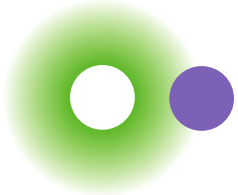


NICTA

The Last Mile
An Empirical Study of
Timing Channels on seL4

David Cock Qian Ge Toby Murray Gernot Heiser

4 November 2014

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

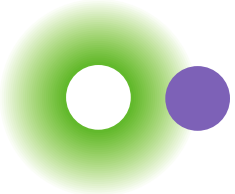
Australian Government
Department of Broadband, Communications and the Digital Economy
Australian Research Council

NICTA Funding and Supporting Members and Partners
Australian National University
UNSW THE UNIVERSITY OF NEW SOUTH WALES
NSW GOVERNMENT Trade & Investment
Victoria The Place To Be
THE UNIVERSITY OF SYDNEY
Queensland Government
Griffith UNIVERSITY
QUT
THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Outline

- Background
  - seL4
  - Channels
  - Experimental Approach
- Local Channels
  - The Cache Channel
  - Instruction-Based Scheduling
  - Cache Colouring
  - New Channels
- Remote Channels
  - Scheduled Delivery
- Summary
  - Outcomes
  - Ongoing Work

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work



NICTA

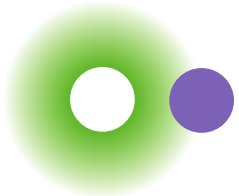
seL4

seL4 is a verified, high-performance microkernel. We have:

- Proof of **functional correctness**.
- Proof of **authority confinement**.
- Proof of **explicit information-flow control**.
- **WCET** analysis.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

seL4



seL4 is a verified, high-performance microkernel. We have:

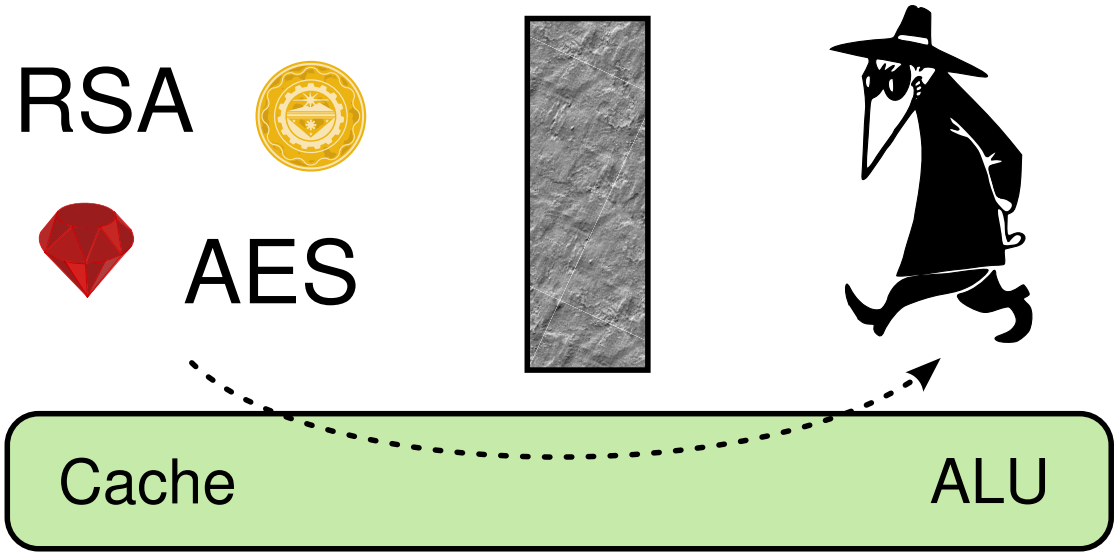
- Proof of **functional correctness**.
- Proof of **authority confinement**.
- Proof of **explicit information-flow control**.
- **WCET** analysis.

We don't have:

- An comprehensive hardware model.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Covert and Side Channels



Unexpected channels invalidate info-flow control.
We can't prove their absence:

- Depend heavily on undocumented chip internals.
- Channels are probabilistic.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Channels of Interest

Channels we consider

- Can subvert our proof: *Timing channels*.
- Could be fixed with OS techniques e.g. *Cache contention*.
- That can be exploited in software.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Channels of Interest

Channels we consider

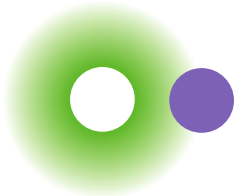
- Can subvert our proof: *Timing channels*.
- Could be fixed with OS techniques e.g. *Cache contention*.
- That can be exploited in software.

Channels we **don't** consider

- Physical attacks e.g. *DPA*.
- Channels already excluded by proof e.g. *Storage channels*.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

An Experimental Corpus



We exploit 2 principal **hardware** channels:

- The L2 cache channel.
- The bus contention channel (not covered today).

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

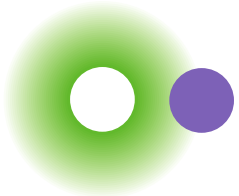
An Experimental Corpus

We exploit 2 principal **hardware** channels:

- The L2 cache channel.
- The bus contention channel (not covered today).
- The data also suggests 3 more (2 as-yet-unrecognised).

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

An Experimental Corpus



We exploit 2 principal **hardware** channels:

- The L2 cache channel.
- The bus contention channel (not covered today).
- The data also suggests 3 more (2 as-yet-unrecognised).

We evaluate 2 cache-channel countermeasures:

- Instruction-based scheduling.
- Cache colouring.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

An Experimental Corpus

We exploit 2 principal **hardware** channels:

- The L2 cache channel.
- The bus contention channel (not covered today).
- The data also suggests 3 more (2 as-yet-unrecognised).

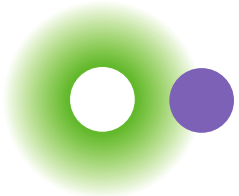
We evaluate 2 cache-channel countermeasures:

- Instruction-based scheduling.
- Cache colouring.

We also consider countermeasures against **remote, algorithmic** channels:

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

An Experimental Corpus



We exploit 2 principal **hardware** channels:

- The L2 cache channel.
- The bus contention channel (not covered today).
- The data also suggests 3 more (2 as-yet-unrecognised).

We evaluate 2 cache-channel countermeasures:

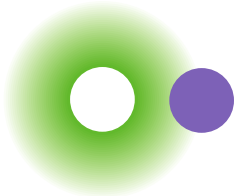
- Instruction-based scheduling.
- Cache colouring.

We also consider countermeasures against **remote, algorithmic** channels:

- Lucky-13 against OpenSSL is our example victim.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

An Experimental Corpus



We exploit 2 principal **hardware** channels:

- The L2 cache channel.
- The bus contention channel (not covered today).
- The data also suggests 3 more (2 as-yet-unrecognised).

We evaluate 2 cache-channel countermeasures:

- Instruction-based scheduling.
- Cache colouring.

We also consider countermeasures against **remote, algorithmic** channels:

- Lucky-13 against OpenSSL is our example victim.
- Scheduled delivery is our countermeasure.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

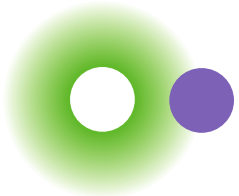
Experimental Platforms

	Core	Date	L2 Cache
iMX.31	ARM1136JF-S (*ARMv6*)	2005	128 KiB
E6550	Conroe (*x86-64*)	2007	4096 KiB
DM3730	Cortex A8 (*ARMv7*)	2010	256 KiB
AM3358	Cortex A8 (*ARMv7*)	2011	256 KiB
iMX.6	Cortex A9 (*ARMv7*)	2011	1024 KiB
Exynos4412	Cortex A9 (*ARMv7*)	2012	1024 KiB

- 7 years and 3 (ARM) core generations.
- 32-fold range of cache sizes.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

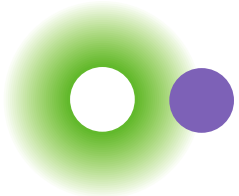
Methodology



- Integrated with nightly regression test.
- Runs each channel with each countermeasure (54 combinations).
- 2,000 hours of data over 12 months, 4.3 GiB.
- Data and analysis tools open source: `http://ssrg.nicta.com.au/projects/TS/timingchannels.pml`

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

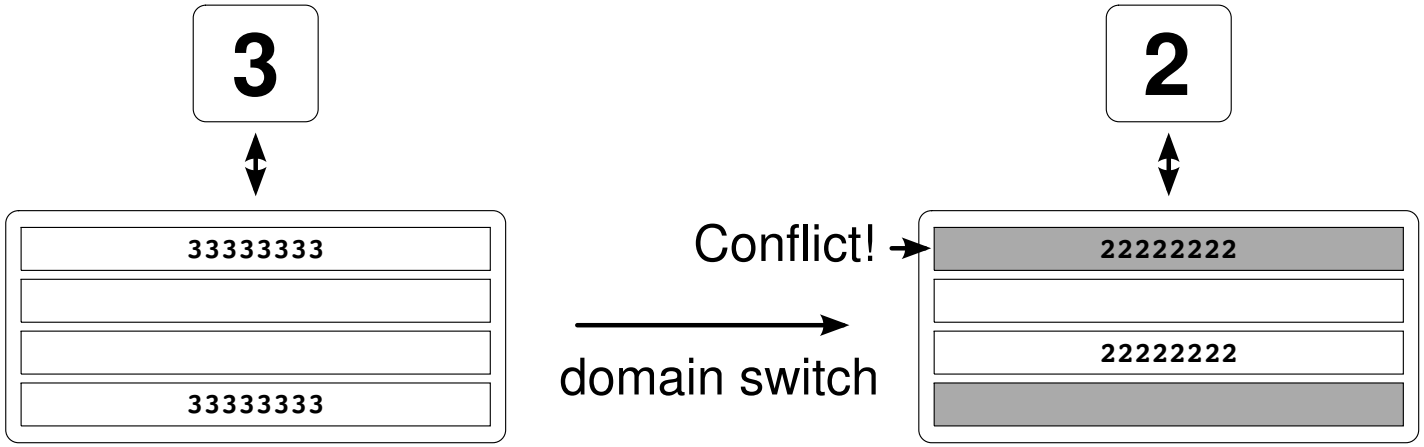
Outline



- Background
  - seL4
  - Channels
  - Experimental Approach
- Local Channels
  - The Cache Channel
  - Instruction-Based Scheduling
  - Cache Colouring
  - New Channels
- Remote Channels
  - Scheduled Delivery
- Summary
  - Outcomes
  - Ongoing Work

