Robust covert channels based on DRAM power consumption

Thales Bandiera Paiva[1] Javier Navaridas[2] Routo Terada[1]

[1]Institute of Mathematics and Statistics
University of Sao Paulo, Brazil

[2]School of Computer Science
University of Manchester, U.K.

tpaiva@ime.usp.br javier.navaridas@manchester.ac.uk rt@ime.usp.br

22nd Information Security Conference

September 17, 2019

Software-based power measurement

Power consumption is a major concern for computing devices

- Small devices have low battery life
- Huge clusters can consume a prohibitive amount of power

Some CPUs of major vendors provide interfaces for power management

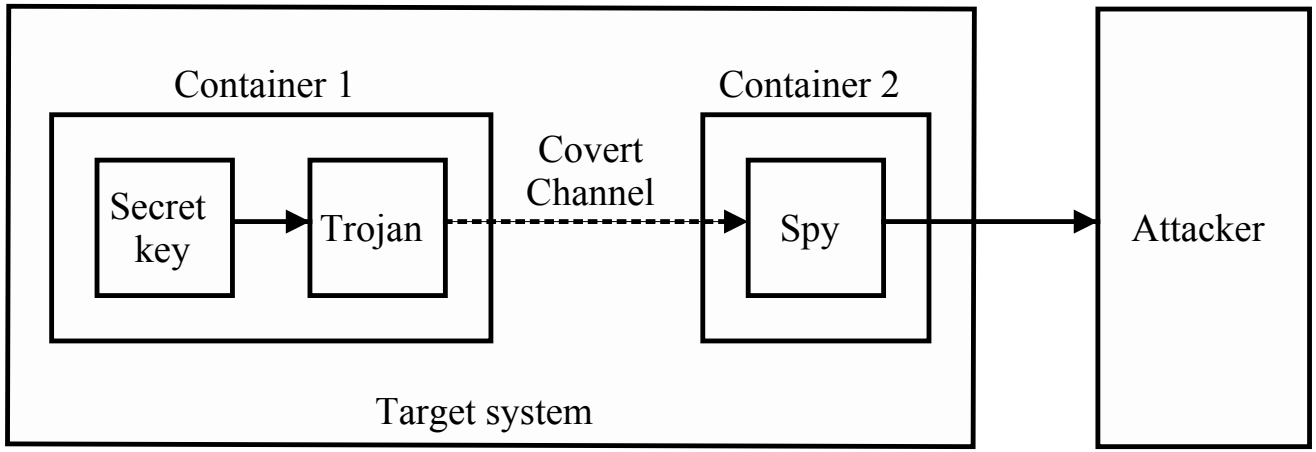
- Intel Running Average Power Limit (RAPL)
- AMD Application Power Management (APM)

Unfortunately this potentially comes with two security risks

- Covert channels
- Side-channel attacks

Covert Channels

- Suppose we have two processes that are meant to be isolated
- A covert channel enables these two processes to communicate
- Are used to exfiltrate secrets from a target system
- Critical systems should resist known covert channels



Power consumption leaks information: Side-channel attacks

Side-channel attacks

- Kocher et al.'99 introduced Simple and Differential Power Analysis
- Security against SPA and DPA is required for crypto implementations
- Main limitation is that attacker needs physical access
- Mantel et al.'18 showed a key distinguishing attack using software-based energy consumption

Kocher, Paul, Joshua Jaffe, and Benjamin Jun. "Differential power analysis." Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1999.

Mantel, Heiko, et al. "How secure is green IT? The case of software-based energy side channels." European Symposium on Research in Computer Security. Springer, Cham, 2018.

Power consumption leaks information: Covert channels

Covert channels[1]

- Murdoch's '06 used temperature's effect on clock skew (21bph)
- Masti's et al. '15 used core temperature (50bps)
  - Long et al. '18 improved to 600 bps in 2018
- Miedl and Thiele '18 showed a covert channel based on RAPL
  - It uses only CPU power
  - It achieves 1000 bps in their notebook and 200 bps in their server

Murdoch, Steven J. "Hot or not: Revealing hidden services by their clock skew." Proceedings of the 13th ACM conference on Computer and communications security. ACM, 2006.

