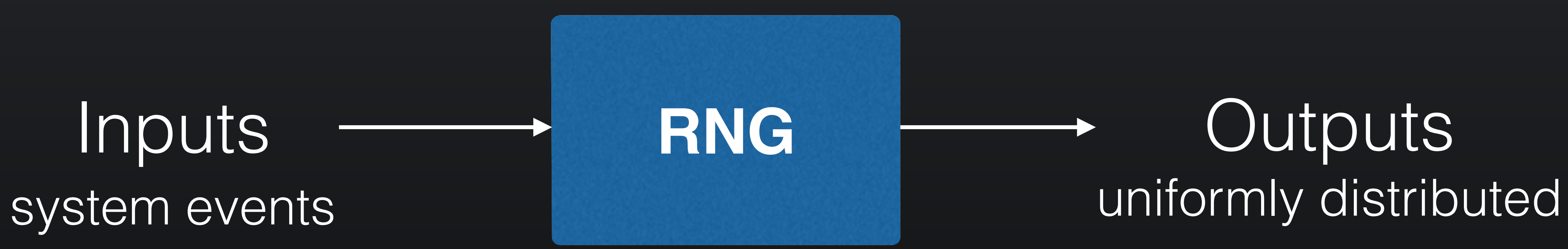
Not-So-Random Numbers in Virtualized Linux and the Whirlwind RNG

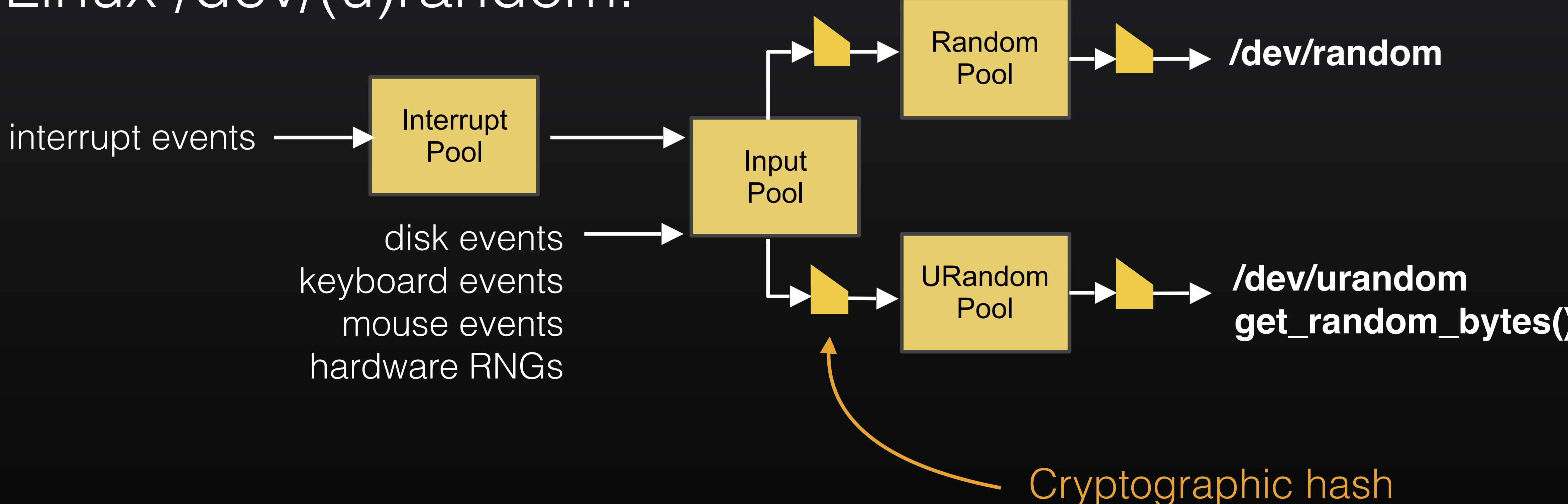
Adam Everspaugh, Yan Zhai, Robert Jellinek,
Thomas Ristenpart, Michael Swift
University of Wisconsin — Madison

Random Number Generators



Random Number Generators

Linux /dev/(u)random:



Random Number Generators

Previous Analyses (mostly showing failures)

Cryptanalysis of Windows RNG [DGP07]

Factorable RSA Keys [HDWH12]

/dev/random not Robust [DPRVW13]

Linux RNG [GPR08]

Linux RNG Revisited [LRSV12]

Taiwan National IDs [BCCCHLS13]

