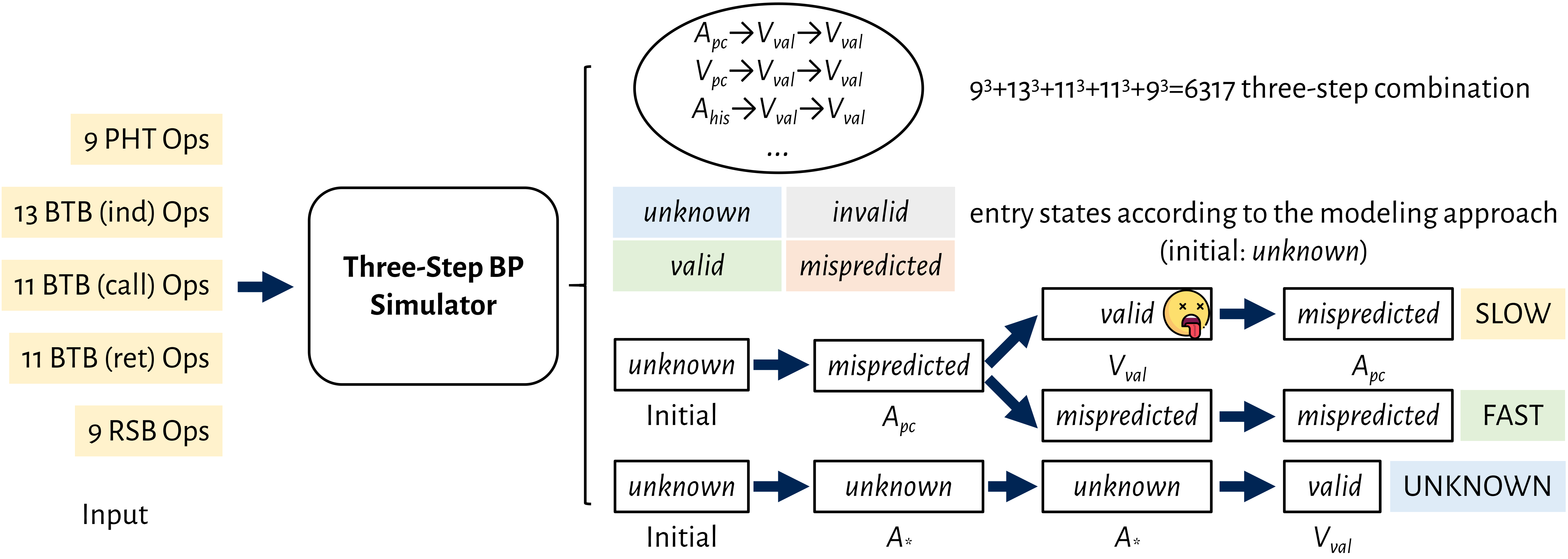
$A_{his} \rightarrow V_{val} \rightarrow V_{val}$