Masti, Ramya Jayaram, et al. "Thermal covert channels on multi-core platforms." 24th USENIX Security Symposium (USENIX Security 15). 2015.

Long, Zijun, et al. "Improving the efficiency of thermal covert channels in multi-/many-core systems." 2018 Design, Automation and Test in Europe Conference and Exhibition (DATE). IEEE, 2018.

Miedl, Philipp, and Lothar Thiele. "The security risks of power measurements in multicores." Proceedings of the 33rd Annual ACM Symposium on Applied Computing. ACM, 2018.

[1] Transmission rates for 15% error

Our contribution

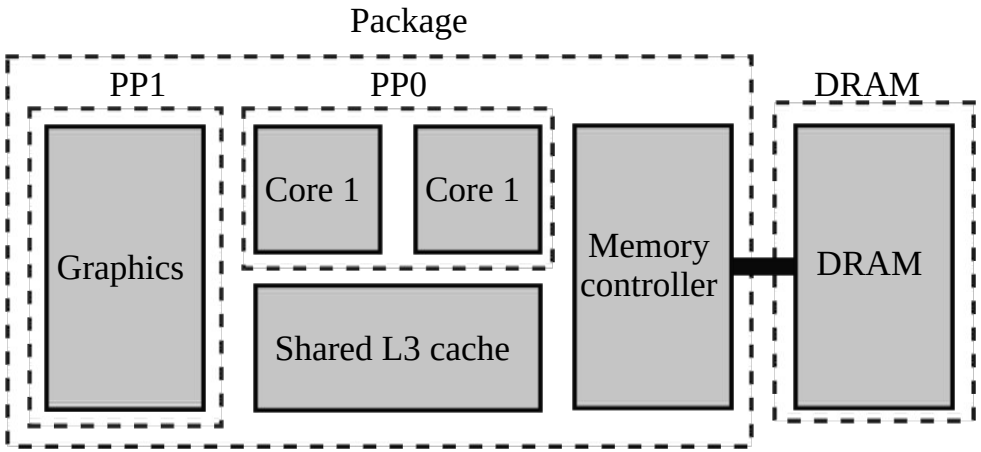
We propose the first covert channel based on DRAM power

- It uses the RAPL interface for software-based power monitoring
- No privileged instruction needed
- The channel construction is validated under two platforms
- We tested its robustness against CPU and DRAM benchmarks
- The results improves upon previous similar proposals with respect to
  - bandwidth
  - error rate
  - robustness against interference

Intel Software-based power measurements

- Internal performance registers give power consumption estimates
- Modern Intel CPU's provide the RAPL interface for power monitoring
- Sampling rate is around 1000 Hz
- In Linux, RAPL estimates can be read using the `powercap` module
  - This does not require special privileges!

