InkTag

Secure Applications on an Untrusted Operating System

Owen S. Hofmann, Sangman Kim, Alan M. Dunn,
Michael Z. Lee, Emmett Witchel

ASPLOS 2013

Motivation

- OS: untrustworthy “root of trust”
- Verification is easier than implementation
- . . . even more so with paraverification

Design

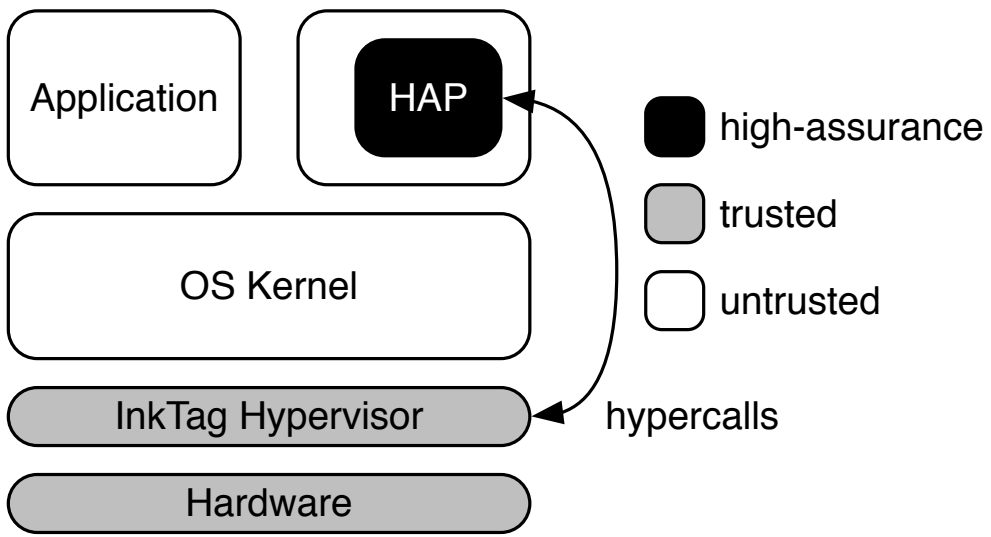
Goals

- Protect applications from untrusted OS
- Secure use of (a subset of) OS services
- Sharing of data among mutually trusting applications
- Fine-grained and flexible access control

Design

Goals

- Protect applications from untrusted OS
- Secure use of (a subset of) OS services
- Sharing of data among mutually trusting applications
- Fine-grained and flexible access control



Building Blocks

Object

- File or memory region
- Comprised of $\mathbb{S}$-pages
- 64-bit object identifier (*OID*)

