Mitigation of Cache Side Channel Attacks with Answer Set Programming

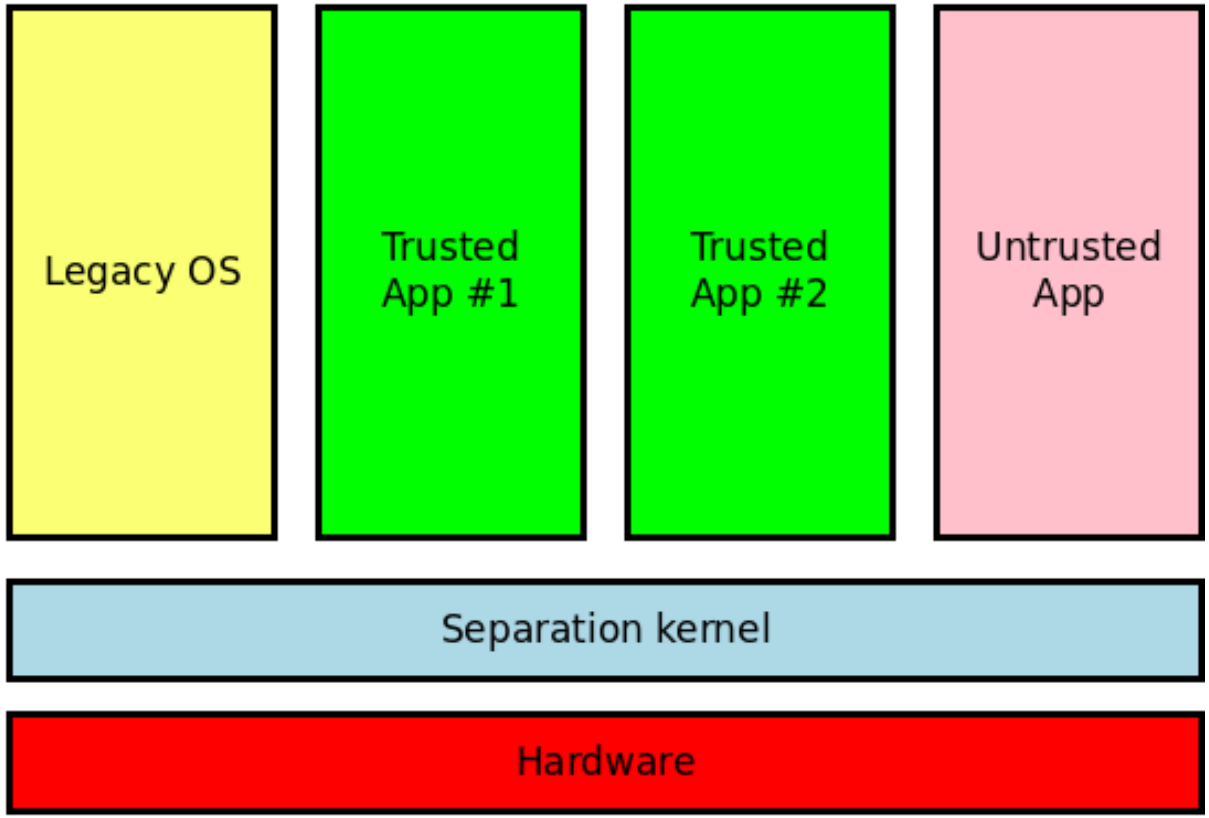
Marl Joos[1,2] Tobias Philipp[2]

[1] *Technische Universität Berlin, Germany, m.joos@campus.tu-berlin.de*

[2] *SINA Development & Verification Team, Division Defence & Space, secunet Security Networks AG, Essen, Germany, tobias.philipp@secunet.com*