RAPL domains available in our notebook's CPU

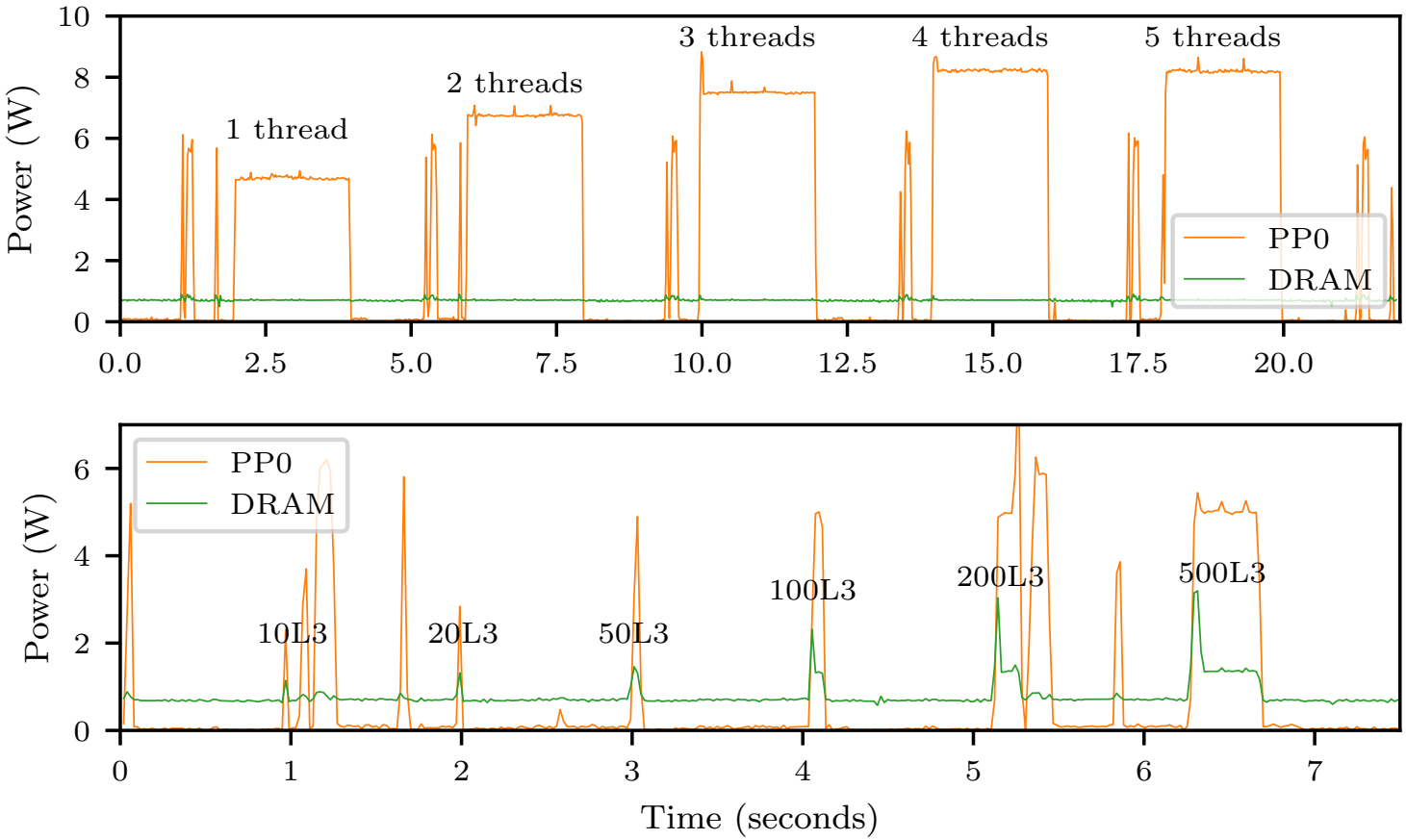


Modulating power consumption

We can modulate power consumption by

- Doing CPU-intensive task with some number of threads
- Doing Memory-intensive task (`memset`)

Power consumption modulation in the Notebook setup (4 hyperthreads)



Our covert channel (in theory)

Shared parameters

- Let t_s be the transmission starting time
- Let t be some fixed modulation time interval in seconds

Encoding (Trojan)

- $1 \Rightarrow$ `memset` and sleep until t seconds have passed
- $0 \Rightarrow$ do nothing for t seconds