…

$9^3+13^3+11^3+11^3+9^3=6317$ three-step combination

Framework: Branch Predictor Simulator

➢ **We perform an enumerative analysis of each three-step combination**

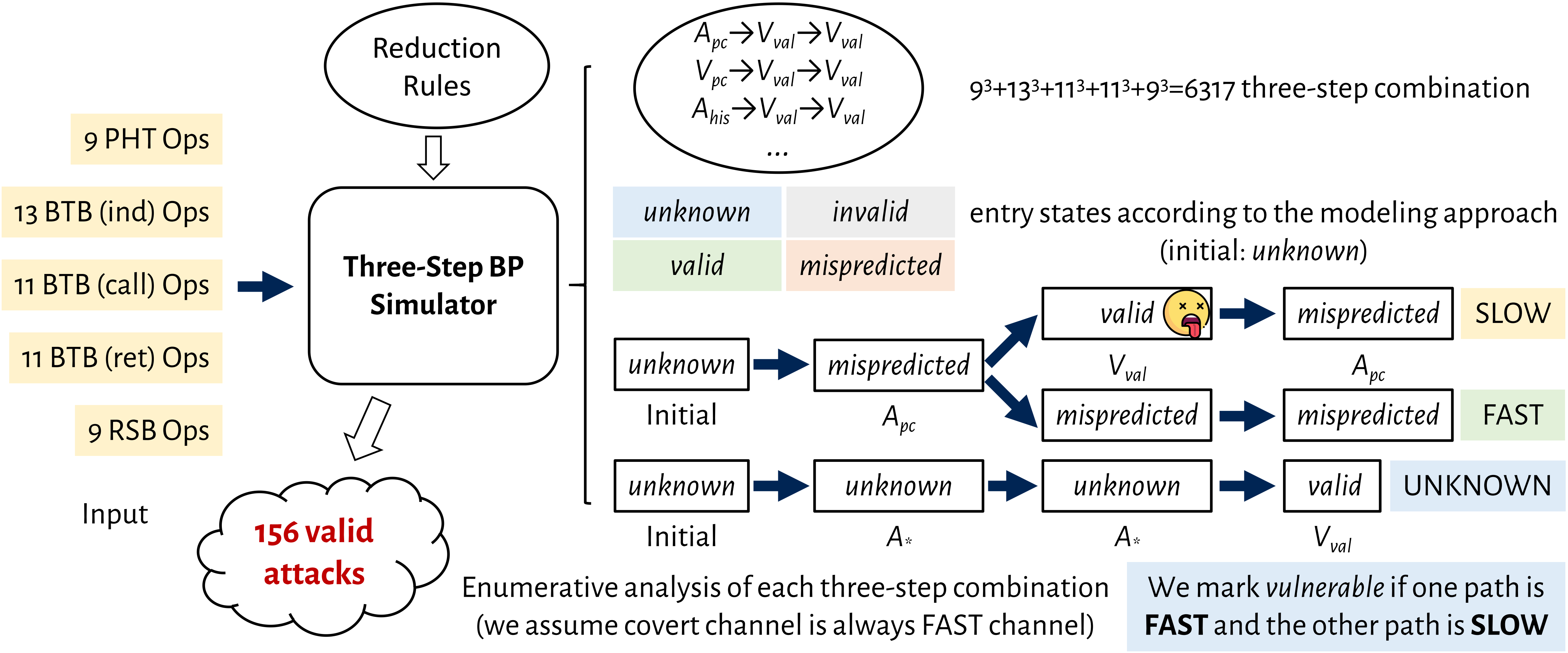


Enumerative analysis of each three-step combination (we assume covert channel is always FAST channel)

We mark *vulnerable* if one path is **FAST** and the other path is **SLOW**

Framework: Branch Predictor Simulator

➢ **We reduce redundancies and finally derive 156 valid attack patterns**



Framework: Summary of Derived Attacks

- **Summary of derived 156 attack patterns**
  - 28 PHT attacks, 116 BTB attacks and 12 RSB attacks
  - 67 known attacks and 89 novel attacks

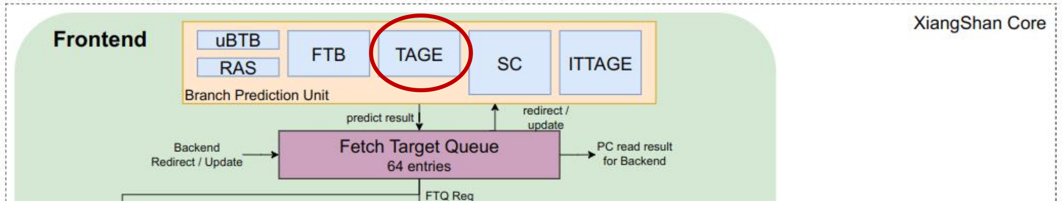
Branch Predictor	Known Attacks	Novel Attacks	Total Attacks
PHT	12	16	28
BTB (ind)	20	36	56
BTB (call)	15	15	30
BTB (ret)	15	15	30
RSB	5	7	12
Total	67	89	156