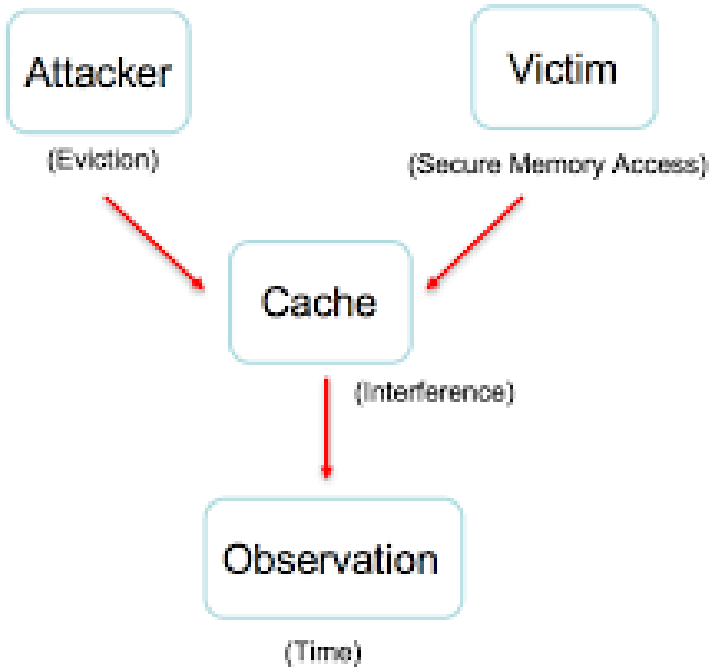
Motivation (1/2): High security project

- Research project: Multilevel Workstation
  - Software and hardware system to separate multiple operating systems
- Software platform: Separation kernel
- Focus: Confidentiality and prevention of information leakage



Motivation (2/2): Cache-based side channel attacks (2/2)

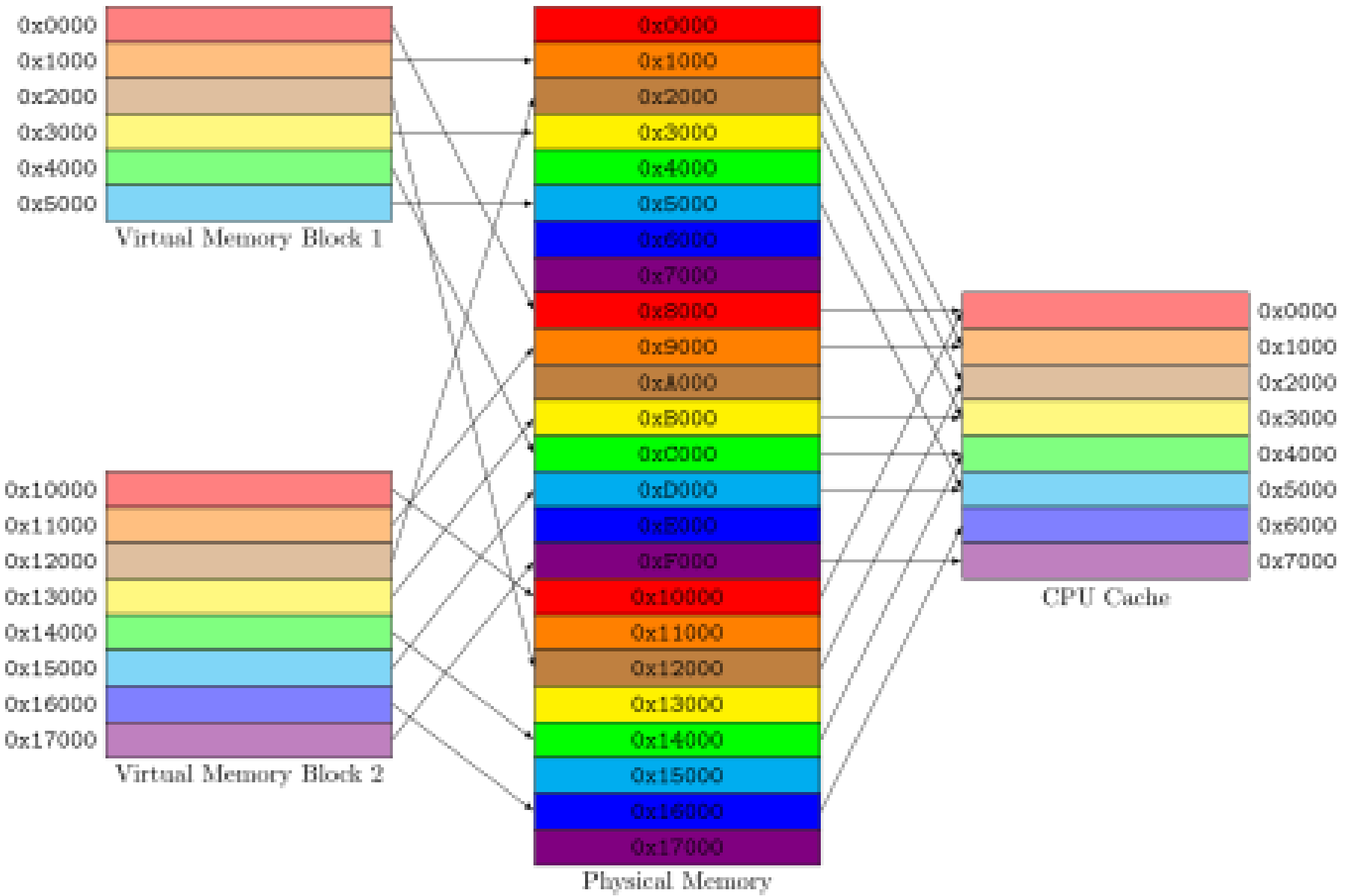
- Uses shared caches of modern processors to leak confidential information
- Recent research showed practicability and relevance of attacks
  - Hu (1992), Kelsey (1998), Bernstein (2005), Osvik (2010), Yarom (2013), ...