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

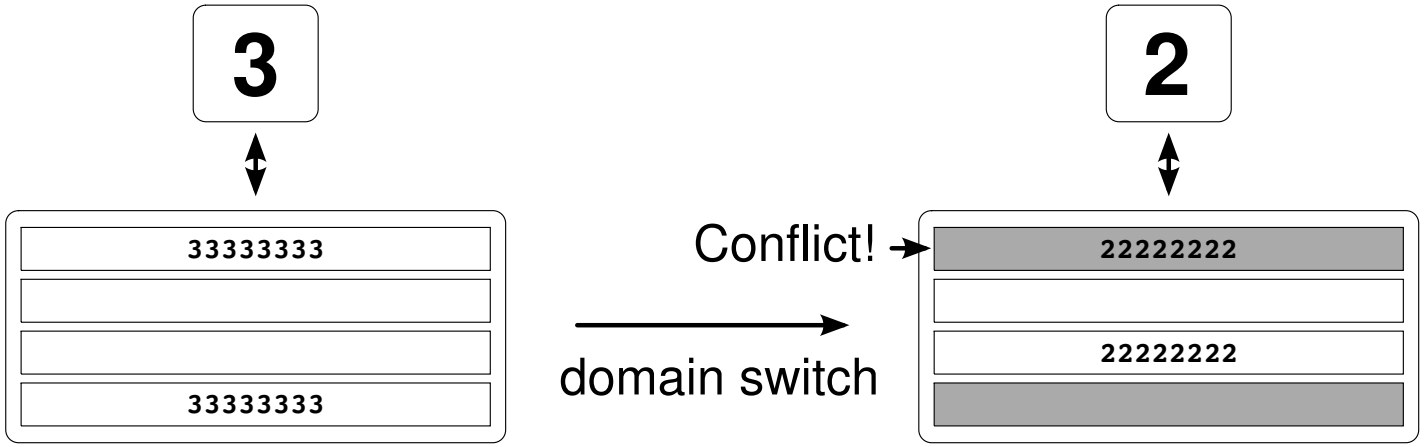
Cache Contention



- If you share a core, you share a cache.
- Hit vs. miss makes a **big** time difference.
- Many published attacks steal keys through the cache.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

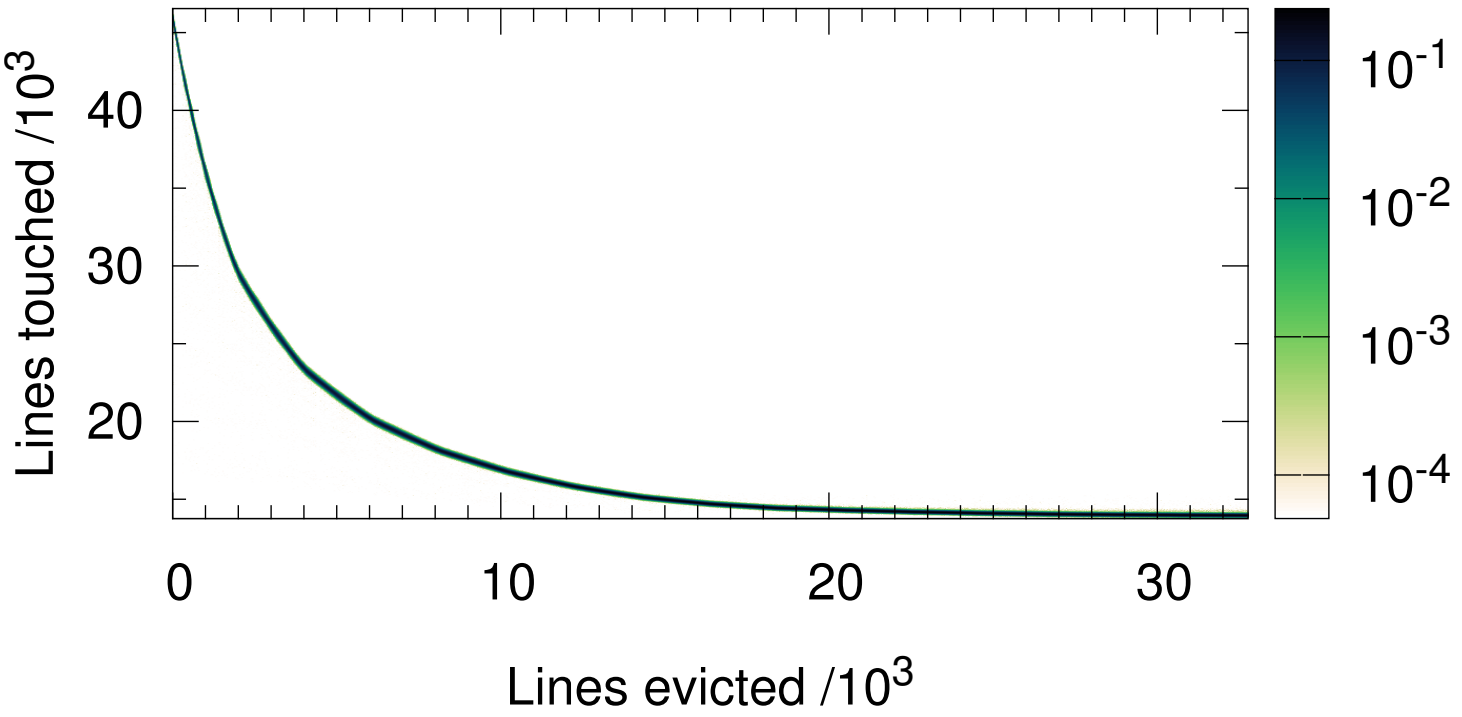
Cache Contention



- If you share a core, you share a cache.
- Hit vs. miss makes a **big** time difference.
- Many published attacks steal keys through the cache.
- We can control cache allocation (more shortly).

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Exynos4412 Cache Channel



- 32,768 cache lines, 1000Hz sample rate (preemption).
- Bandwidth: 2400b/s.
- Baseline for comparison.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

What is Instruction-Based Scheduling (IBS)?

Our example attack used preemption as a clock. If it's tied to progress, the channel should vanish.
This is a form of **deterministic execution**.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

What is Instruction-Based Scheduling (IBS)?

Our example attack used preemption as a clock. If it's tied to progress, the channel should vanish.
This is a form of **deterministic execution**.

Advantages

- Applies to any channel.
- Simple to implement (18 lines in seL4).

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

What is Instruction-Based Scheduling (IBS)?

Our example attack used preemption as a clock. If it's tied to progress, the channel should vanish.
This is a form of **deterministic execution**.

Advantages

- Applies to any channel.
- Simple to implement (18 lines in seL4).

Disadvantages

- Restrictive — Need to remove **all** clocks.
- Performance counter accuracy critical.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

What is Instruction-Based Scheduling (IBS)?

Our example attack used preemption as a clock. If it's tied to progress, the channel should vanish.
This is a form of **deterministic execution**.

Advantages

- Applies to any channel.
- Simple to implement (18 lines in seL4).

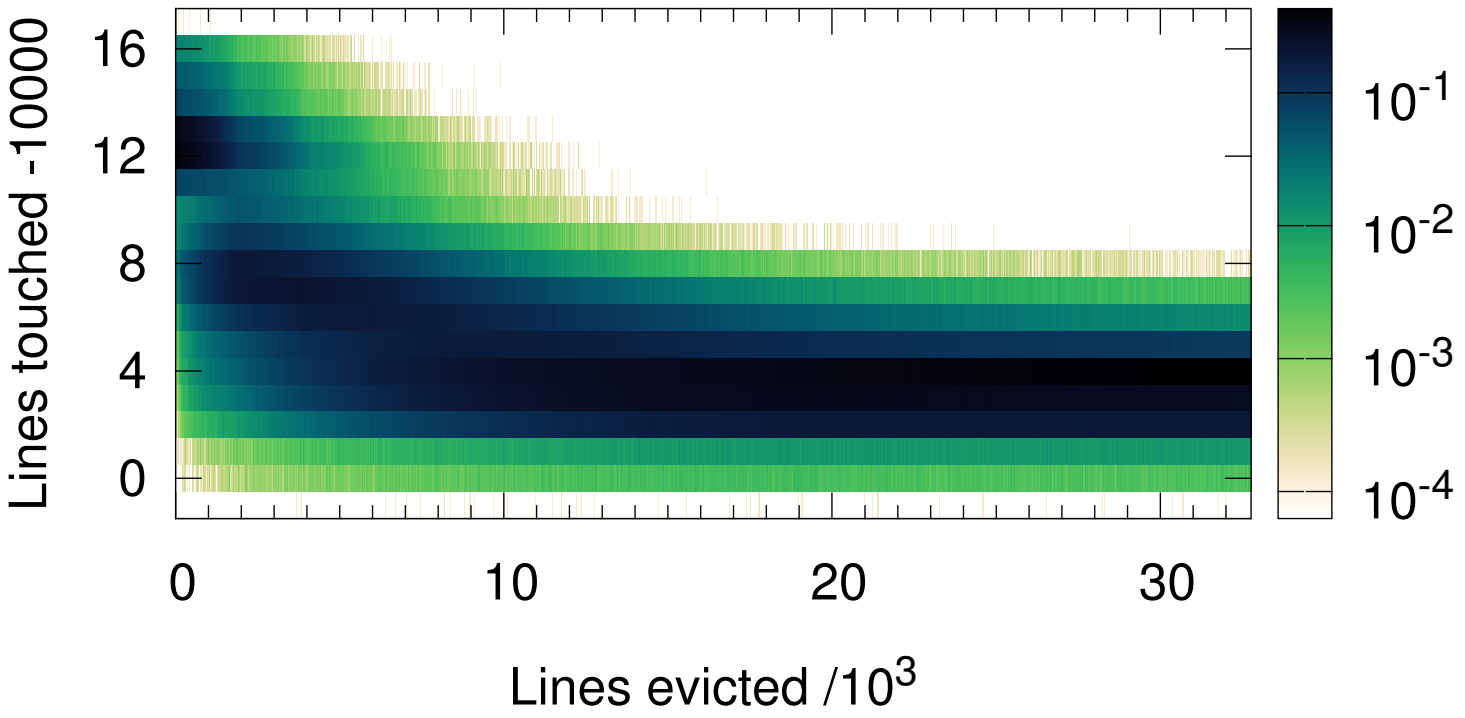
Disadvantages

- Restrictive — Need to remove **all** clocks.
- Performance counter accuracy critical.

Works great on older chips (iMX.31), but. . .