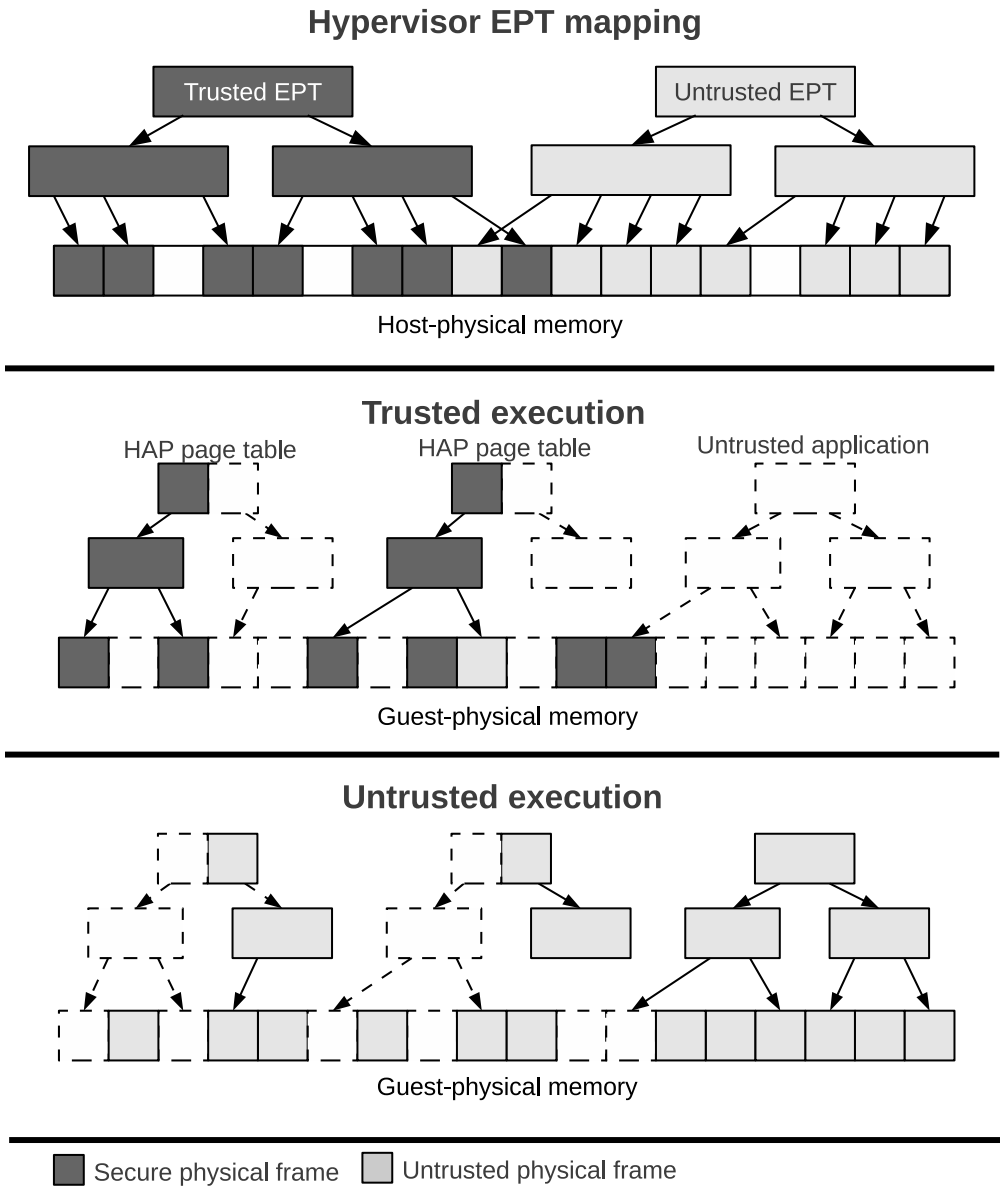
$\mathbb{S}$-page

- Data (4 kB on x86)
- Metadata: <OID, offset>, hash, crypto IV

Nested Paging

- PT: guest-virtual $\rightarrow$ guest-physical (OS)
- EPT: guest-physical $\rightarrow$ host-physical (Hypervisor)
- Independent EPTs for un-/trusted execution