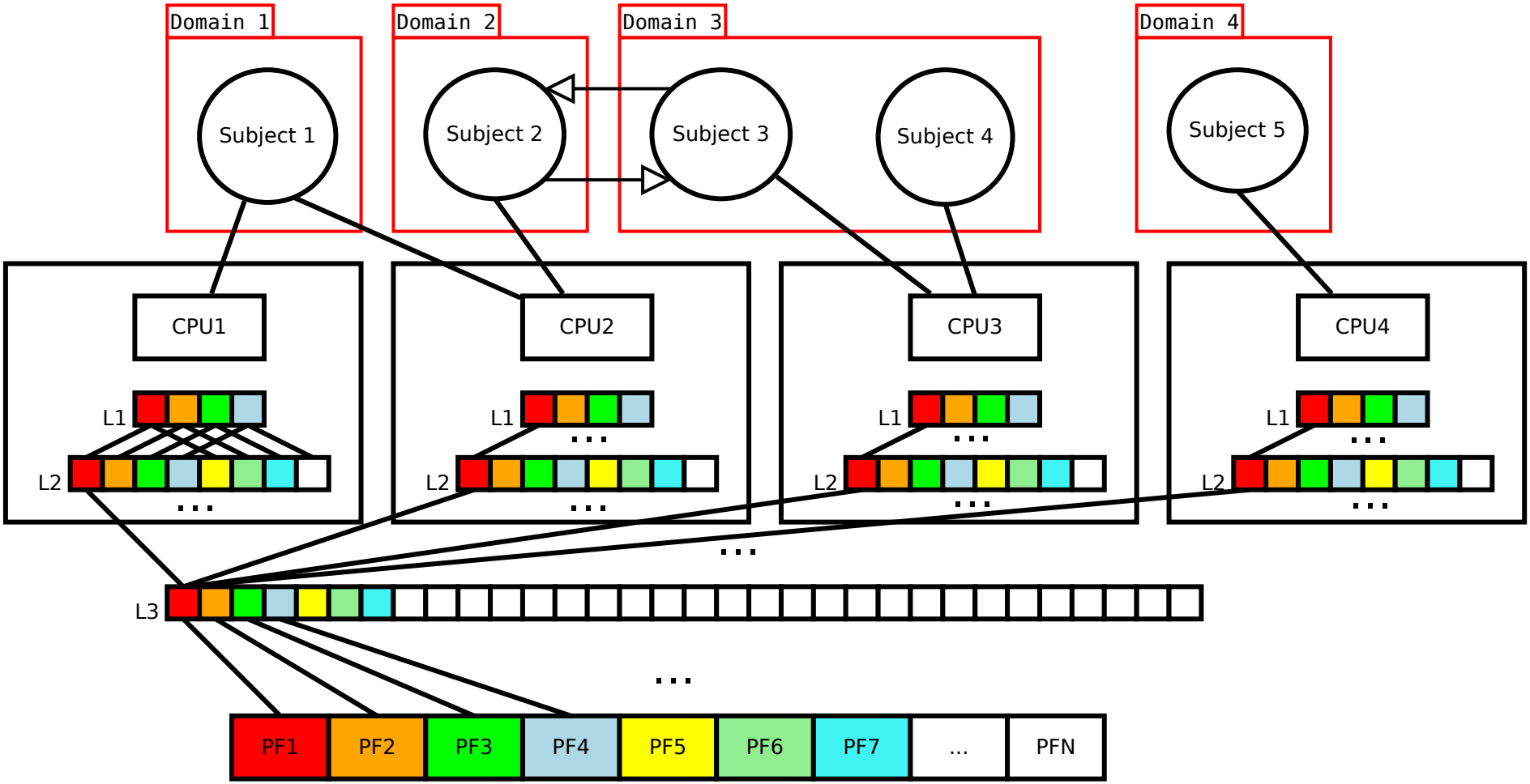
Outline

- Page coloring problem
- Theoretical framework
- ASP-based PCP solver
- Experiment results
- Conclusions

Page Coloring



Page Coloring Problem (PCP) 1/2



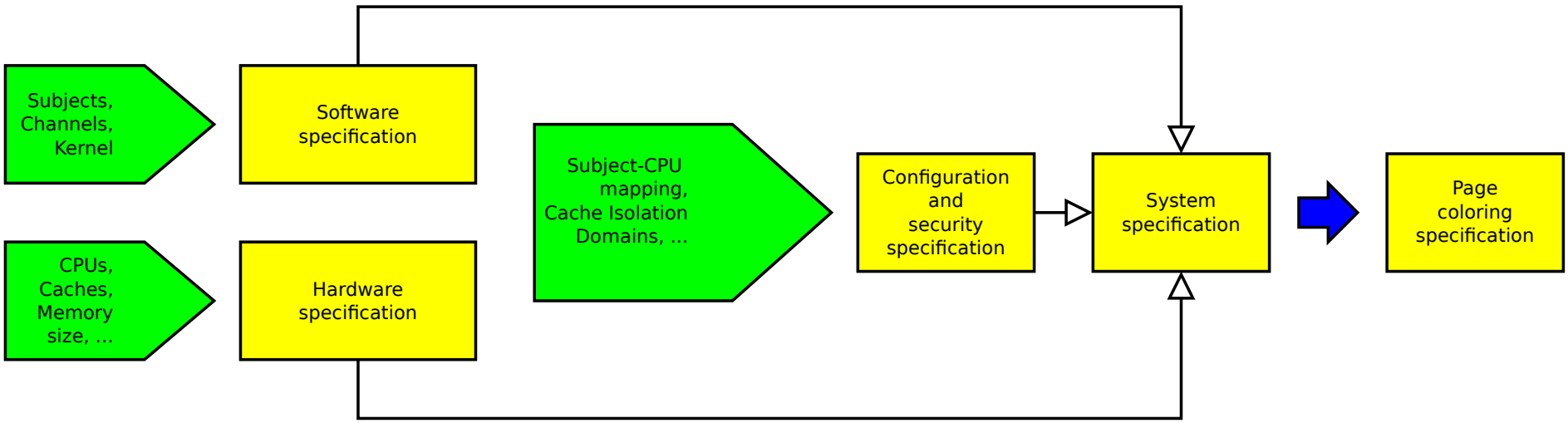
Page Coloring Problem (PCP) 2/2

Given:

- Subjects
- Channels
- Hardware system (CPU cores, Cache organization, ...)
- Security requirements ("subject X may not interfere with subject Y")
- Other constraints (minimum/maximum colors of a specific subject)