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

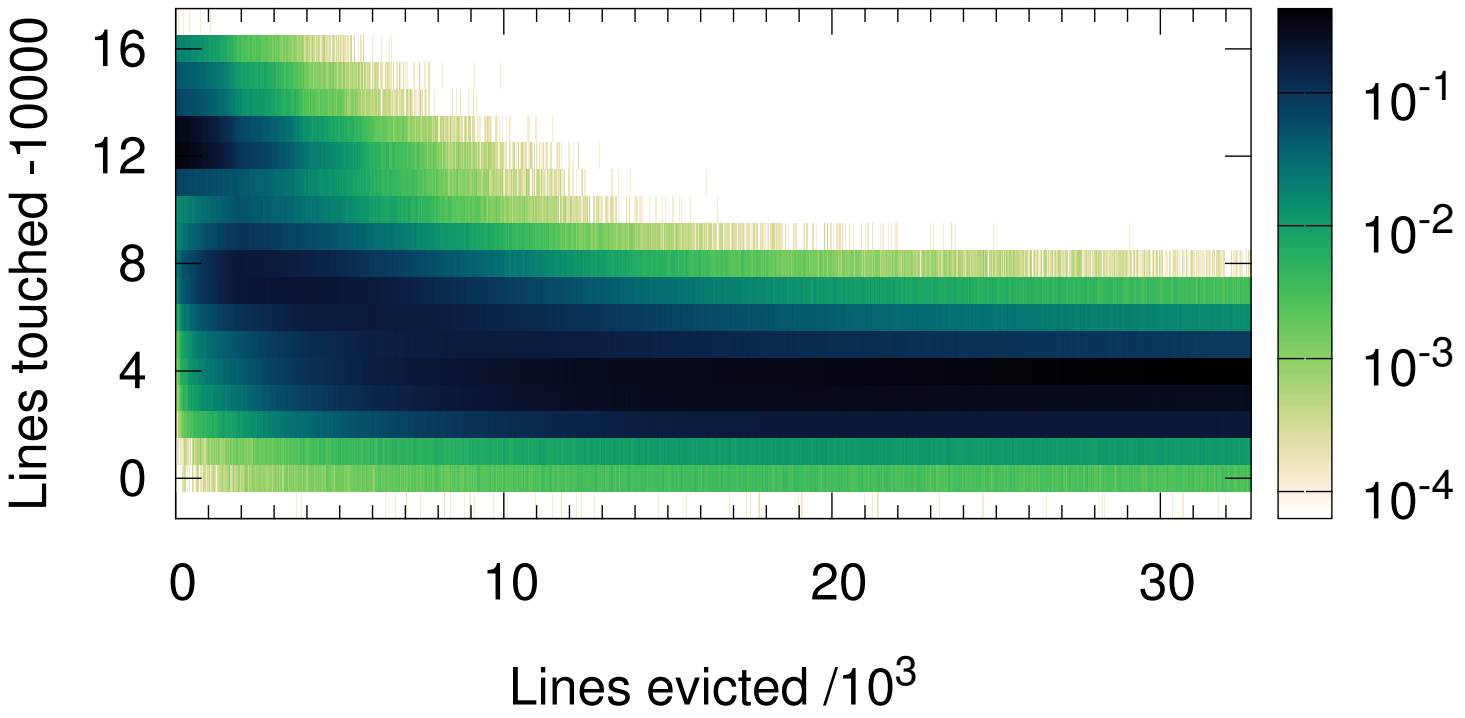
Exynos4412 Cache Channel with IBS



- Preempt after 10^5 instructions. Bandwidth: 400b/s.
- $6\times$ reduction — very poor result.
- Event delivery is imprecise thanks to speculation.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

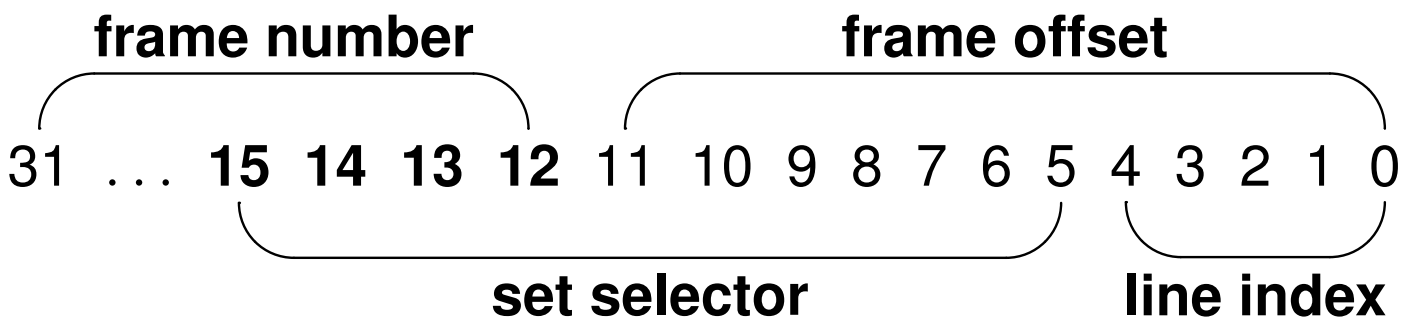
Exynos4412 Cache Channel with IBS



- Preempt after 10^5 instructions. Bandwidth: 400b/s.
- 6× reduction — very poor result.
- Event delivery is imprecise thanks to speculation.
- Attacker can modulate it!

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

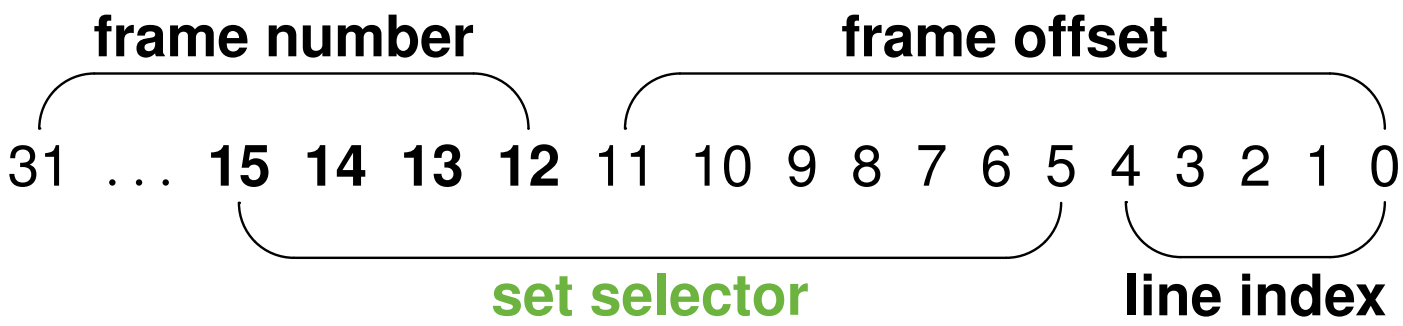
What is Colouring?



- Caches are divided into **sets** by **physical address**.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

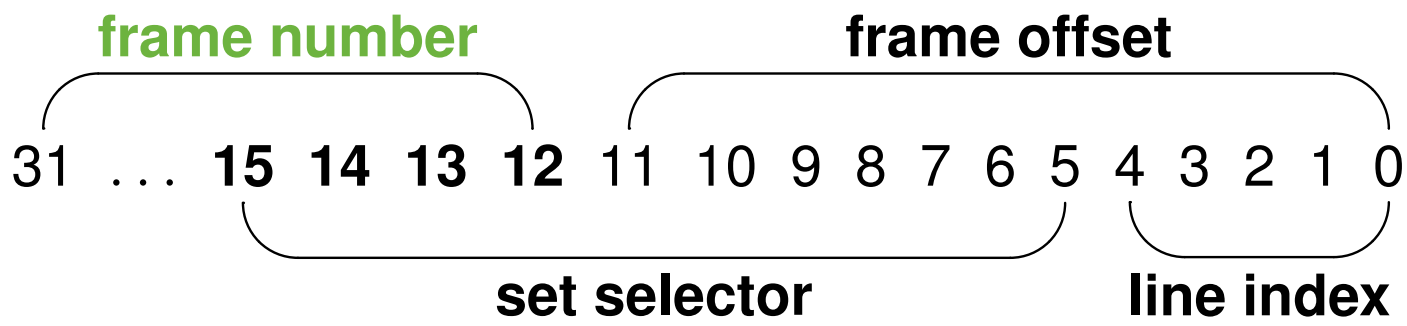
What is Colouring?



- Caches are divided into **sets** by **physical address**.
- The **set selector** bits of the address choose the set.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

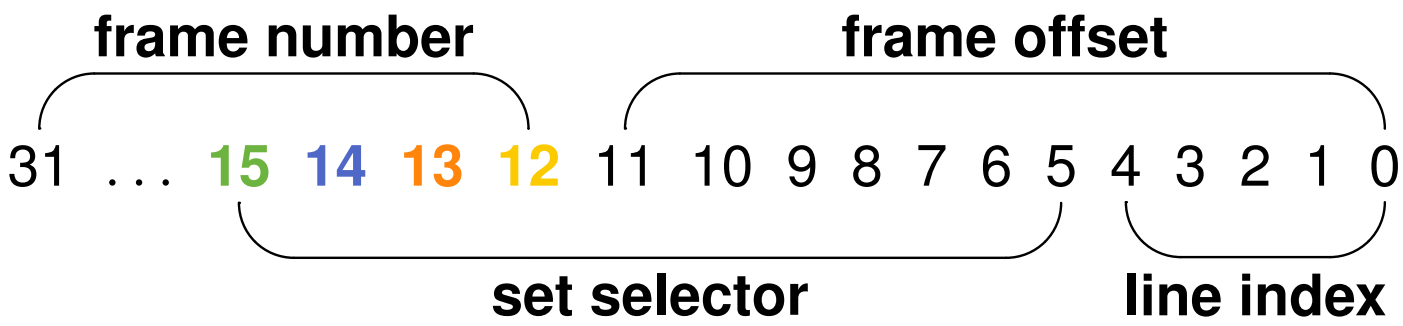
What is Colouring?



- Caches are divided into **sets** by **physical address**.
- The **set selector** bits of the address choose the set.
- These bits (usually) overlap the **frame number**.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

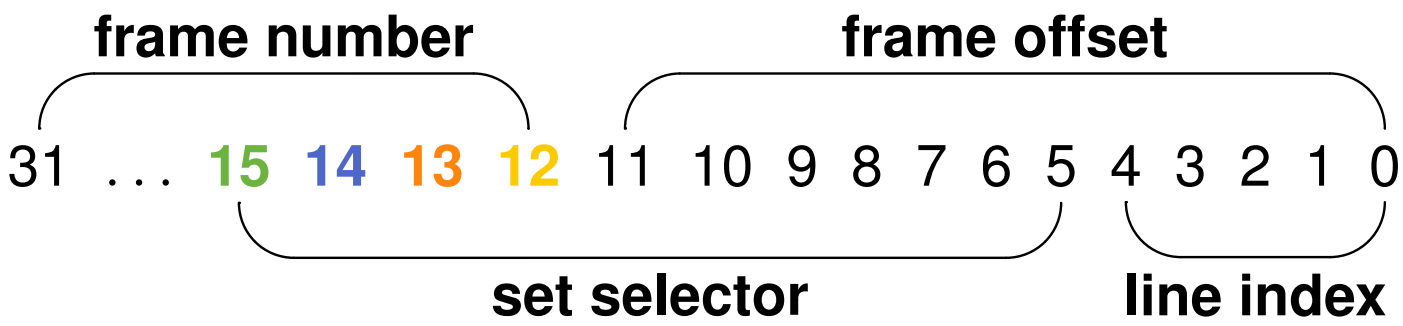
What is Colouring?



- Caches are divided into **sets** by **physical address**.
- The **set selector** bits of the address choose the set.
- These bits (usually) overlap the **frame number**.
- If these **colour bits** differ, the frames cannot collide.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

What is Colouring?



- Caches are divided into **sets** by **physical address**.
- The **set selector** bits of the address choose the set.
- These bits (usually) overlap the **frame number**.
- If these **colour bits** differ, the frames cannot collide.
- The frame number (phys. addr.) is **under OS control**.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Colouring in seL4

In seL4, resource allocation is securely delegated. The initial task splits RAM in to coloured **pools**.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Colouring in seL4

In seL4, resource allocation is securely delegated. The initial task splits RAM in to coloured **pools**.

Advantages

- Very few kernel changes (in seL4).
- Low overhead.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Colouring in seL4

In seL4, resource allocation is securely delegated. The initial task splits RAM in to coloured **pools**.