RNGs in Virtualized Environments

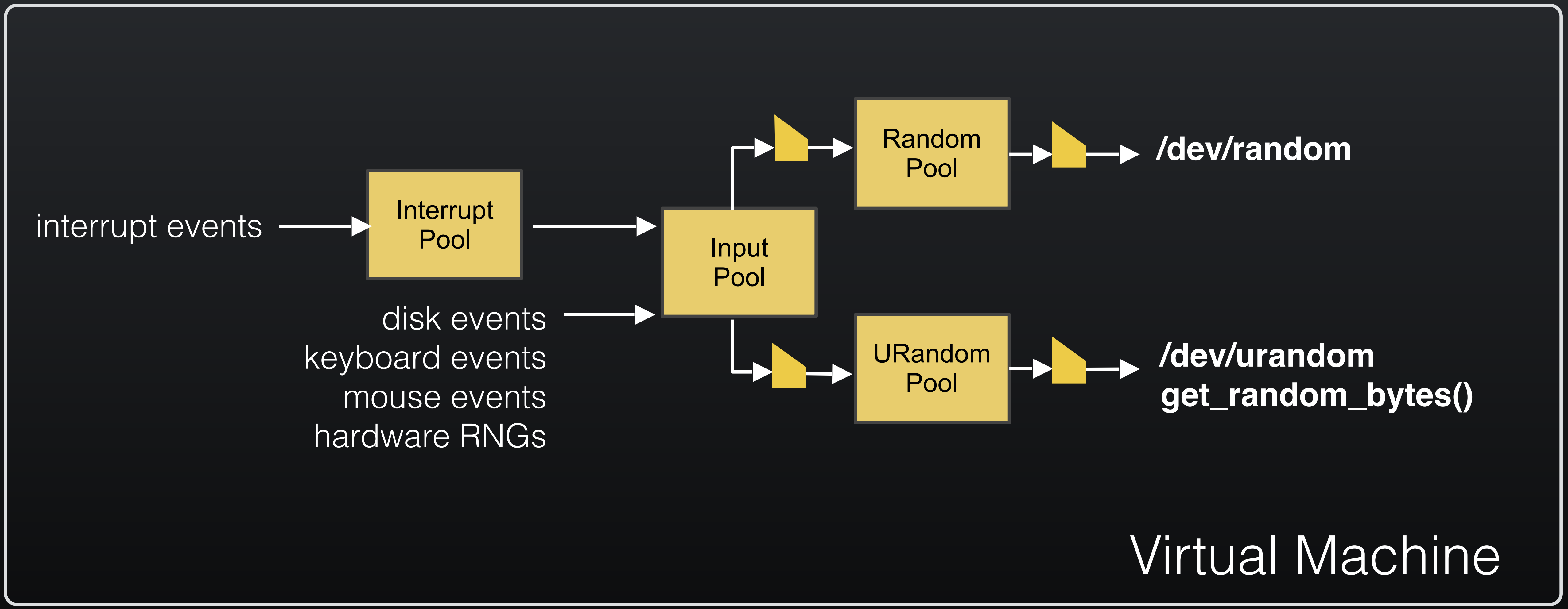
1. Desktop virtualization, data center virtualization, and cloud computing are increasingly popular