Unit	Step1	Step2	Step3	Category	Type	Attack	Step1	Step2	Step3	Category	Type	Attack
PHT	V_{val}	A_{pc}	V_{val} (slow)	EM	TSCA/CCA	new	V_{val}	V_{pc}	V_{val} (slow)	IM	TSCA/CCA	new
	V_{val}	A_{his}	V_{val} (slow)	EM	TSCA/CCA	new	V_{val}	V_{his}	V_{val} (slow)	IM	TSCA/CCA	new
	A_{pc}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	(1)	A_{pc}	V_{val}	A_{pc} (slow)	EM	TSCA/CCA	(1)
	A_{pc}	V_{val}	V_{pc} (slow)	IM	TSCA/CCA	(1)	A_{pc}	V_{val}	A_{his} (slow)	EM	TSCA/CCA	(1)
	A_{pc}	V_{val}	V_{his} (slow)	IM	TSCA/CCA	(1)	V_{pc}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	new
	V_{pc}	V_{val}	A_{pc} (slow)	EM	TSCA/CCA	new	V_{pc}	V_{val}	V_{pc} (slow)	IM	TSCA/CCA	new
	V_{pc}	V_{val}	A_{his} (slow)	EM	TSCA/CCA	new	V_{pc}	V_{val}	V_{his} (slow)	IM	TSCA/CCA	new
	A_{his}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	(2)	A_{his}	V_{val}	A_{pc} (slow)	EM	TSCA/CCA	(2)
	A_{his}	V_{val}	V_{pc} (slow)	IM	TSCA/CCA	(2)	A_{his}	V_{val}	A_{his} (slow)	EM	TSCA/CCA	(2)
	A_{his}	V_{val}	V_{his} (slow)	IM	TSCA/CCA	(2)	V_{his}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	new
	V_{his}	V_{val}	A_{pc} (slow)	EM	TSCA/CCA	new	V_{his}	V_{val}	V_{pc} (slow)	IM	TSCA/CCA	new
	V_{his}	V_{val}	A_{his} (slow)	EM	TSCA/CCA	new	V_{his}	V_{val}	V_{his} (slow)	IM	TSCA/CCA	new
	A_{pc}	V_{val}	A_{cc} (fast)	EH	TEA	new	V_{pc}	V_{val}	A_{cc} (fast)	EH	TEA	(3)
	A_{his}	V_{val}	A_{cc} (fast)	EH	TEA	new	V_{his}	V_{val}	A_{cc} (fast)	EH	TEA	(4)
BTB (ind)	A_{inv}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	(1)	V_{inv}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	new
	V_{val}	A_{pc}	V_{val} (slow)	EM	TSCA/CCA	new	V_{val}	V_{pc}	V_{val} (slow)	IM	TSCA/CCA	new
	V_{val}	A_{his}	V_{val} (slow)	EM	TSCA/CCA	new	V_{val}	V_{his}	V_{val} (slow)	IM	TSCA/CCA	new
	V_{val}	A_{alias}	V_{val} (slow)	EM	TSCA/CCA	new	V_{val}	V_{alias}	V_{val} (slow)	IM	TSCA/CCA	new
	A_{pc}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	(1)	A_{pc}	V_{val}	A_{pc} (slow)	EM	TSCA/CCA	(1)
	A_{pc}	V_{val}	V_{pc} (slow)	IM	TSCA/CCA	(1)	A_{pc}	V_{val}	A_{his} (slow)	EM	TSCA/CCA	(1)
	A_{pc}	V_{val}	V_{his} (slow)	IM	TSCA/CCA	(1)	A_{pc}	V_{val}	A_{alias} (slow)	EM	TSCA/CCA	(1)
	A_{pc}	V_{val}	V_{alias} (slow)	IM	TSCA/CCA	(1)	V_{pc}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	new
	V_{pc}	V_{val}	A_{pc} (slow)	EM	TSCA/CCA	new	V_{pc}	V_{val}	V_{pc} (slow)	IM	TSCA/CCA	new
	V_{pc}	V_{val}	A_{his} (slow)	EM	TSCA/CCA	new	V_{pc}	V_{val}	V_{his} (slow)	IM	TSCA/CCA	new
	V_{pc}	V_{val}	A_{alias} (slow)	EM	TSCA/CCA	new	V_{pc}	V_{val}	V_{alias} (slow)	IM	TSCA/CCA	new
	A_{his}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	new	A_{his}	V_{val}	A_{pc} (slow)	EM	TSCA/CCA	new
	A_{his}	V_{val}	V_{pc} (slow)	IM	TSCA/CCA	new	A_{his}	V_{val}	A_{his} (slow)	EM	TSCA/CCA	new
	A_{his}	V_{val}	V_{his} (slow)	IM	TSCA/CCA	new	A_{his}	V_{val}	A_{alias} (slow)	EM	TSCA/CCA	new
	A_{his}	V_{val}	V_{alias} (slow)	IM	TSCA/CCA	new	V_{his}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	new
	V_{his}	V_{val}	A_{pc} (slow)	EM	TSCA/CCA	new	V_{his}	V_{val}	V_{pc} (slow)	IM	TSCA/CCA	new
	V_{his}	V_{val}	A_{his} (slow)	EM	TSCA/CCA	new	V_{his}	V_{val}	V_{his} (slow)	IM	TSCA/CCA	new
	V_{his}	V_{val}	A_{alias} (slow)	EM	TSCA/CCA	new	V_{his}	V_{val}	V_{alias} (slow)	IM	TSCA/CCA	new