Advantages

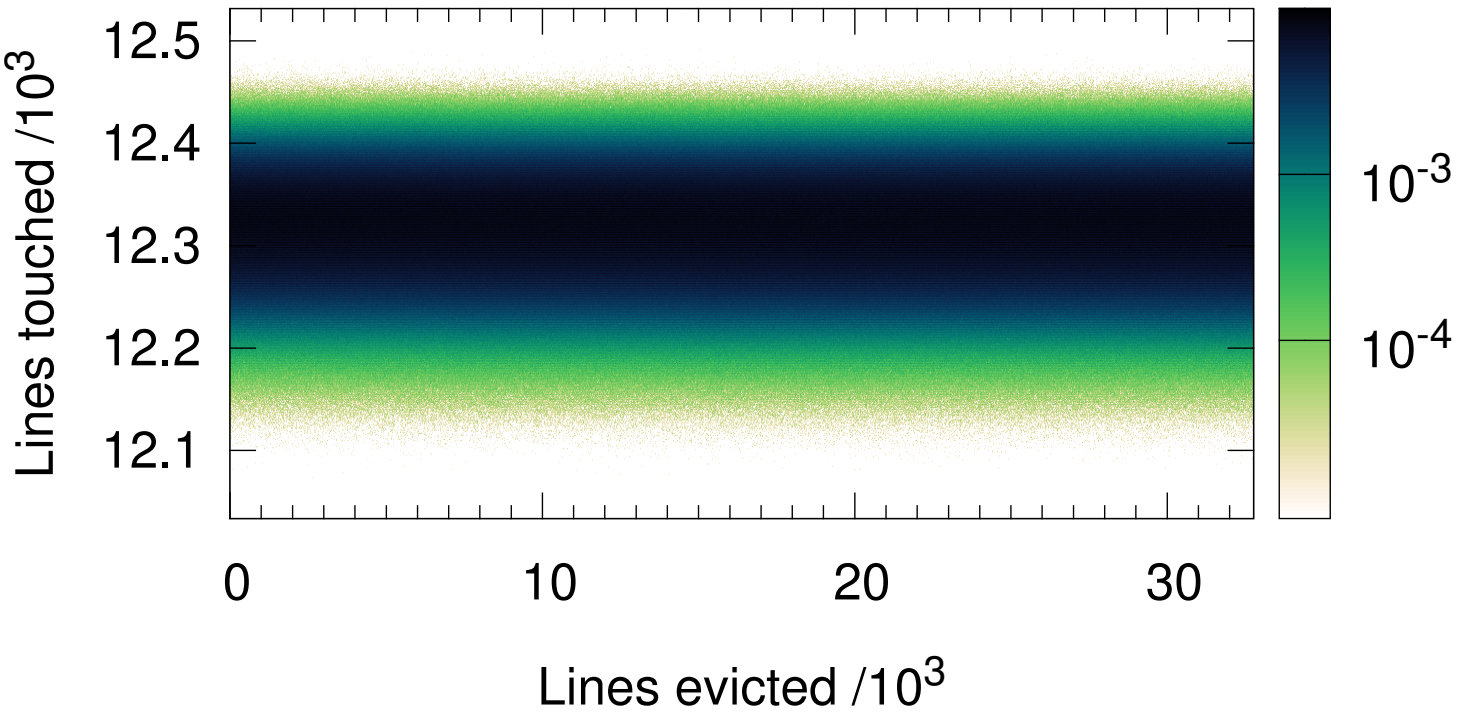
- Very few kernel changes (in seL4).
- Low overhead.

Disadvantages

- Only applies to the cache channel.
- Relies on internal details of cache operation.
- Doesn't work with large pages.
- Hashed caches break everything.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

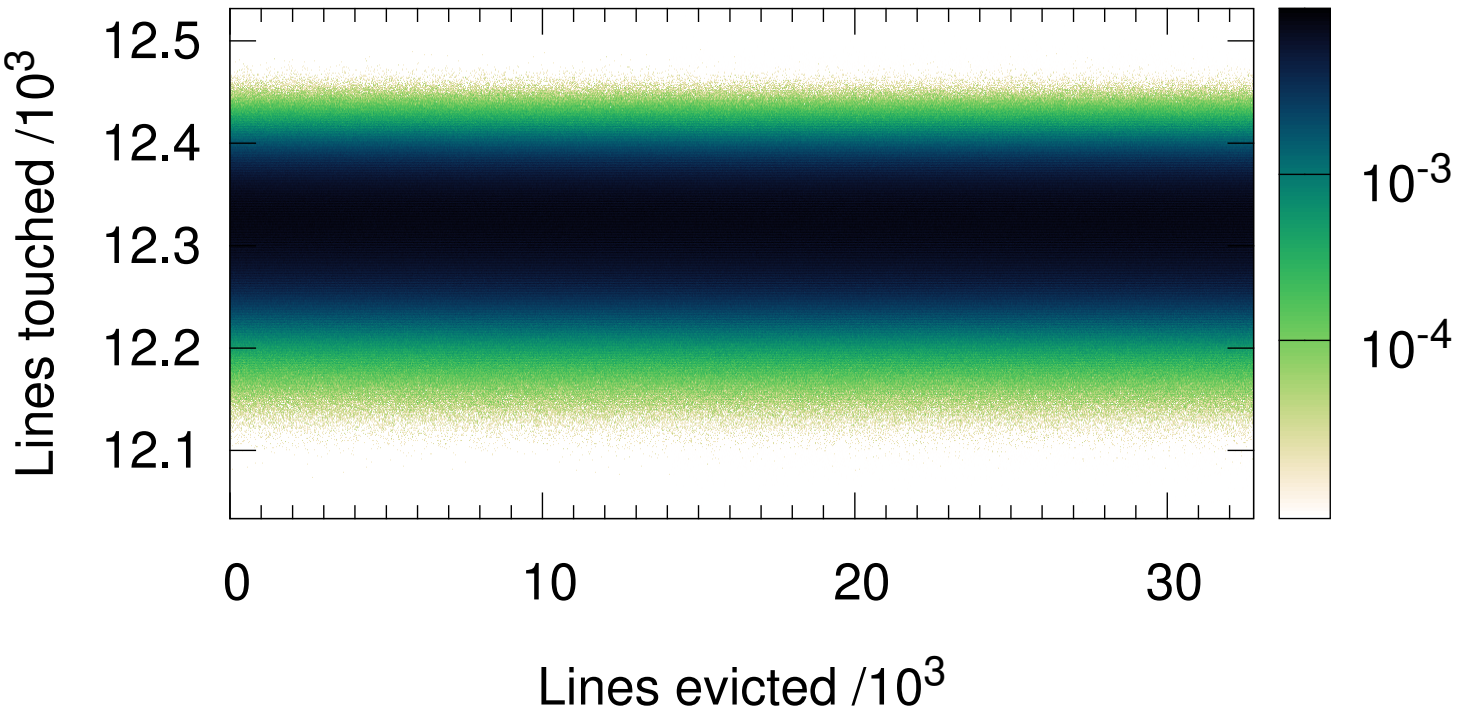
Exynos4412 Cache Channel, Partitioned



- Bandwidth: 15b/s.
- 160× reduction — Much better, but not perfect.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Exynos4412 Cache Channel, Partitioned

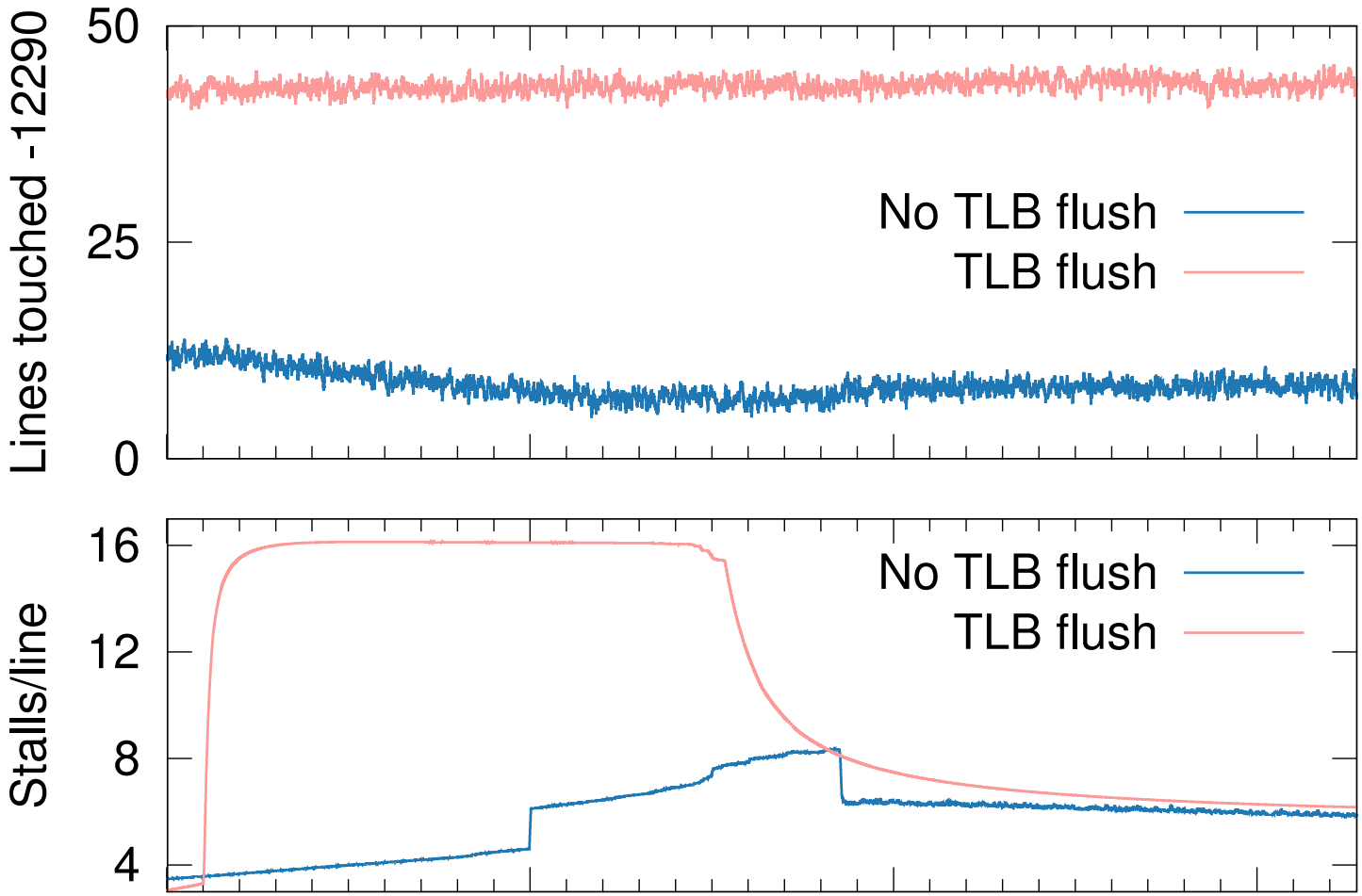


- Bandwidth: 15b/s.
- 160× reduction — Much better, but not perfect.

Where does that 15b/s come from?