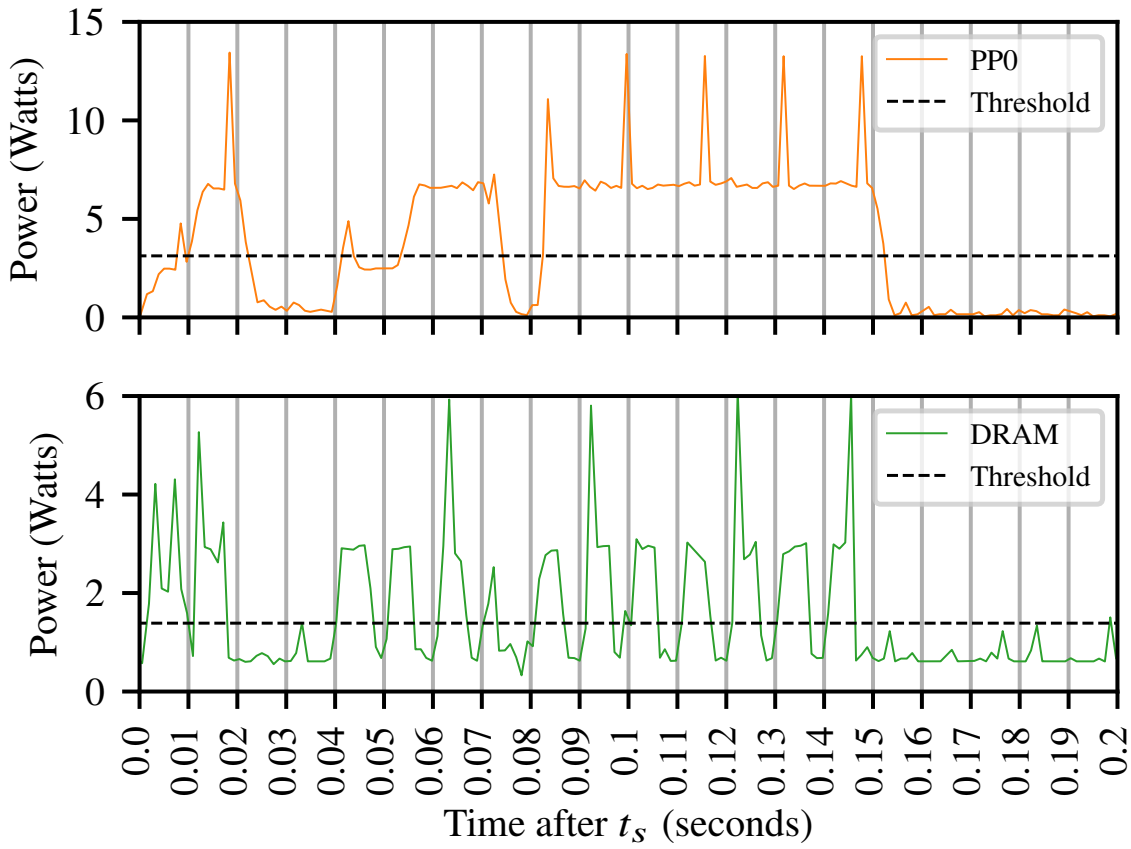
Decoding (spy)

- Start recording power consumption at instant t_s
- To decode the i-th bit do
  - Let P be the average power between $(t_s + (i-1)t)$ and $(t_s + it)$
  - If P is above some threshold, decode as 1
  - Otherwise decode as 0

Our covert channel (in practice)

Example of the covert channel transmitting at 100 bps

- Using only CPU power modulation (top)
- Using only DRAM power modulation (bottom)



Technical problems

Sync: It is not reasonable to expect Trojan and spy to share a clock

- Trojan can start transmission by sending a predefined message
- Spy uses t_c as the point that better decodes to the syncing message

Length: The spy needs to know the message length

- Messages of fixed size may be too restrictive
- Can use a number of the first bits to encode the message length

Threshold: Choose a decoding threshold

- When message bits are balanced, use the average power as threshold
- Trojan sends initial known message for spy to learn the best threshold