2. RNGs designed without virtualization in mind (1990s)

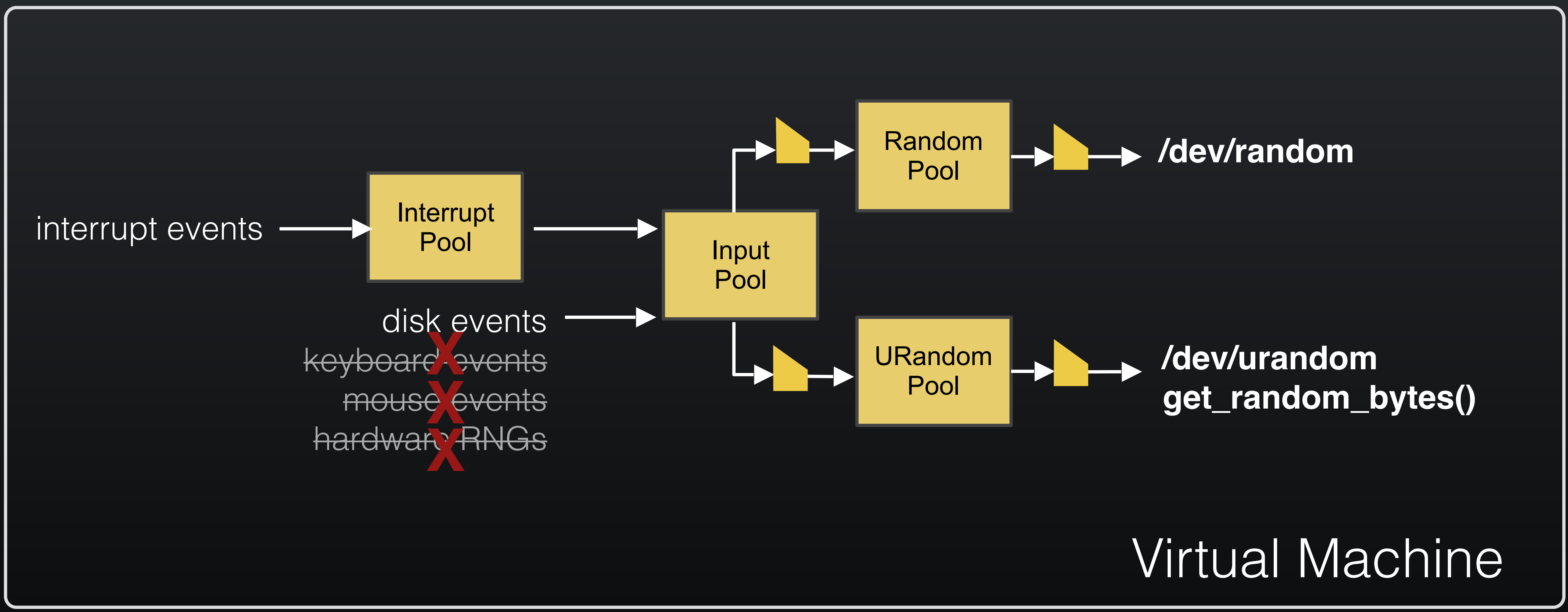
Are system RNGs secure in virtualized environments?