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

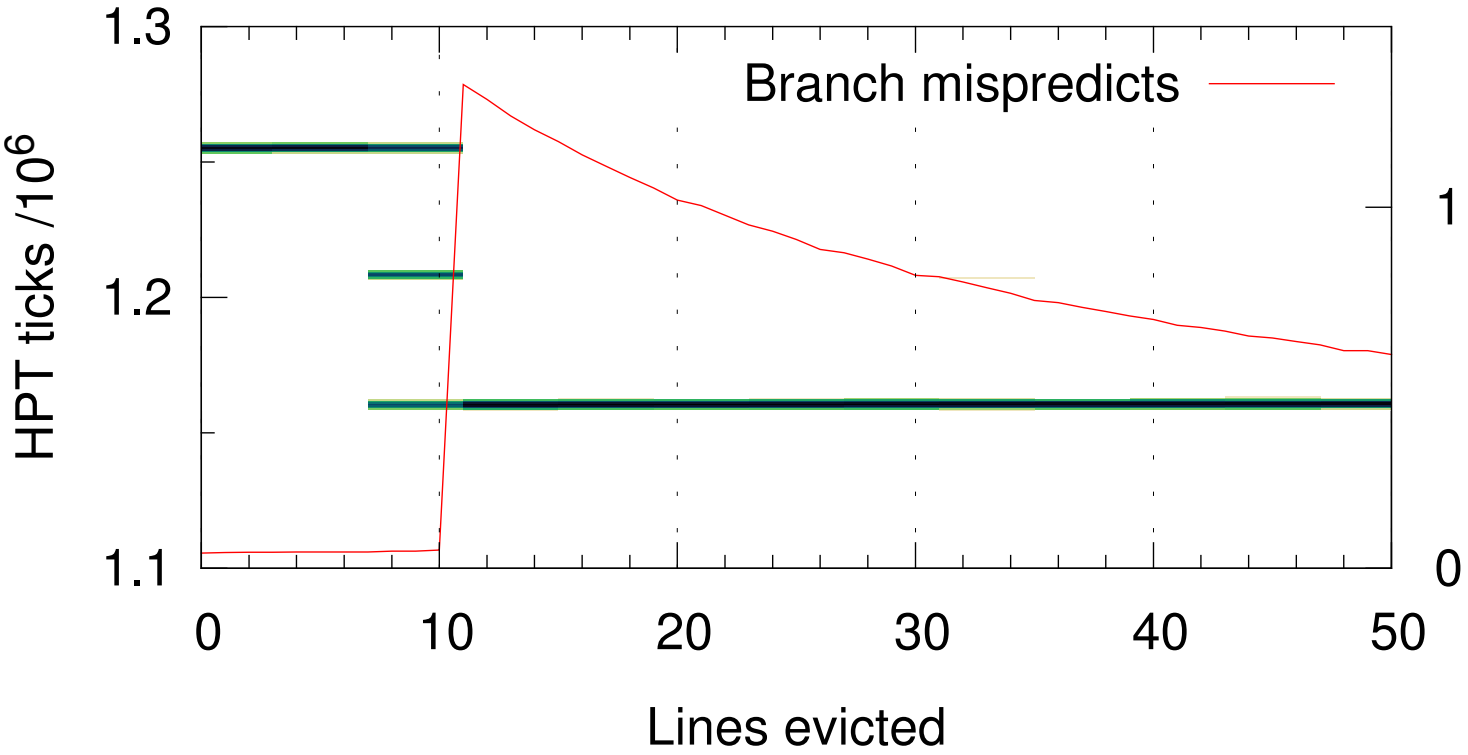
Exynos4412 TLB Channel



- Average rate correlates with TLB misses.
- Flushing on switch removes the signal.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

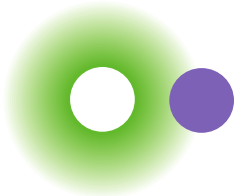
Misprediction and the Cycle Counter



- Cycle counter affected by invisible mispredicts.
- A new (an **unexpected**) channel.
- Event delivery is **precise**, the cycle counter is wrong.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Outline



- Background
  - seL4
  - Channels
  - Experimental Approach
- Local Channels
  - The Cache Channel
  - Instruction-Based Scheduling
  - Cache Colouring
  - New Channels
- Remote Channels
  - Scheduled Delivery
- Summary
  - Outcomes
  - Ongoing Work

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

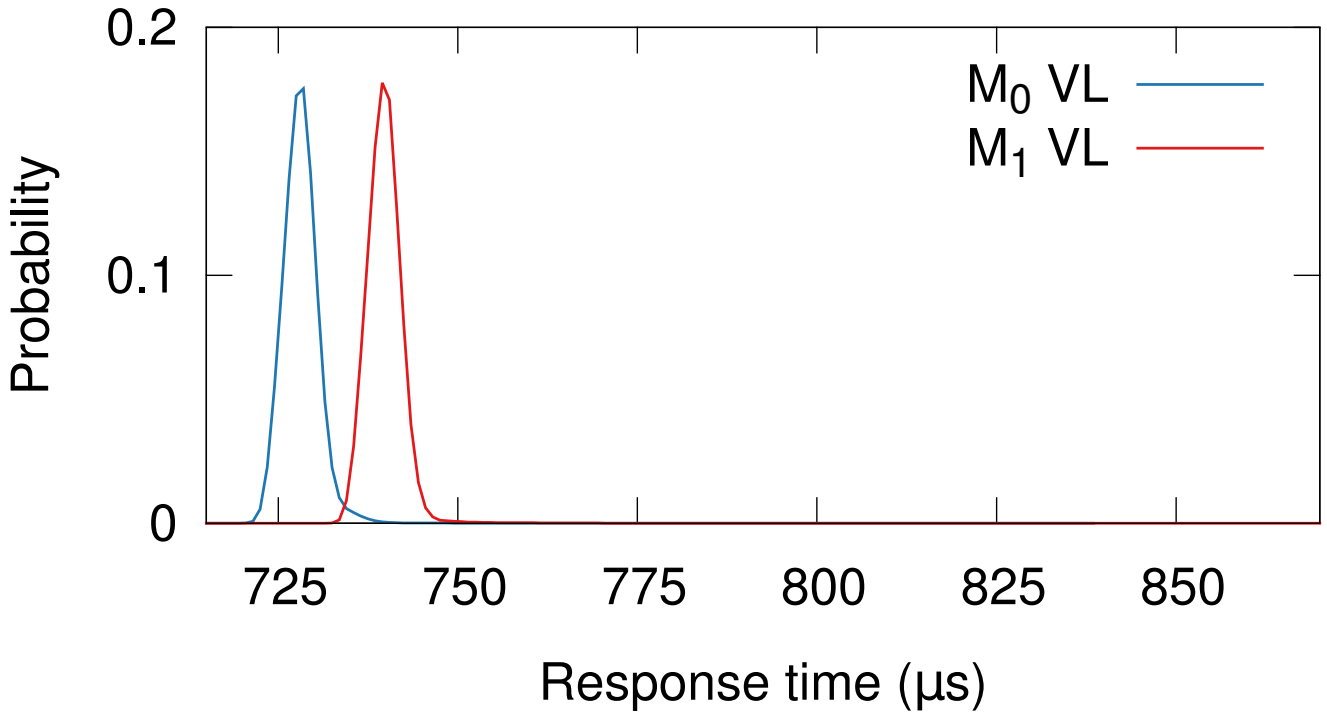
The Lucky-13 Attack on TLS

Al Fardan & Paterson, SSP 2013

- Exploits non-constant-time MAC calculation.
- This is an **Algorithmic side-channel**.
- **Remotely exploitable**.
- We reproduce the distinguishing attack.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

OpenSSL 1.0.1c Response Times



- Nearby attacker (crossover cable).
- Modified vs. unmodified packet.
- Distinguishable with $\approx$ 100% probability.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

The Upstream Fix: Constant-Time Code

Fixed in OpenSSL version 1.0.1e

A constant-time padding/MAC check.

- Now secure (on x86).
- We tested on ARM — still a small channel.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

The Upstream Fix: Constant-Time Code

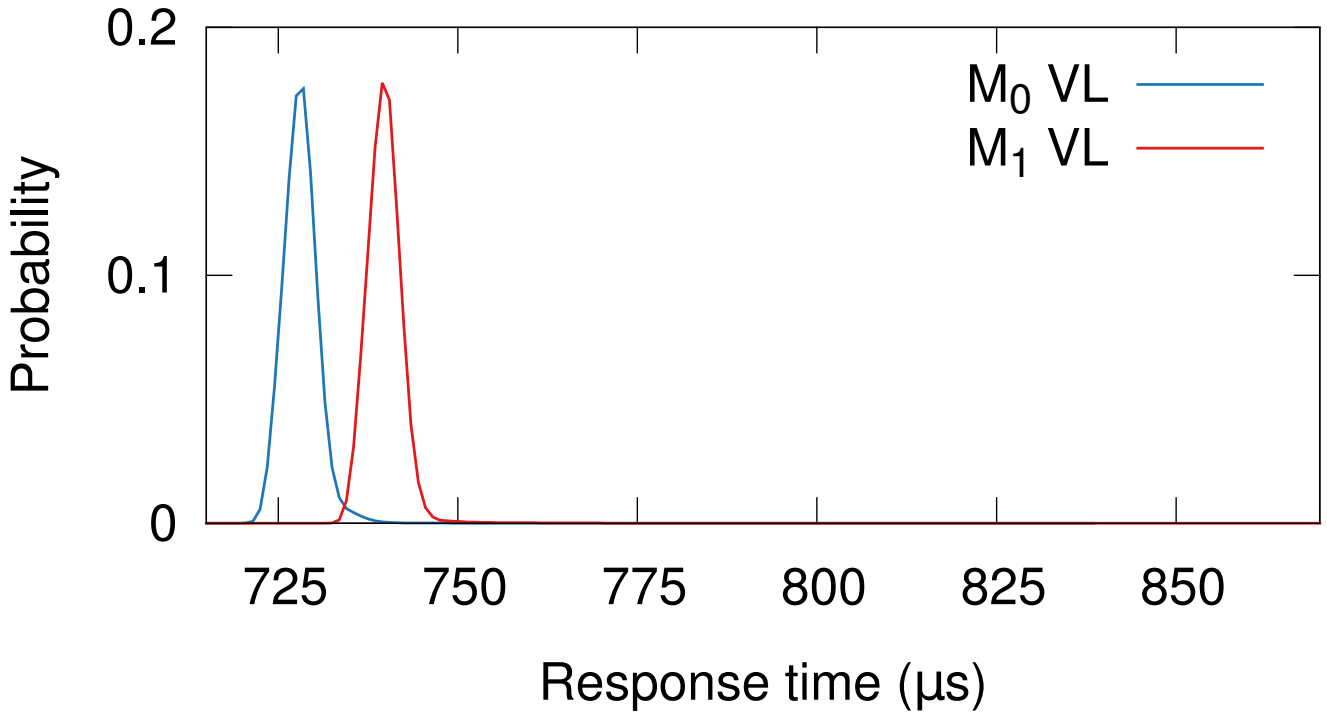
Fixed in OpenSSL version 1.0.1e

A constant-time padding/MAC check.

- Now secure (on x86).
- We tested on ARM — still a small channel.
- We present an OS-level solution, with:
  - Better performance.
  - Lower latency.
  - Lower CPU overhead.
  - No modification of OpenSSL.