Evaluating the covert channels

Three main targets

- Transmission rate
- Error rate
- Robustness against application interference

Experiments

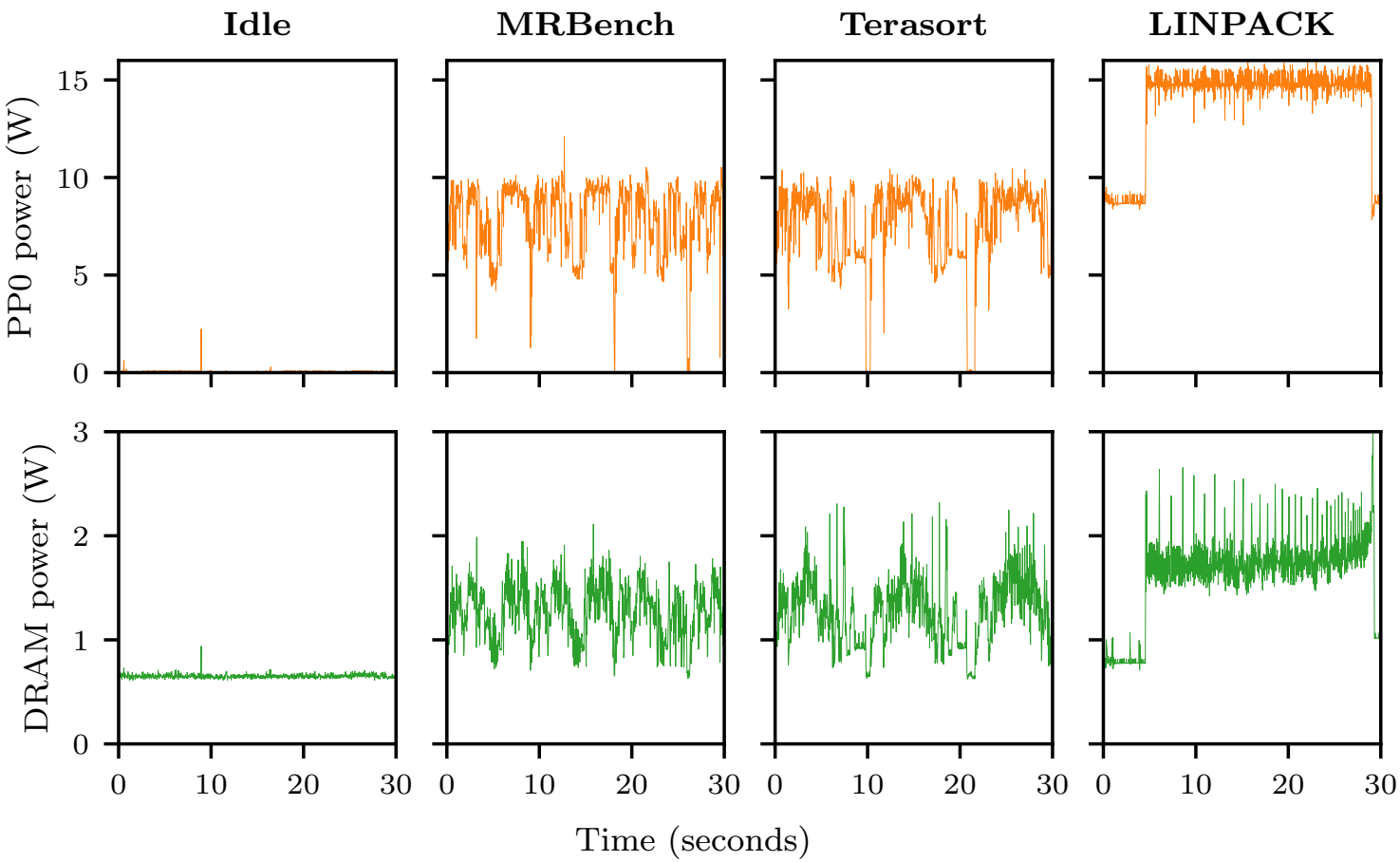
- `memset` with memory chunks of different sizes
- Transmission rates between 300 and 600 bps
- Two environments
  - **Notebook:** Intel i5-4210U (dual-core), L3 of $L3_N = 3MB$
  - **Desktop:** Intel i7-8700 (hexa-core), L3 of $L3_D = 12MB$
- 10 messages generated at random for each set of parameters
- Messages of 1000 bits: 100 for syncing and 900 for payload