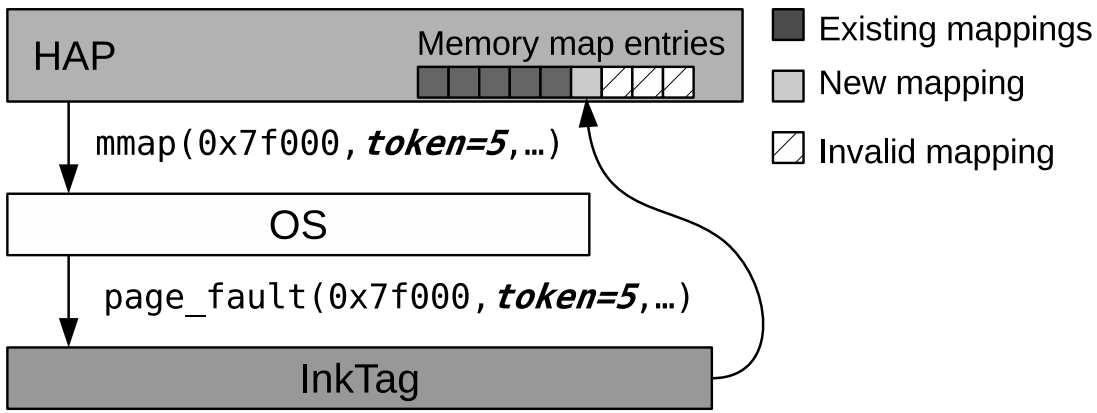
Secure Memory Management



Paraverification

HAP

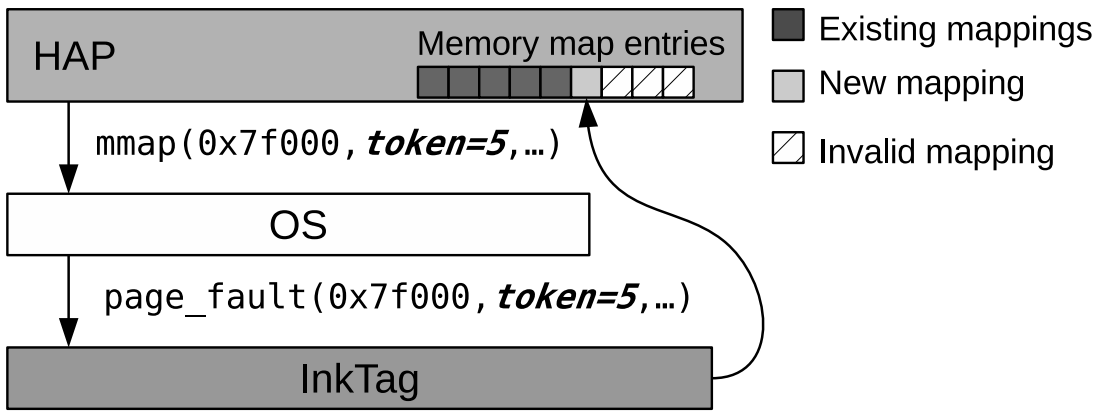
- Maintains array: <address range, OID, offset>
- Hands *token* to OS with each mapping request



Paraverification

HAP

- Maintains array: <address range, OID, offset>
- Hands *token* to OS with each mapping request



Hypervisor

- OS provides page table update + token
- Look up OID associated with virtual address using token
- Check access permissions and integrity of frame

Access Control

Attribute

- Hierarchically composed string (`.user.alice`)
- Attached to HAPs, kept across `fork` and `exec`
- Access control lists on OIDs (read, write, modify) and attributes (add, modify)

Access Control

Attribute

- Hierarchically composed string (`.user.alice`)
- Attached to HAPs, kept across `fork` and `exec`
- Access control lists on OIDs (read, write, modify) and attributes (add, modify)

Namespace

- Attribute used to model directory (`.ns.etc`)
- OID = hash(namespace + file name)
- HAP needs namespace in its attribute list for file creation

Access Control

Attribute

- Hierarchically composed string (`.user.alice`)
- Attached to HAPs, kept across `fork` and `exec`
- Access control lists on OIDs (read, write, modify) and attributes (add, modify)

Namespace

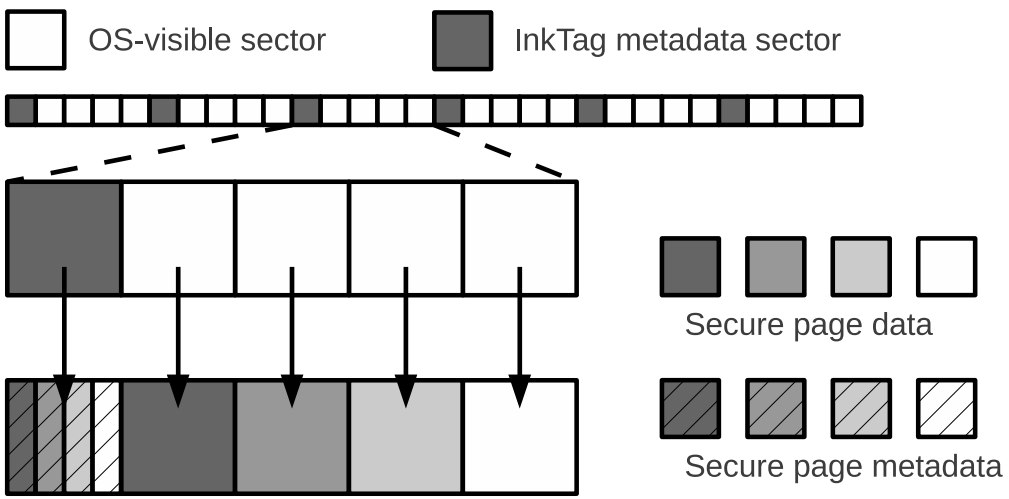
- Attribute used to model directory (`.ns.etc`)
- OID = hash(namespace + file name)
- HAP needs namespace in its attribute list for file creation

HAP startup

- Hypercall: OID of binary + memory layout
- Hypervisor sets up HAP context and adds (`.bin.<oid>`)