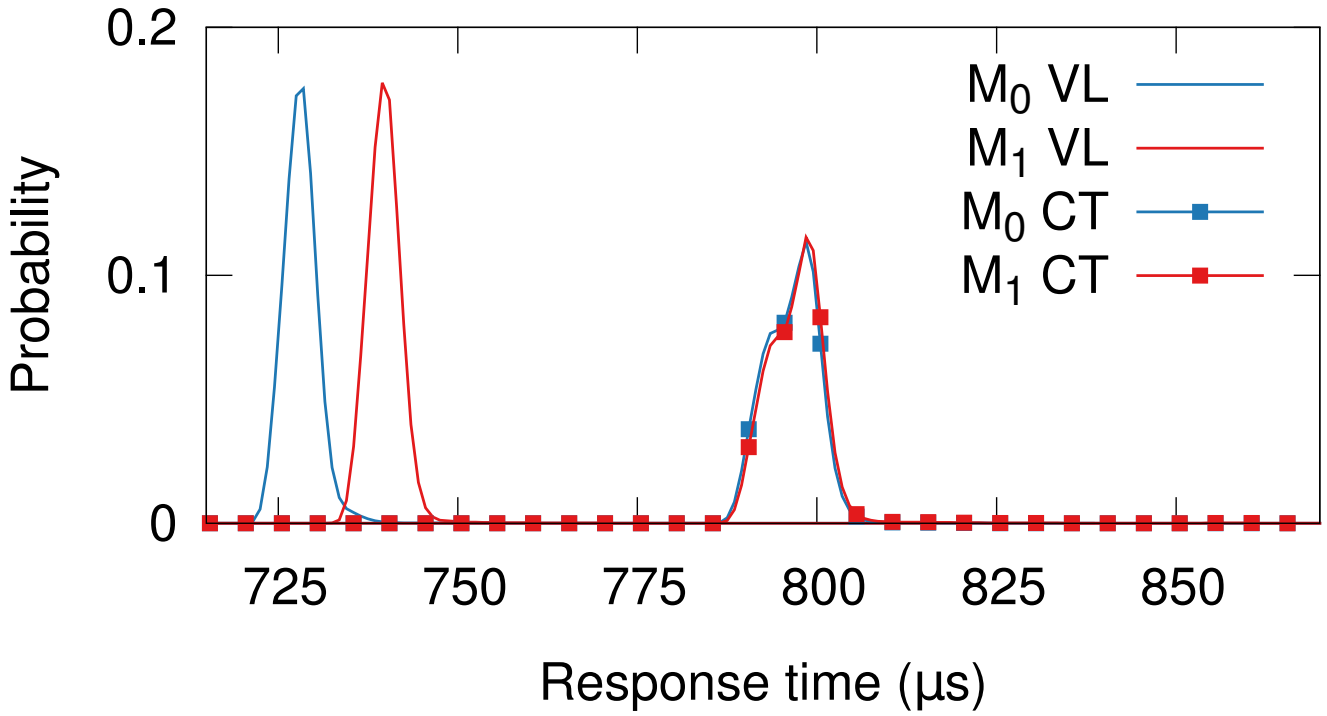
Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

OpenSSL 1.0.1e Response Times



Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

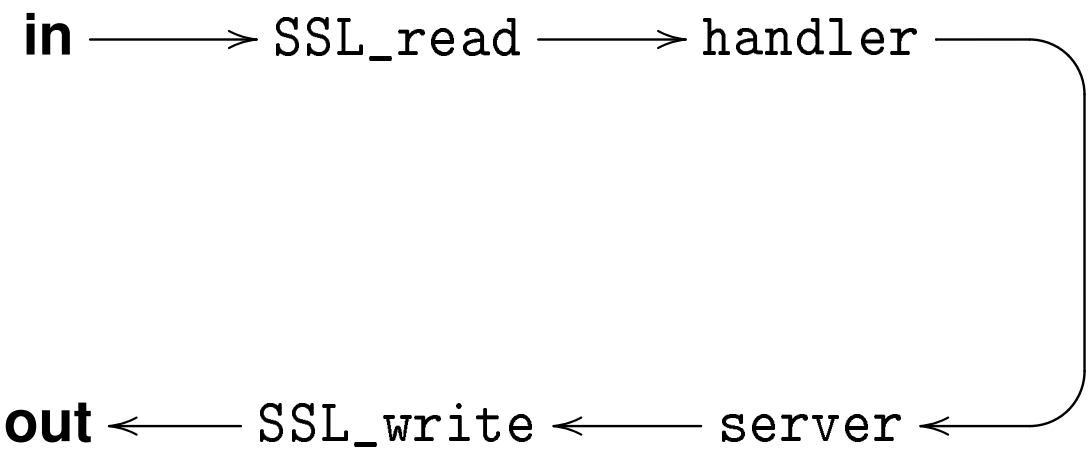
OpenSSL 1.0.1e Response Times

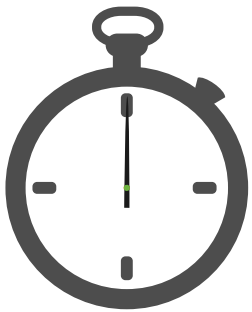


- Constant-time implementation.
- Better, but still distinguishable — 62%.
- 60μs (8.1%) latency penalty.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

What is Scheduled Delivery?

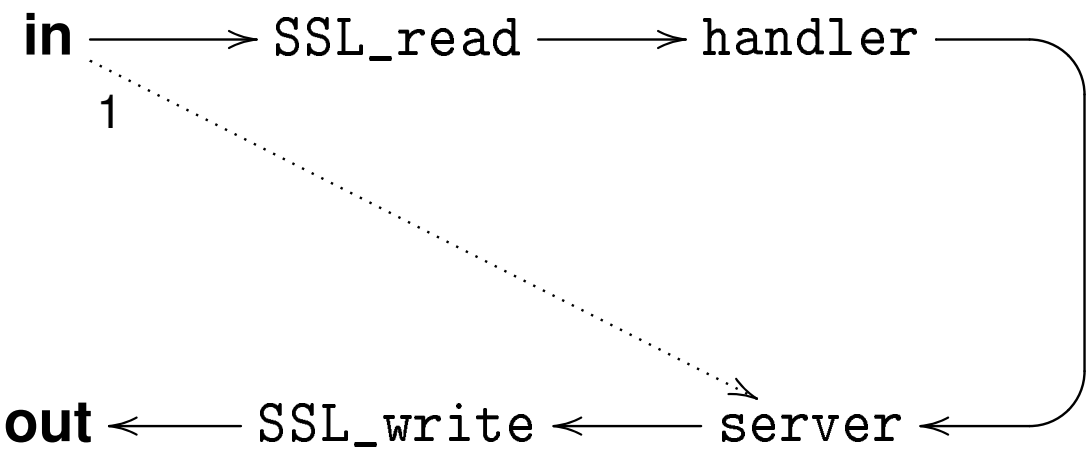


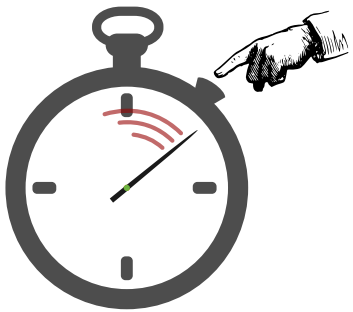


- Separate OpenSSL and application.
- Announce packets over IPC.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

What is Scheduled Delivery?

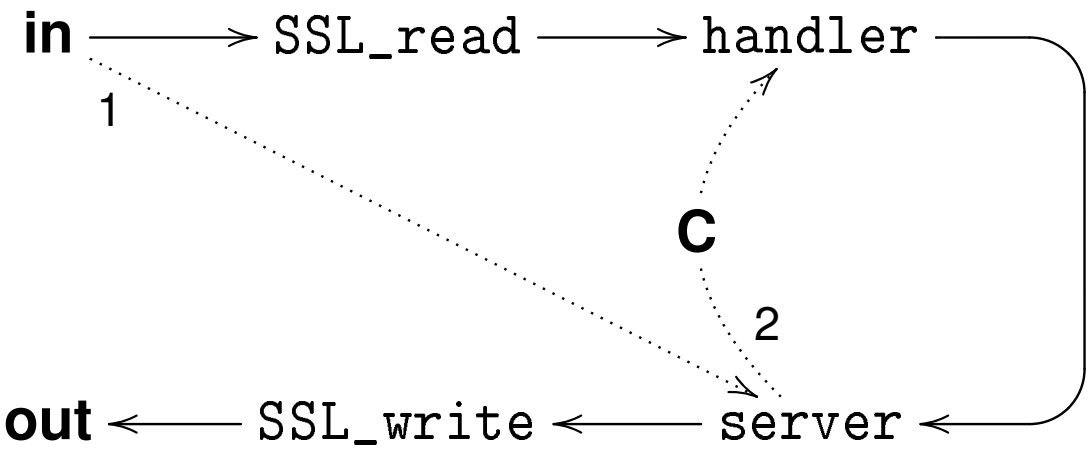


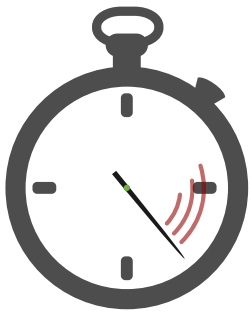


- Separate OpenSSL and application.
- Announce packets over IPC.
- Record arrival time.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

What is Scheduled Delivery?

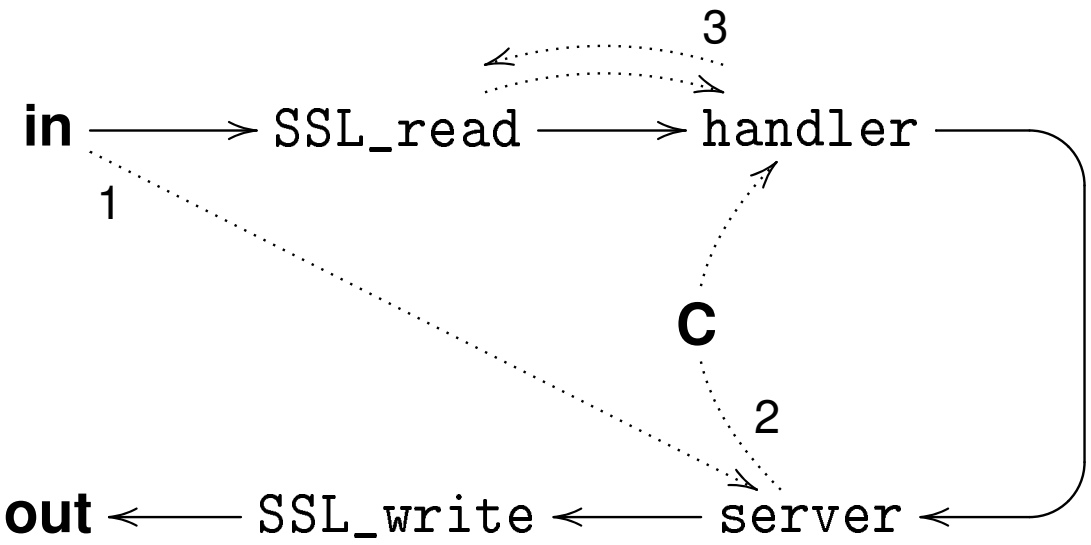


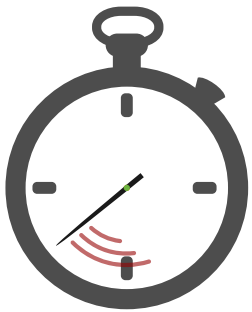


- Separate OpenSSL and application.
- Announce packets over IPC.
- Record arrival time.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

What is Scheduled Delivery?