	A_{alias}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	(1)	A_{alias}	V_{val}	A_{pc} (slow)	EM	TSCA/CCA	(1)
	A_{alias}	V_{val}	V_{pc} (slow)	IM	TSCA/CCA	(1)	A_{alias}	V_{val}	A_{his} (slow)	EM	TSCA/CCA	(1)
	A_{alias}	V_{val}	V_{his} (slow)	IM	TSCA/CCA	(1)	A_{alias}	V_{val}	A_{alias} (slow)	EM	TSCA/CCA	(1)
	A_{alias}	V_{val}	V_{alias} (slow)	IM	TSCA/CCA	(1)	V_{alias}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	new
	V_{alias}	V_{val}	A_{pc} (slow)	EM	TSCA/CCA	new	V_{alias}	V_{val}	V_{pc} (slow)	IM	TSCA/CCA	new
	V_{alias}	V_{val}	A_{his} (slow)	EM	TSCA/CCA	new	V_{alias}	V_{val}	V_{his} (slow)	IM	TSCA/CCA	new
	V_{alias}	V_{val}	A_{alias} (slow)	EM	TSCA/CCA	new	V_{alias}	V_{val}	V_{alias} (slow)	IM	TSCA/CCA	new
	A_{pc}	V_{val}	A_{cc} (fast)	EH	TEA	(2)	V_{pc}	V_{val}	A_{cc} (fast)	EH	TEA	(2)
	A_{his}	V_{val}	A_{cc} (fast)	EH	TEA	(3)	V_{his}	V_{val}	A_{cc} (fast)	EH	TEA	new
	A_{alias}	V_{val}	A_{cc} (fast)	EH	TEA	(2)	V_{alias}	V_{val}	A_{cc} (fast)	EH	TEA	(2)
BTB (call/ret)	A_{inv}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	(1)	V_{inv}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	new
	V_{val}	A_{pc}	V_{val} (slow)	EM	TSCA/CCA	new	V_{val}	V_{pc}	V_{val} (slow)	IM	TSCA/CCA	new
	V_{val}	A_{alias}	V_{val} (slow)	EM	TSCA/CCA	new	V_{val}	V_{alias}	V_{val} (slow)	IM	TSCA/CCA	new
	A_{pc}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	(1)	A_{pc}	V_{val}	A_{pc} (slow)	EM	TSCA/CCA	(1)
	A_{pc}	V_{val}	V_{pc} (slow)	IM	TSCA/CCA	(1)	A_{pc}	V_{val}	A_{alias} (slow)	EM	TSCA/CCA	(1)
	A_{pc}	V_{val}	V_{alias} (slow)	IM	TSCA/CCA	(1)	V_{pc}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	new
	V_{pc}	V_{val}	A_{pc} (slow)	EM	TSCA/CCA	new	V_{pc}	V_{val}	V_{pc} (slow)	IM	TSCA/CCA	new
	V_{pc}	V_{val}	A_{alias} (slow)	EM	TSCA/CCA	new	V_{pc}	V_{val}	V_{alias} (slow)	IM	TSCA/CCA	new
	A_{alias}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	(1)	A_{alias}	V_{val}	A_{pc} (slow)	EM	TSCA/CCA	(1)
	A_{alias}	V_{val}	V_{pc} (slow)	IM	TSCA/CCA	(1)	A_{alias}	V_{val}	A_{alias} (slow)	EM	TSCA/CCA	(1)
	A_{alias}	V_{val}	V_{alias} (slow)	IM	TSCA/CCA	(1)	V_{alias}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	new
	V_{alias}	V_{val}	A_{pc} (slow)	EM	TSCA/CCA	new	V_{alias}	V_{val}	V_{pc} (slow)	IM	TSCA/CCA	new
	V_{alias}	V_{val}	A_{alias} (slow)	EM	TSCA/CCA	new	V_{alias}	V_{val}	V_{alias} (slow)	IM	TSCA/CCA	new
	A_{pc}	V_{val}	A_{cc} (fast)	EH	TEA	(2)	V_{pc}	V_{val}	A_{cc} (fast)	EH	TEA	(2)
	A_{alias}	V_{val}	A_{cc} (fast)	EH	TEA	(2)	V_{alias}	V_{val}	A_{cc} (fast)	EH	TEA	(2)
RSB	A_{inv}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	(1)	V_{inv}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	new
	V_{val}	A_{alias}	V_{val} (slow)	EM	TSCA/CCA	new	V_{val}	V_{alias}	V_{val} (slow)	IM	TSCA/CCA	new
	A_{alias}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	(1)	A_{alias}	V_{val}	A_{alias} (slow)	EM	TSCA/CCA	(1)
	A_{alias}	V_{val}	V_{alias} (slow)	IM	TSCA/CCA	(1)	V_{alias}	V_{val}	V_{val} (fast)	IH	TSCA/CCA	new
	V_{alias}	V_{val}	A_{alias} (slow)	EM	TSCA/CCA	new	V_{alias}	V_{val}	V_{alias} (slow)	IM	TSCA/CCA	new
	A_{alias}	V_{val}	A_{cc} (fast)	EH	TEA	(2)	V_{alias}	V_{val}	A_{cc} (fast)	EH	TEA	new