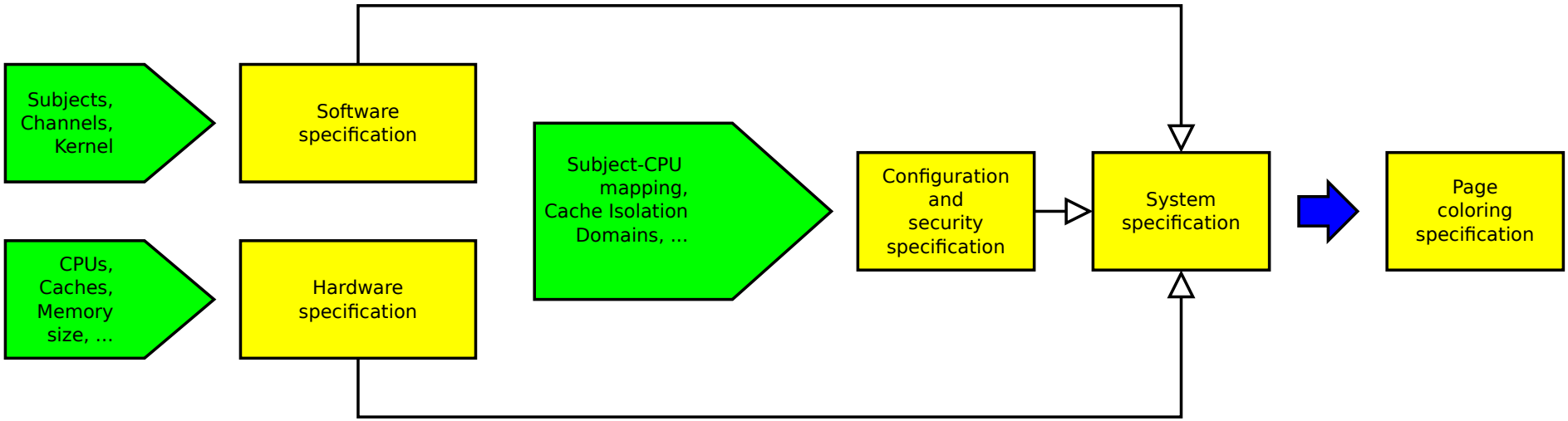
Problem: How to distribute memory page frames to the applications so that there is only cache interference as specified by the security policy?

Theoretical Framework (1/4)



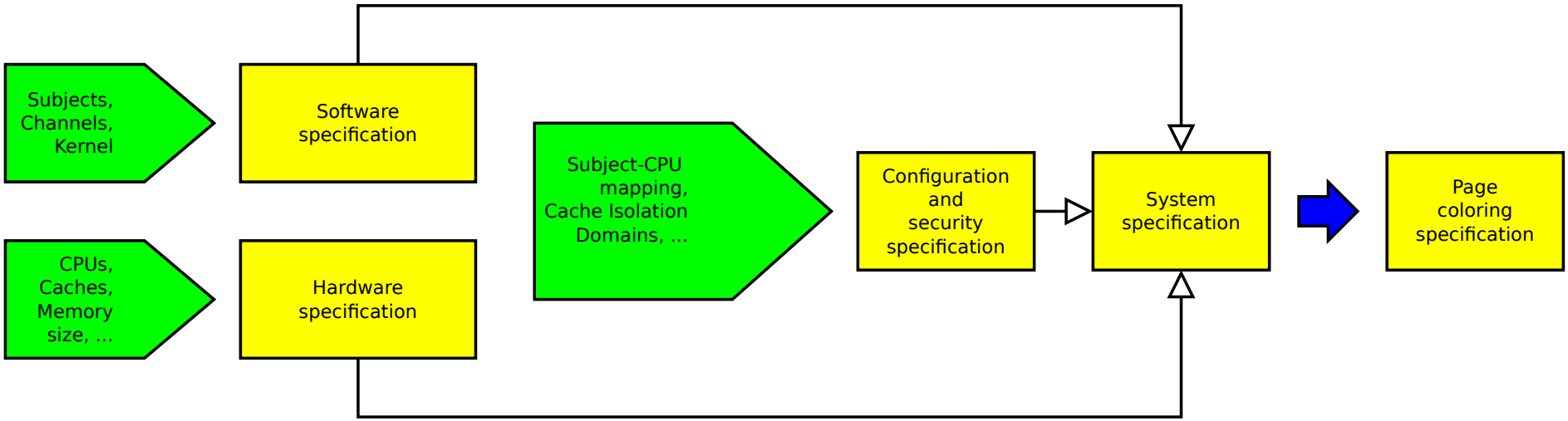
- Software specification: Information about subjects, channel, kernel such as memory size

Theoretical Framework (2/4)