Storage

- Metadata interleaved with $\mathbb{S}$-pages
- OS presented with virtual disk lacking metadata blocks
- Keep two hashes for $\mathbb{S}$-pages during update



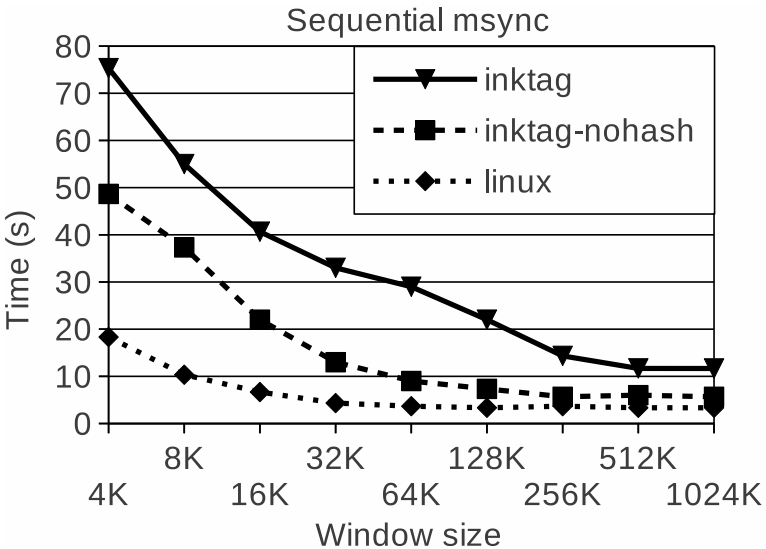
Prototype

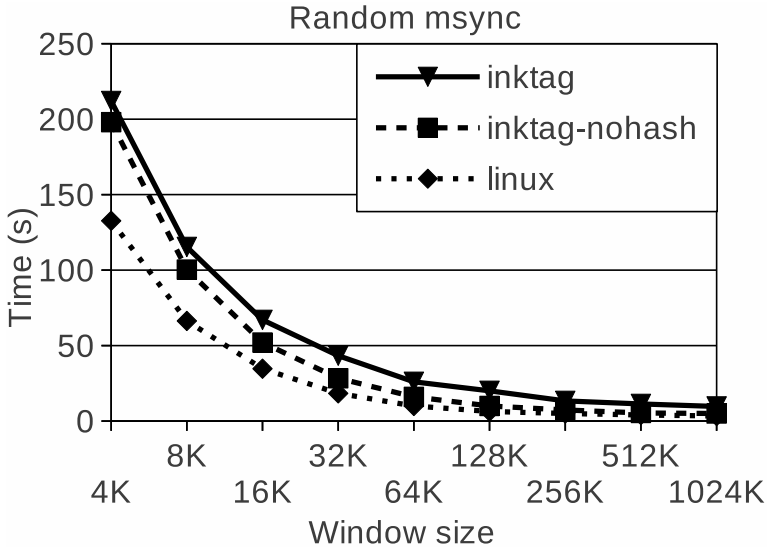
- Linux 2.6.36 + extension to KVM
- QEMU block driver for virtual disk
- `libinktag` (2000 SLOC) wrapped by `libC`