- PHT: (1) BranchScope [27]; (2) Bluethunder [38]; (3) Spectre V1 [40]; (4) BranchSpectre [18].
- BTB: (1) Predicting Keys [1]–[3]; (2) Spectre V2 [40]; (3) BHI [5].
- RSB: (1) Predicting Keys [13]; (2) Spectre V5 [42], [45].

Framework: Extensibility of Our Modeling

- **Case study 1: modeling of TAGE branch predictor**
  - TAGE is widely deployed in popular open-source processors
    - e.g., XiangShan

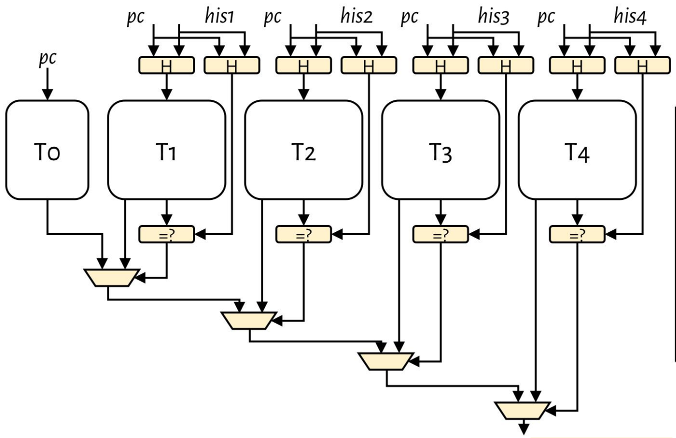


25 Operations
7 States
106 Attacks

  - Modeling T1~T4 as independent units
    - 16 extra operations and 4 extra states

9 Operations
3 States
28 Attacks

extension



TAGE	Op	State	TAGE	Op	State
T1	A_{pc1}	mispredict1	T3	A_{pc3}	mispredict3
	V_{pc1}	mispredict1		V_{pc3}	mispredict3
	A_{his1}	mispredict1		A_{his3}	mispredict3
	V_{his1}	mispredict1		V_{his3}	mispredict3
T2	A_{pc2}	mispredict2	T4	A_{pc4}	mispredict4
	V_{pc2}	mispredict2		V_{pc4}	mispredict4
	A_{his2}	mispredict2		A_{his4}	mispredict4
	V_{his2}	mispredict2		V_{his4}	mispredict4

Baseline

TAGE

Framework: Viability of Novel Attacks

武漢大學 WUHAN UNIVERSITY

- **Case study 2: evaluation of two novel PHT attacks**
  - A V_{pc}-based attack variant and a V_{his}-based attack variant
  - Transmission of random "0" and "1" bits repeated 1,000,000 times
  - Leakage of sensitive information with a substantial bandwidth on Intel processors