Power consumption profiles of the benchmarks

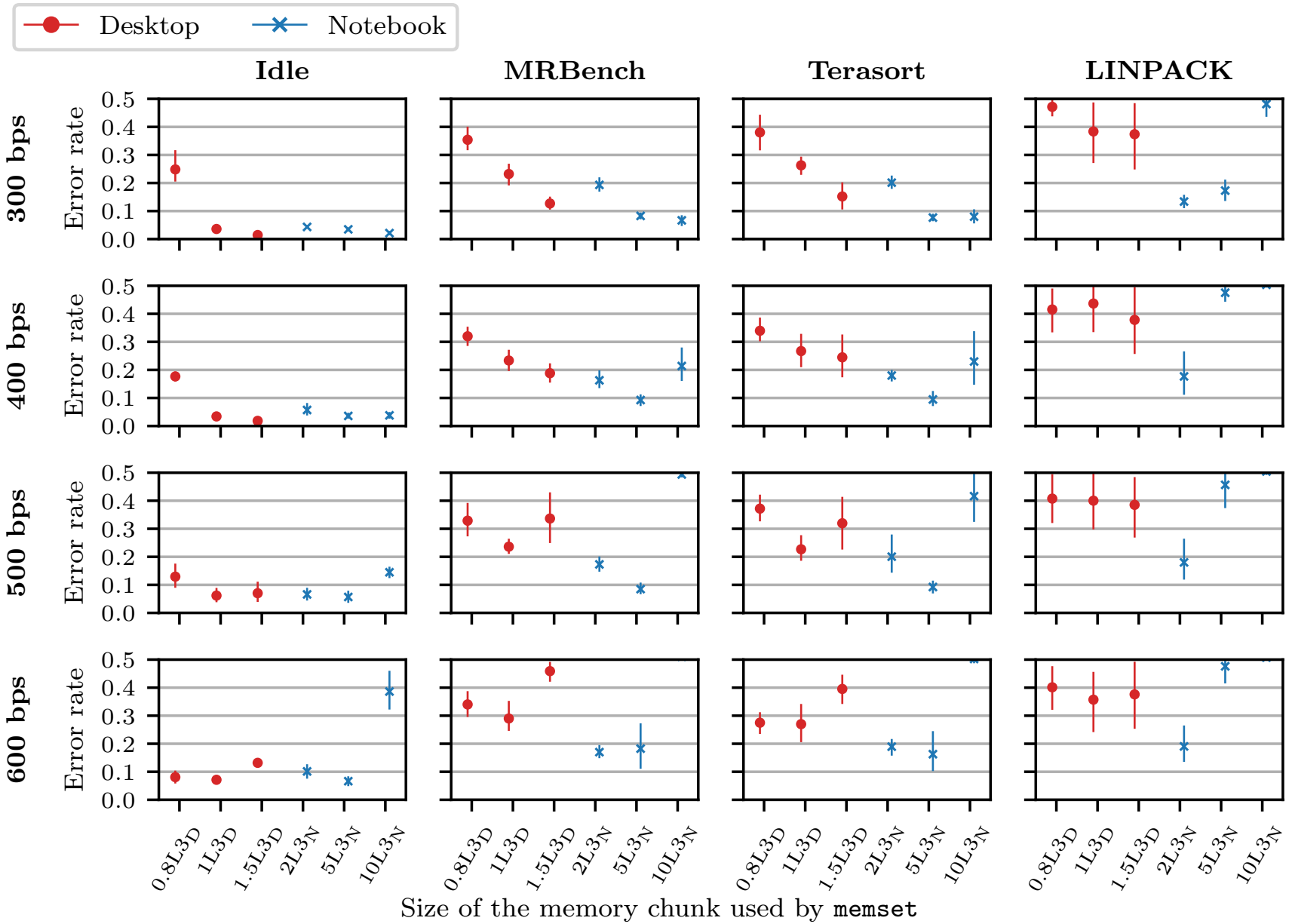
We considered the power consumption of 3 benchmarks

- MRBench and Terasort are similar
- LINPACK appears to impact power consumption the most