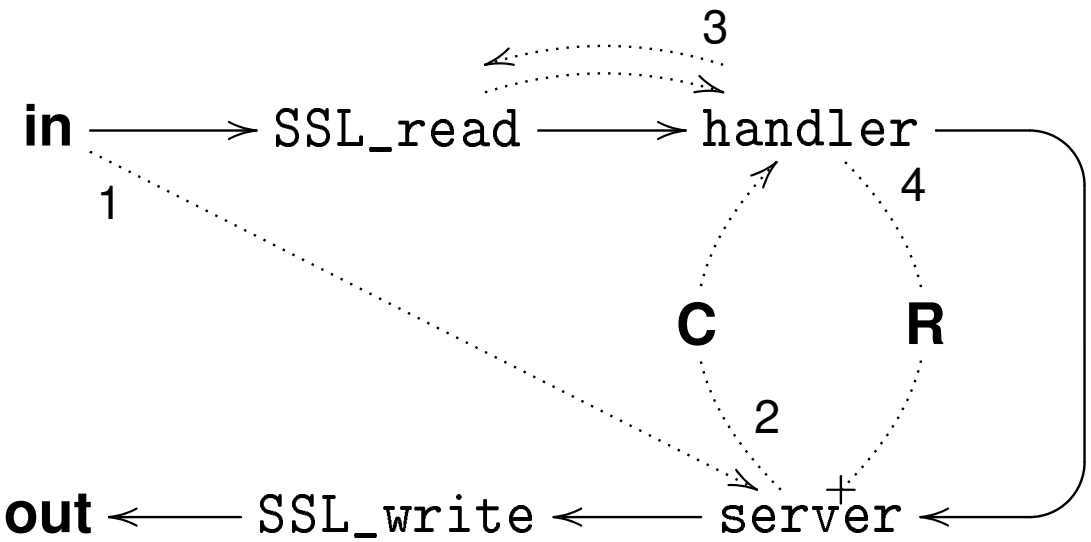


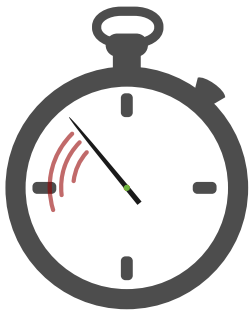
- Separate OpenSSL and application.
- Announce packets over IPC.
- Record arrival time.

NICTA

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

What is Scheduled Delivery?

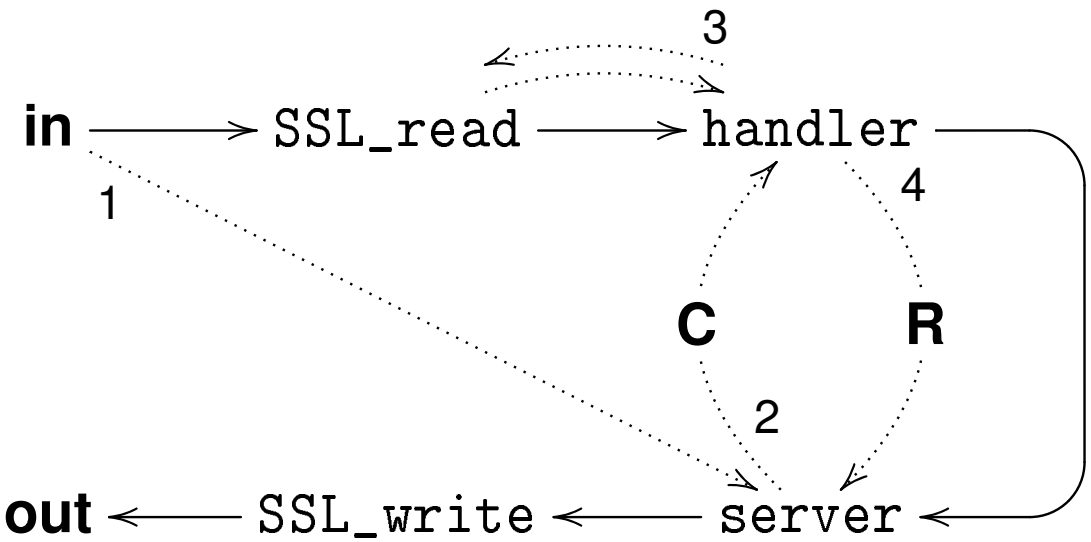




- Separate OpenSSL and application.
- Announce packets over IPC.
- Record arrival time.
- Block response on timer.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

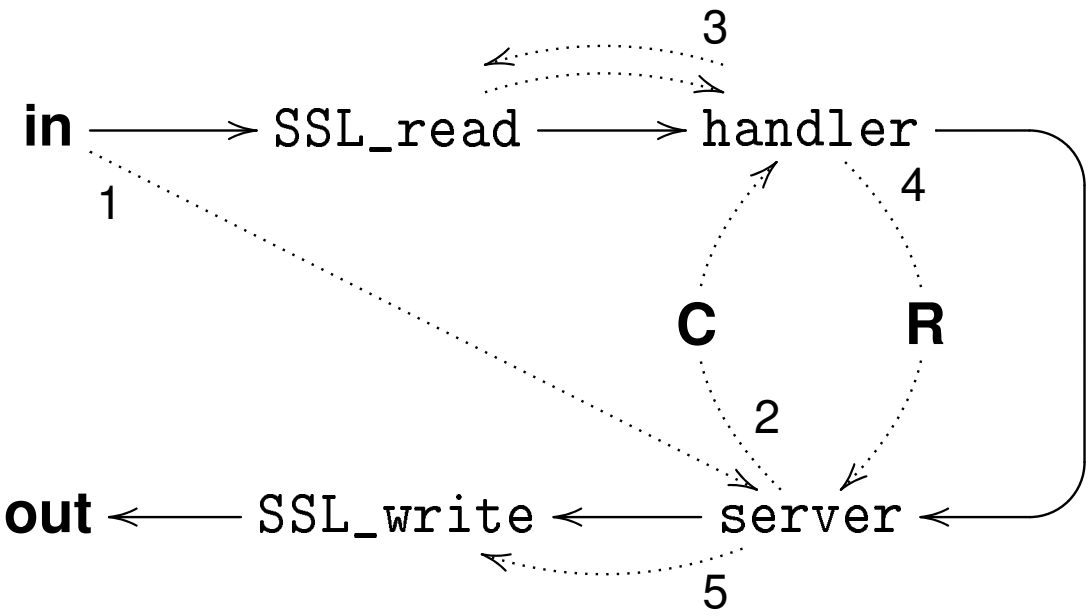
What is Scheduled Delivery?

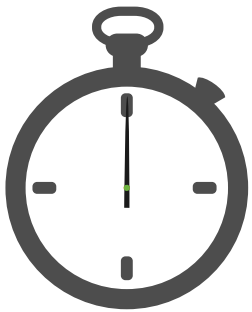


- Separate OpenSSL and application.
- Announce packets over IPC.
- Record arrival time.
- Block response on timer.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

What is Scheduled Delivery?

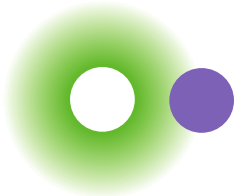




- Separate OpenSSL and application.
- Announce packets over IPC.
- Record arrival time.
- Block response on timer.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Scheduled Delivery on seL4



We use **real-time scheduling** to precisely delay messages.
Provides an efficient **mechanism** to enforce a delay **policy**:
See *Askarov et. al., CCS 2010*.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Scheduled Delivery on seL4

We use **real-time scheduling** to precisely delay messages.
Provides an efficient **mechanism** to enforce a delay **policy**:
See *Askarov et. al., CCS 2010*.

Advantages

- Uses existing IPC controls — no modifications.
- Fast and effective (see next slide).

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Scheduled Delivery on seL4

We use **real-time scheduling** to precisely delay messages.
Provides an efficient **mechanism** to enforce a delay **policy**:
See *Askarov et. al., CCS 2010*.

Advantages

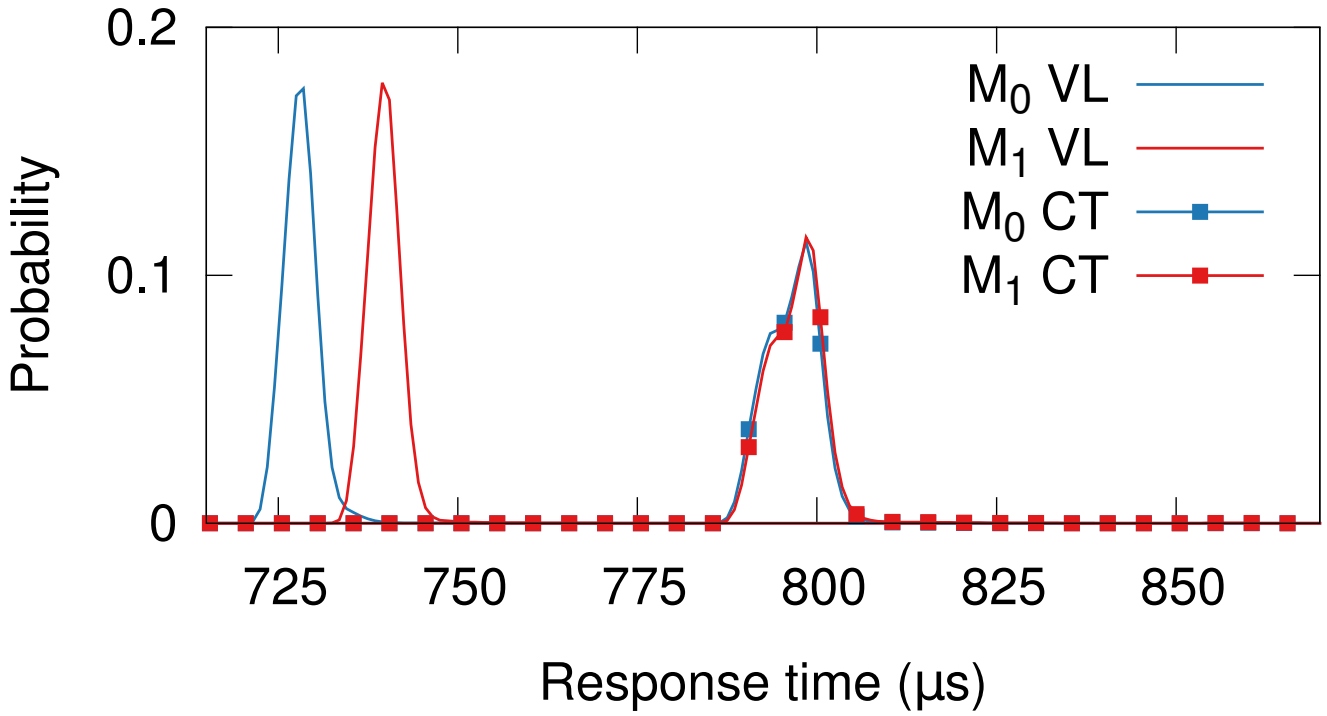
- Uses existing IPC controls — no modifications.
- Fast and effective (see next slide).