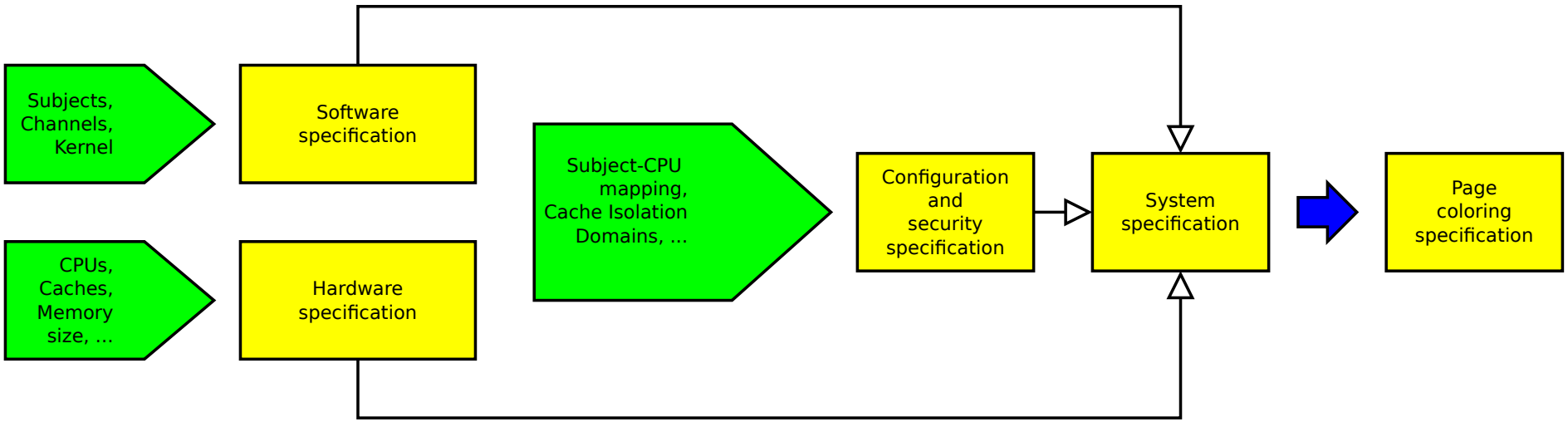
- Hardware specification: CPU cores, cache organization, main memory size, ...

Theoretical Framework (3/4)



- Configuration and security specification
  - Uses information of software spec. and hardware spec.
  - Contains security requirements on cache interference

Theoretical Framework (4/4)



- System specification: Combines software spec., hardware spec. and configuration and security spec.
- Page coloring specification: Contains all information to solve a page coloring problem instance

ASP-based PCP solver

Encoded the problem with 13 predicates such as:

Predicate	**Description**
`mo(M)`	Define a memory object (subject, channel)
`cache_color(C,CCI)`	Define cache color
`cache_cpu(C,CPU)`	Define cache cpu mapping (cache org.)
`mo_cc(M, CC)`	Map memory object to cache color
...	...

ASP-based PCP solver

Rule syntax:

head :- body.

Four rules:

1. Assign page colors to memory objects based on the minimum and maximum page colors requirements.
2. Assign cache colors to memory objects based on the CPU cores assigned to the memory objects.
3. Avoid cache interference of memory objects of different cache isolation domains.
4. Assign as much cache colors as possible.

ASP rules

One example ASP rule:

- `MIN { mo_pc(M, pc(PC)) : pc(PC) } MAX :-`
  `mo(M) , min_pcs(M, MIN) , max_pcs(M, MAX ).`

“Assign at least MIN to MAX page colors to memory object M.“

Experiments and results

1. 128 generated PCP instances with up to 1000 memory objects
2. 1 PCP instance derived from MLW project
3. Solved within reasonable time and memory limit (standard desktop PC, single core, 16 GB main memory)
    1. less than 10 min for randomly generated page coloring instances
    2. about 20 seconds for industrial page coloring instance

Conclusions

- Cache-based side channels are very important and hard to mitigate
- Developed a theoretical framework to describe the problem
- ASP is effective and useful for industrial use cases
- Implementation contains less than 100 lines of code

Mitigation of Cache Side Channel Attacks with Answer Set Programming

Marl Joos[1,2] Tobias Philipp[2]

[1] *Technische Universität Berlin, Germany, m.joos@campus.tu-berlin.de*

[2] *SINA Development & Verification Team, Division Defence & Space, secunet Security Networks AG, Essen, Germany, tobias.philipp@secunet.com*

Image sources

- Cache-based side channel attacks: `http://palms.princeton.edu/system/files/Micro-camera-ready-final.pdf`
- Page coloring: `https://en.wikipedia.org/wiki/Cache_coloring#/media/\protect\@normalcr\relaxFile:Page_Cache_Coloring.svg`