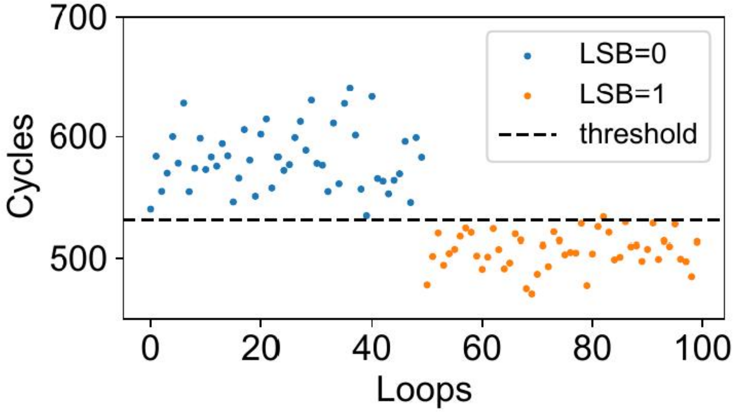
Number	Attack Pattern	Processor	Timing Resolution	Capacity
#10	$V_{pc} \rightarrow V_{val} \rightarrow V_{val}$	Intel Core i5-1135G7	92 vs 108 cycles	865.7 Kbps
		Intel Core i7-12700	69 vs 83 cycles	925.5 Kbps
#20	$V_{his} \rightarrow V_{val} \rightarrow V_{val}$	Intel Core i5-1135G7	91 vs 114 cycles	690.7 Kbps
		Intel Core i7-12700	67 vs 83 cycles	734.1 Kbps

Framework: Practicality of Novel Attacks

- **Case study 3: recovery of LSB in OpenSSL with a novel BTB variant**
  - *EVP_EncryptUpdate()* in *libcrypto* library of OpenSSL 1.1.1b is vulnerable (CCS'19)
  - We demonstrate the practicality of a novel variant exploiting the same vulnerability
  - We implement the PoC of #31 ($V_{val} \rightarrow A_{pc} \rightarrow V_{val}$) to recover the LSB of the first bytes

mislead to '1' | SLOW (LSB=0) | FAST (LSB=1)

The attacker observes the **threshold timing** of V_{val} based on branch hit and miss	The attacker conducts the branch target injection with A_{pc} to **mislead** the indirect branch in the *libcrypto*	The attacker trigger the V_{val} and measures the **execution timing** to infer the target LSB
Step 1: V_{val} →	Step 2: A_{pc} →	Step 3: V_{val}



Recovering LSB in OpenSSL on Intel Core i7-12700

Analysis: Modeling Typical Secure Designs

WUHAN UNIVERSITY

- ➢ **Our framework is applicable to evaluating secure designs (as instances)**
- ➢ **We model 8 secure branch predictors and 4 secure speculation schemes**

Secure BP	Remaining Ops	Reference
Lock-Based BTB	25/53	TrustCom 2014
MI6	33/53	MICRO 2019
BRB	33/53	HPCA 2019
Two-Level Encryption	22/53	TACO 2020
Noisy-XOR-BP	22/53	DAC 2021
PSC	31/53	JCST 2021
LS-BP	22/53	ASP-DAC 2022
HyBP	16/53	HPCA 2022

Secure Speculation	Blocked Ops	Reference
DAWG	A_{cc} for cache (different domains)	MICRO 2018
CSF-LFENCE	V_{val} for PHT	ASPLOS 2019
STT	V_{val} for PHT	MICRO 2019
InvisiSpec	A_{cc} for cache	MICRO 2018

We select **four representative hardware-based defenses** against speculative attacks that introduce low-performance overhead

We conduct a comprehensive analysis of **remaining operations** in our model for each secure branch predictor

We perform a thorough analysis of **blocked operations** for each secure speculation scheme

Analysis: Overview of Secure BP Evaluation

武漢大學 WUHAN UNIVERSITY

- **Secure branch predictor evaluation for all 156 three-step attacks**
  - PSC and HyBP are the most effective secure branch predictors for mitigating PHT and BTB security vulnerabilities under ideal circumstances
  - The best-performing HyBP can shield about 79% of the attack patterns
  - The worst-performing MI6 and BRB can only cover about 16% of the attack patterns

Secure BP	PHT	BTB (ind)	BTB (call)	BTB (ret)	RSB	Total
Lock-Based BTB	28/28	19/56	11/30	11/30	5/12	74/156
MI6	10/28	56/56	30/30	30/30	5/12	131/156
BRB	10/28	56/56	30/30	30/30	5/12	131/156
Two-Level Encryption	18/28	12/56	2/30	2/30	5/12	39/156
Noisy-XOR-BP	18/28	12/56	2/30	2/30	5/12	39/156
PSC (ideal)	0/28	56/56	30/30	30/30	5/12	121/156
LS-BP	18/28	12/56	2/30	2/30	5/12	39/156
HyBP	18/28	10/56	0/30	0/30	5/12	33/156