LMbench

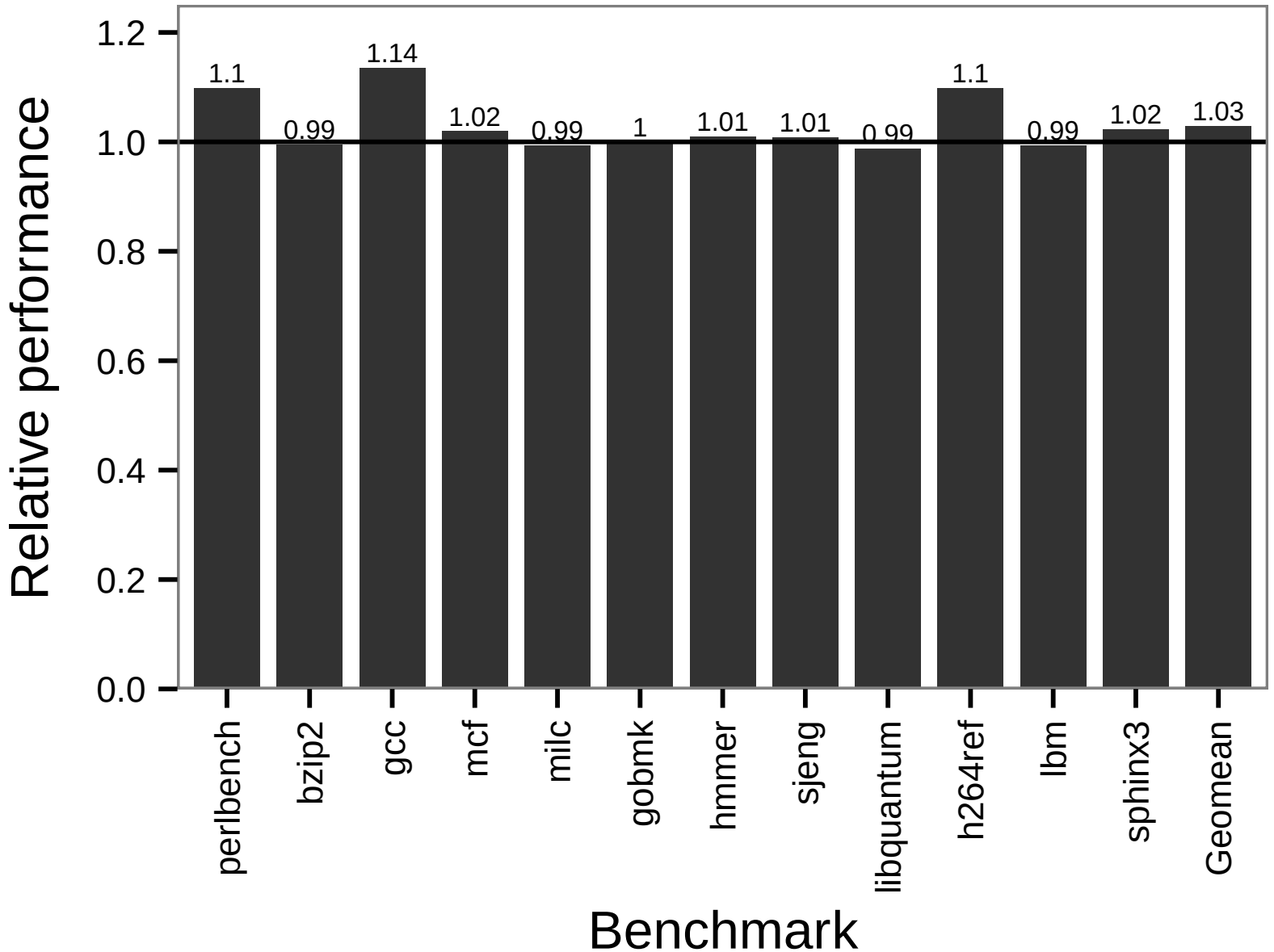
	Linux	InkTag	Overhead
null	0.04	2.23	55.80×
open/close	0.87	6.90	7.95×
ctxsw 2p/0k	0.71	1.01	1.41×
File create	5.46	12.92	2.36×
File delete	3.40	7.56	2.23×
mmap	4059.20	40360.00	9.94×
pagefault	0.89	6.68	7.50×
fork	99.00	567.80	5.74×
fork+exec	290.60	882.60	3.04×

Storage


Sequential msync
Time (s)
80
70
60
50
40
30
20
10
0
inktag
inktag-nohash
linux
4K
8K
16K
32K
64K
128K
256K
512K
1024K
Window size


Random msync
Time (s)
250
200
150
100
50
0
inktag
inktag-nohash
linux
4K
8K
16K
32K
64K
128K
256K
512K
1024K
Window size

SPEC 2006 (C only)



Benchmark II
Large Applications

	Linux	InkTag
Apache latency	195 ms	220 ms (1.13×)
Apache throughput	462.42 req/s	453.93 req/s (1.02×)
Dokuwiki throughput	13.6 req/s	8.83 req/s (1.54×)

	Apache		DokuWiki	
	Linux	InkTag	Linux	InkTag
Check hash	-	209	-	2,911,649
Check zero hash	-	57	-	2,893,517
Update hash	-	82	-	1,029
EPT fault	689	1,131	10,668	78,055
VM-exit	171,145	1,217,042	138,801	11,216,363