Power consumption profiles of the benchmarks in the Notebook setup



Experimental results



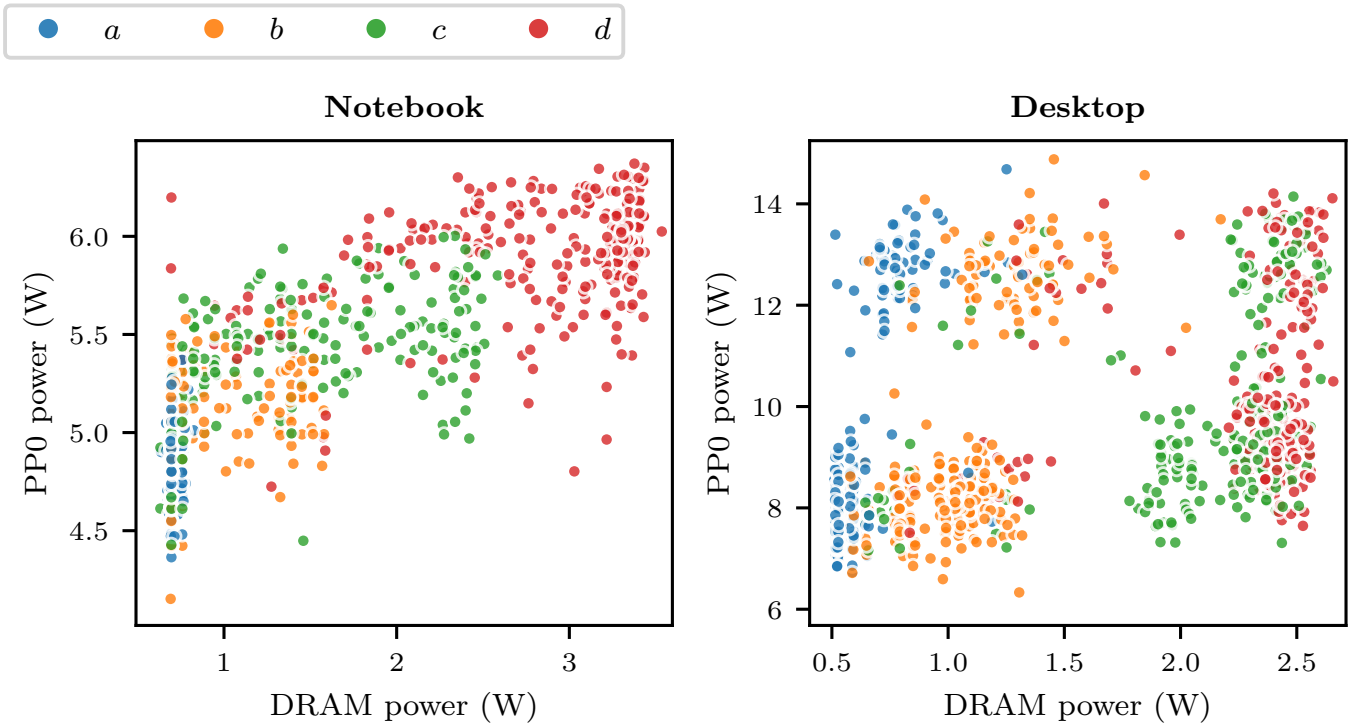
Experimental results

- Intermediate `memset` parameters are better
  - Easily distinguishable
  - Do not cause synchronization issues
- The notebook is more robust than the Desktop
  - Desktop's larger number of cores work too much under high load
- LINPACK is the benchmark which affects the channel the most

Achieving higher transmission rates

Recall: size of the `memset` argument is correlated with power consumption

Symbol	*a*	*b*	*c*	*d*
(Notebook) `memset`	0L3	1.5L3	3L3	6L3
(Desktop) `memset`	0L3	0.8L3	1.2L3	1.4L3



Caveat: more complex decoder

Better decoders with clustering algorithms

The simple binary decoder procedure does not make sense anymore

- We propose the use of the Random Forests Classifier
- This classifier needs a training set with correct labels