Analysis: Evaluation for Known/New Attacks

武漢大學 WUHAN UNIVERSITY

- **Secure branch predictor evaluation for known/new attacks**
  - HyBP provides the best protection against known and newly derived attacks
  - Two-Level Encryption, Noisy-XOR-BP, and LS-BP have better protection coverage
  - Lock-Based BTB has significant omissions for newly derived attacks
  - MI6 and BRB do not adequately protect against known and newly derived attacks

Secure BP	PHT (known)	BTB (known)	RSB (known)	PHT (new)	BTB (new)	RSB (new)
Lock-Based BTB	12/12	6/50	0/5	16/16	35/66	5/7
MI6	2/12	50/50	0/5	8/16	66/66	5/7
BRB	2/12	50/50	0/5	8/16	66/66	5/7
Two-Level Encryption	5/12	7/50	0/5	9/16	35/66	5/7
Noisy-XOR-BP	5/12	7/50	0/5	9/16	35/66	5/7
PSC (ideal)	0/12	50/50	0/5	0/16	66/66	5/7
LS-BP	5/12	7/50	0/5	9/16	35/66	5/7
HyBP	5/12	4/50	0/5	13/16	6/66	5/7

Analysis: Secure BPs vs Secure Speculation

- **Evaluation of secure BPs and HW defenses against speculative attacks**
  - Hardware-based secure speculation can only mitigate a limited number of speculative execution attacks or only mitigate specific cache covert channels
  - Secure branch predictor designs can mitigate more speculative execution attacks

Defense Strategy	Speculative Attacks (cache channel)	Speculative Attacks (other channel)	Defense Strategy	Speculative Attacks (cache channel)	Speculative Attacks (other channel)
Lock-Based BTB	12/20	12/20	MI6	17/20	17/20
BRB	17/20	17/20	Two-Level Encryption	6/20	6/20
Noisy-XOR-BP	6/20	6/20	PSC (ideal)	15/20	15/20
LS-BP	6/20	6/20	HyBP	6/20	6/20
DAWG	17/20	19/20	CSF-LFENCE	15/20	15/20
STT	15/20	15/20	InvisiSpec	15/20	19/20

Analysis: Secure BPs vs Secure Speculation

- **Evaluation of secure BPs and HW defenses against speculative attacks**
  - Hardware-based secure speculation can only mitigate a limited number of speculative execution attacks or only mitigate specific cache covert channels
  - Secure branch predictor designs can mitigate more speculative execution attacks

Secure branch predictor designs are promising solutions in mitigating branch predictor security vulnerabilities and preserving the confidentiality and integrity of computer systems!

BRB	17/20	17/20	Two-Level Encryption	6/20	6/20
Noisy-XOR-BP	6/20	6/20	PSC (ideal)	15/20	15/20
LS-BP	6/20	6/20	HyBP	6/20	6/20
DAWG	17/20	19/20	CSF-LFENCE	15/20	15/20
STT	15/20	15/20	InvisiSpec	15/20	19/20

Conclusion

WUHAN UNIVERSITY

- **Modeling: propose a three-step branch predictor modeling methodology**

We propose a three-step modeling approach for evaluating the security properties of branch predictors at the microarchitecture design stage. Our technique abstractly characterizes 19 branch predictor states and 53 operations of the attacker and victim that could affect these states.

- **Framework: derive 156 effective attack patterns with 89 novel variants**

We develop a comprehensive and automated evaluation framework based on the proposed model that leverages symbolic execution to analyze all potential three-step combinations, yielding 156 valid attack patterns against branch predictors, with 89 novel attacks never discovered.

- **Analysis: conduct security analysis of exisiting HW-based secure designs**

We apply our security analysis framework to 8 existing secure branch predictor designs and four typical hardware alleviations against speculative execution attacks, and the results show that secure branch predictors are promising solutions for enhancing the security of the computer system.

Thanks

- **Artifact**
  - Archival: https://doi.org/10.5281/zenodo.10297402
  - Latest: https://github.com/iamywang/bp-security-framework
- **Contact**

http://csccl.whu.edu.cn

CSCCL (CryptoChip Lab) at Wuhan University

wangquancheng@whu.edu.cn