Disadvantages

- Specific to remote/network attacks.
- Need to wrap (but not modify) vulnerable component.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

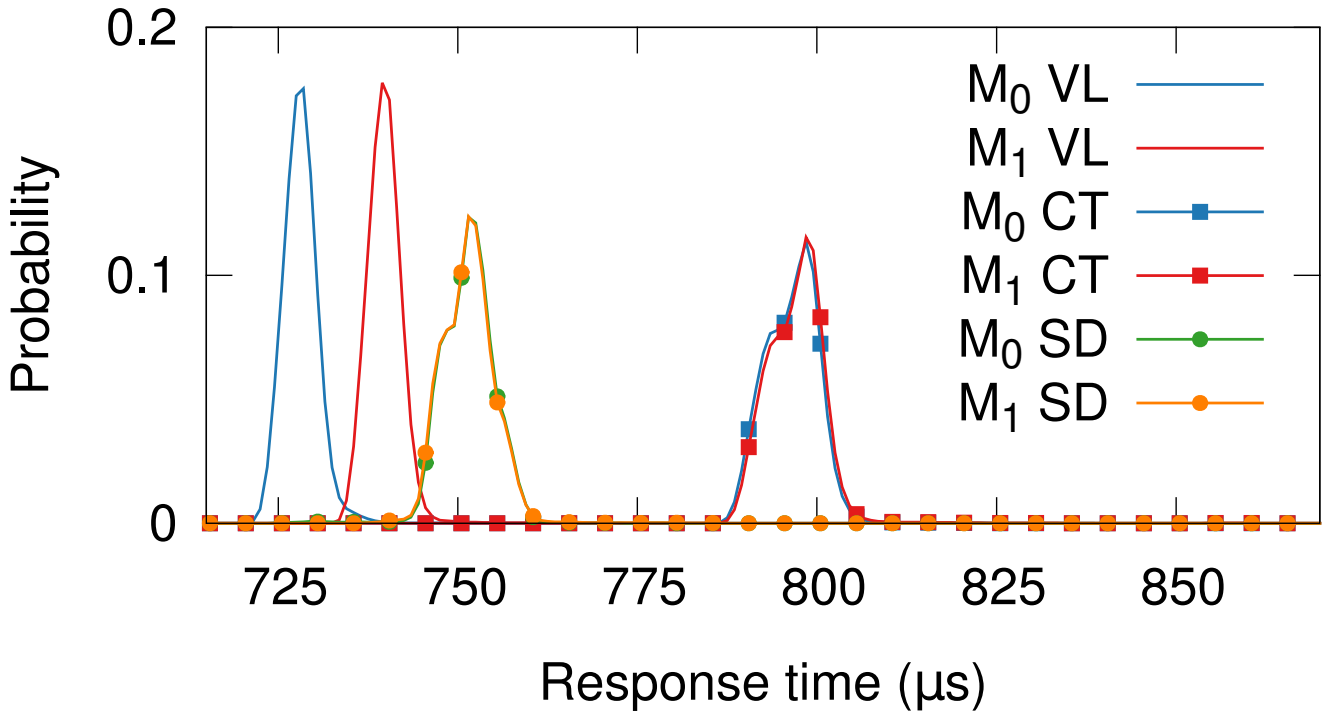
Security of Scheduled Delivery



NICTA

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

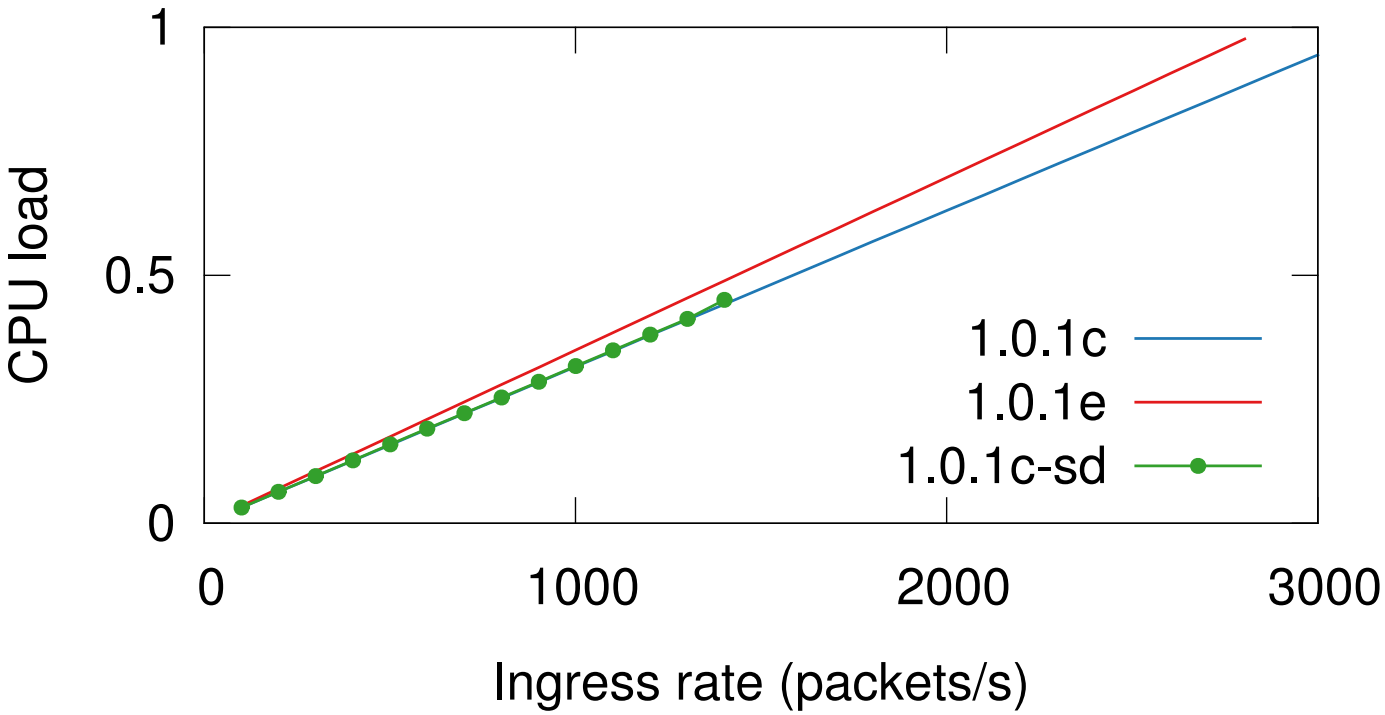
Security of Scheduled Delivery



- More secure — 57% distinguishable.
- Faster — 10μs (1.4%) latency penalty.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

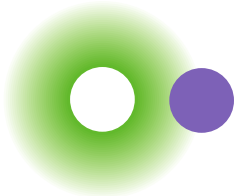
Performance of Scheduled Delivery



- Lower overhead (2% vs. 11%), but earlier saturation.
- This is a worst-case benchmark — no server work.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Outline



- Background
  - seL4
  - Channels
  - Experimental Approach
- Local Channels
  - The Cache Channel
  - Instruction-Based Scheduling
  - Cache Colouring
  - New Channels
- Remote Channels
  - Scheduled Delivery
- Summary
  - Outcomes
  - Ongoing Work

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

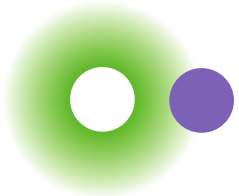
Outcomes

These channels are real

- We managed to exploit every channel we tried.
- The bandwidth is high, and growing.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Outcomes



These channels are real

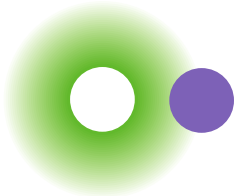
- We managed to exploit every channel we tried.
- The bandwidth is high, and growing.

They're getting worse

- Countermeasures get less effective on newer chips.
- New hardware channels e.g. branch predictor.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Outcomes



These channels are real

- We managed to exploit every channel we tried.
- The bandwidth is high, and growing.

They're getting worse

- Countermeasures get less effective on newer chips.
- New hardware channels e.g. branch predictor.

But there is hope

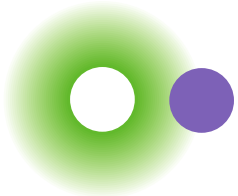
- Resource partitioning (e.g. colouring) is effective.
- Repurpose hardware QoS features.
- OS-level techniques can help.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Ongoing Work

- Ongoing project at NICTA (Qian Ge).
- Developing a fully cache-coloured seL4.
- We use these tools to evaluate the result.

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work



NICTA

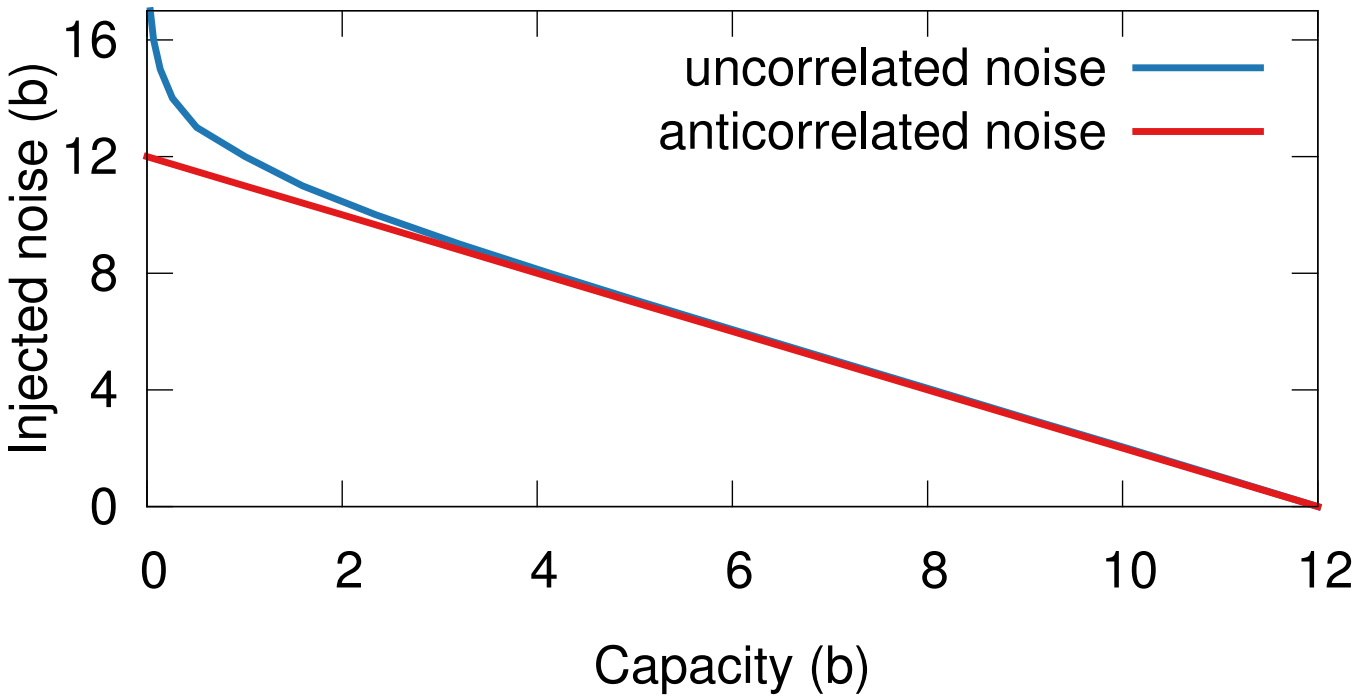
Questions?

Data and Tools `http://ssrg.nicta.com.au/projects/TS/timingchannels.pml`

seL4 Is Open Source! `http://sel4.systems`

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work

Capacity vs. Injected Noise



- Noise injection doesn't scale.
- Increasing determinism is the way to go.

NICTA

Background
seL4
Channels
Experimental Approach
Local Channels
The Cache Channel
Instruction-Based Scheduling
Cache Colouring
New Channels
Remote Channels
Scheduled Delivery
Summary
Outcomes
Ongoing Work