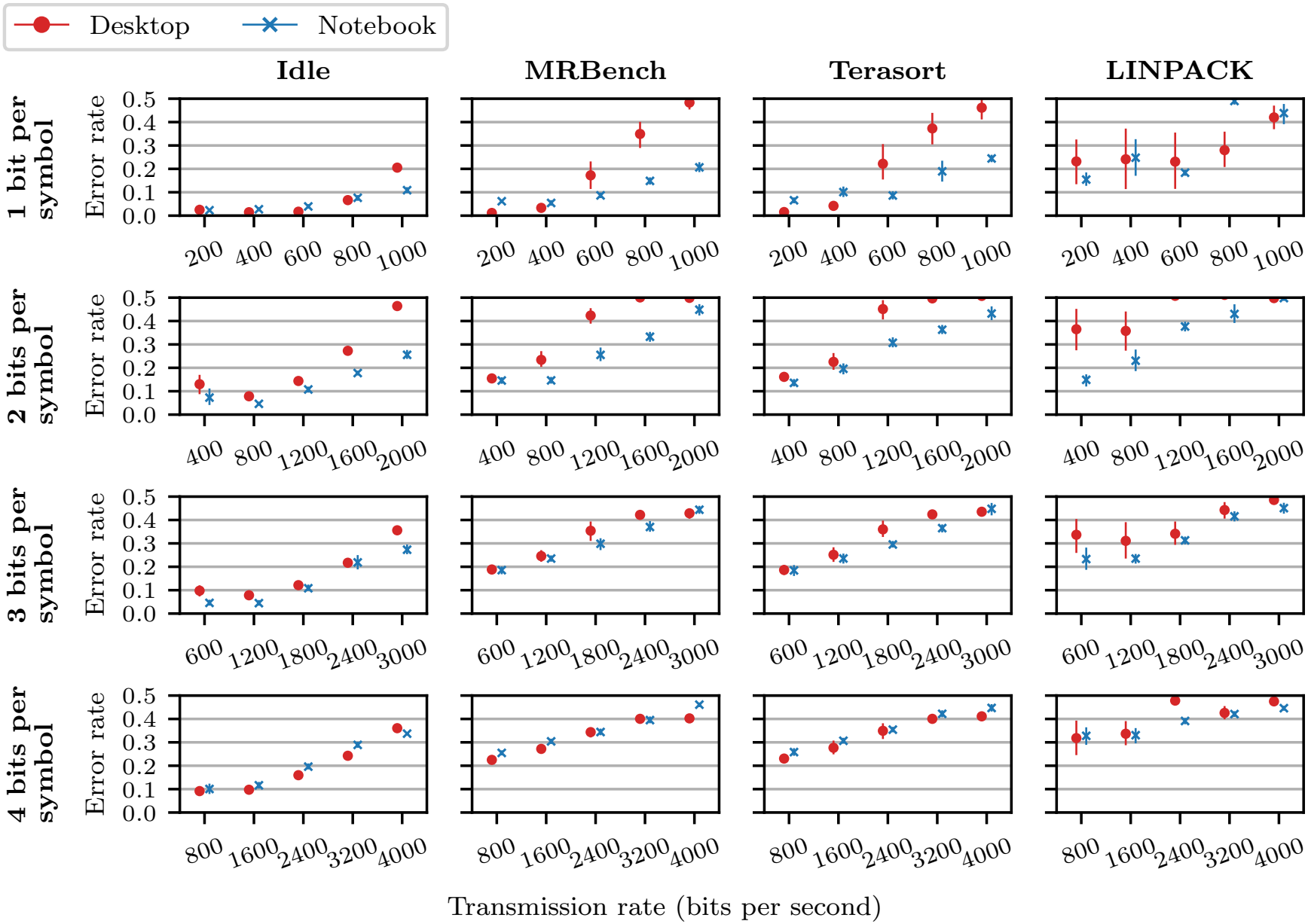
Trojan and spy now use two predefined messages

- First a syncing message as before
- Then a message for the spy to train its classifier

To analyze the performance, we considered

- A training message of size 5000
- 10 independent messages of size 1000
- Spy used the same trained classifier for all 10 messages

Experimental results



Experimental results

- Results are much better than with the simple decoder
- Notebook still more robust than Desktop
- It is more difficult for the Desktop to transmit at high rates
  - Its L3 is too big, and `memset` starts causing synchronization problems
- Results can be significantly improved if specific parameters are chosen

Comparison with a similar proposal

DRAM and CPU power modulation results in a considerable improvement

Miedl and Thiele [MT08]			**Our work**		
Processor	Transfer rate	Error rate	Processor	Transfer rate	Error rate
Intel Xeon E5-2690 (octa-core)	200 bps	≈15%	Intel Core i7-8700 (hexa-core)	2400 bps	≈15%
Intel Core i7-4710MQ (quad-core)	1000 bps	≈15%	Intel Core i5-4210U (dual-core)	1800 bps	≈10%

Conclusion

- We presented the first covert channel based on DRAM power
- It uses simple algorithms for encoding and decoding
- It can can transmit at high rates with low error rates
- It is robust against application interference
- The channel does not use privileged instructions
- It poses a real threat to critical systems

Future work

- Use software-based power measurement for side-channel attacks
- Improve bandwidth with better distinguishers for memory power states

Code available at `www.ime.usp.br/~tpaiva`