Amazon EC2 **Rackspace** **Microsoft Azure**

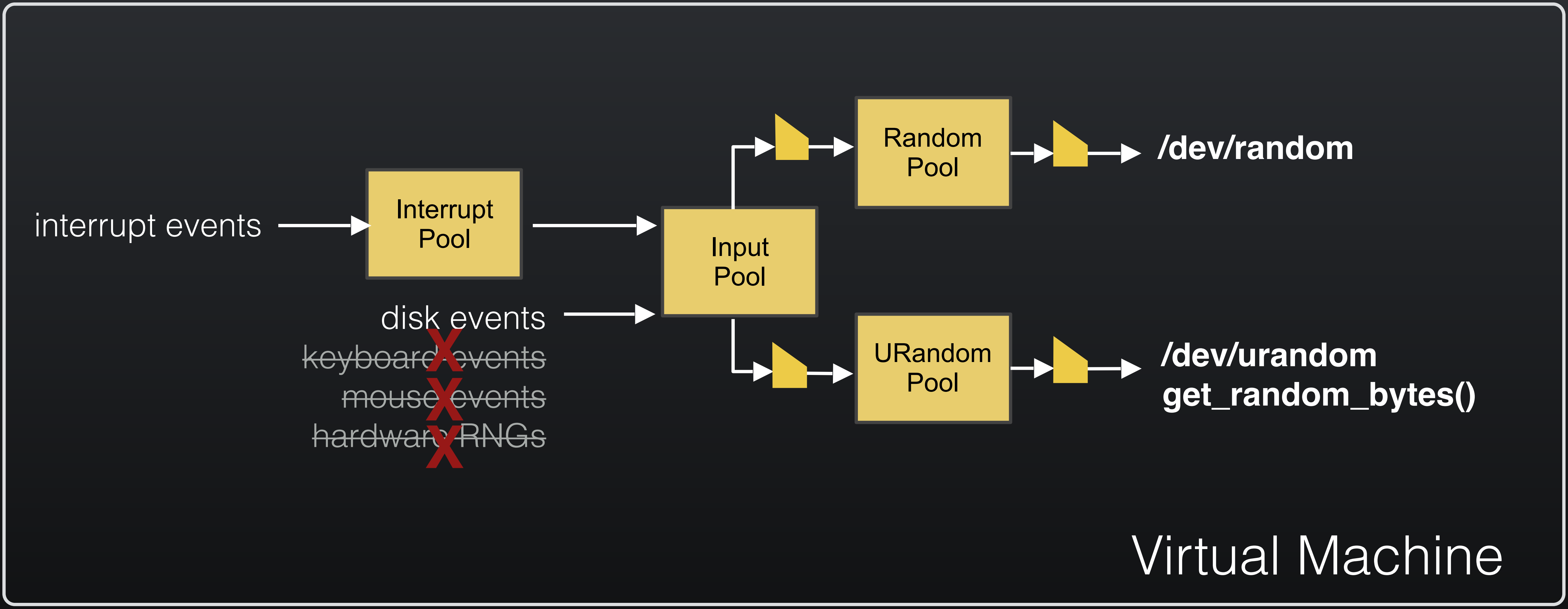
RNGs in Virtualized Environments



RNGs in Virtualized Environments



RNGs in Virtualized Environments



Folklore concerns regarding security:

1. Do full-memory snapshots cause problems for system RNGS? [Garfinkel, Rosenblum 05] [Ristenpart, Yilek 10]

2. Are input sources entropy-poor inside a virtual machine? [Stamos, Becherer, Wilcox 09]

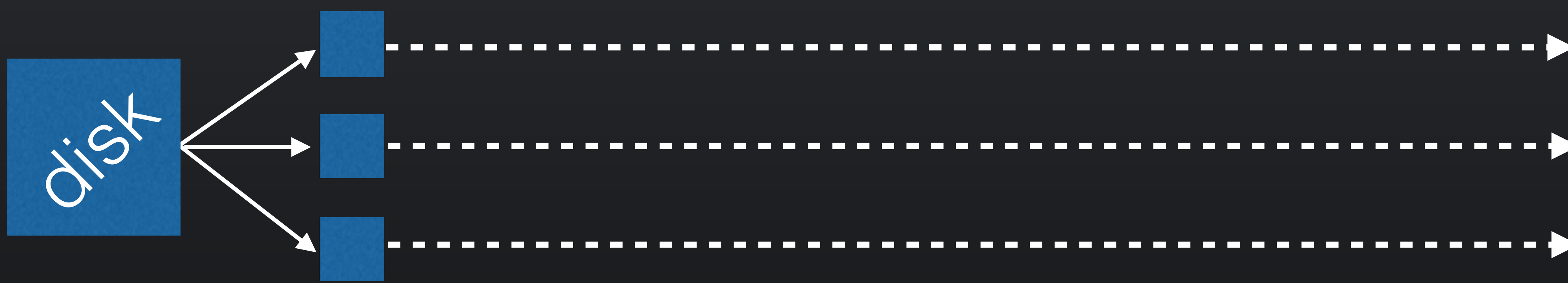
Our Contributions

- First study of system RNGs in modern virtualized settings
- Snapshots cause problems? → YES

  Bad RSA keys from OpenSSL
- Entropy-poor inputs? → NO
- New clean-slate RNG design → Whirlwind

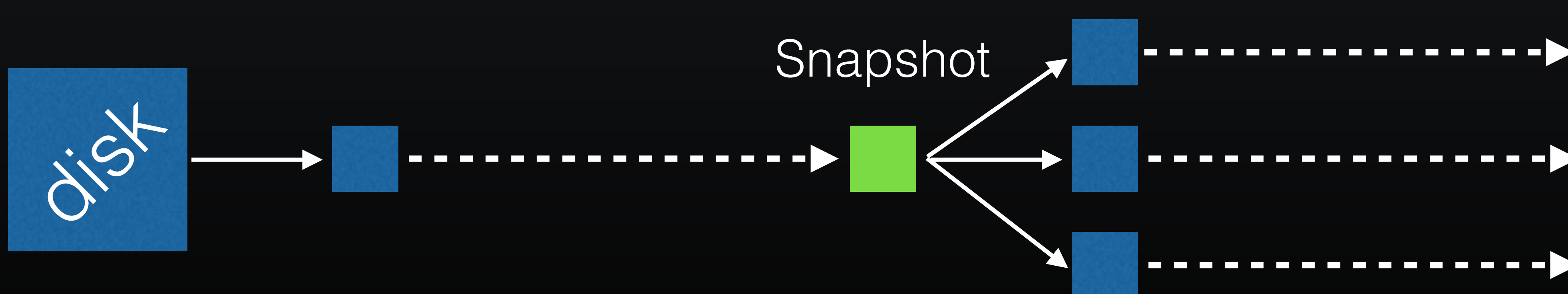
VM Use Cases

Boot-from-image



Amazon EC2 Rackspace Microsoft Azure

Snapshot-Reset

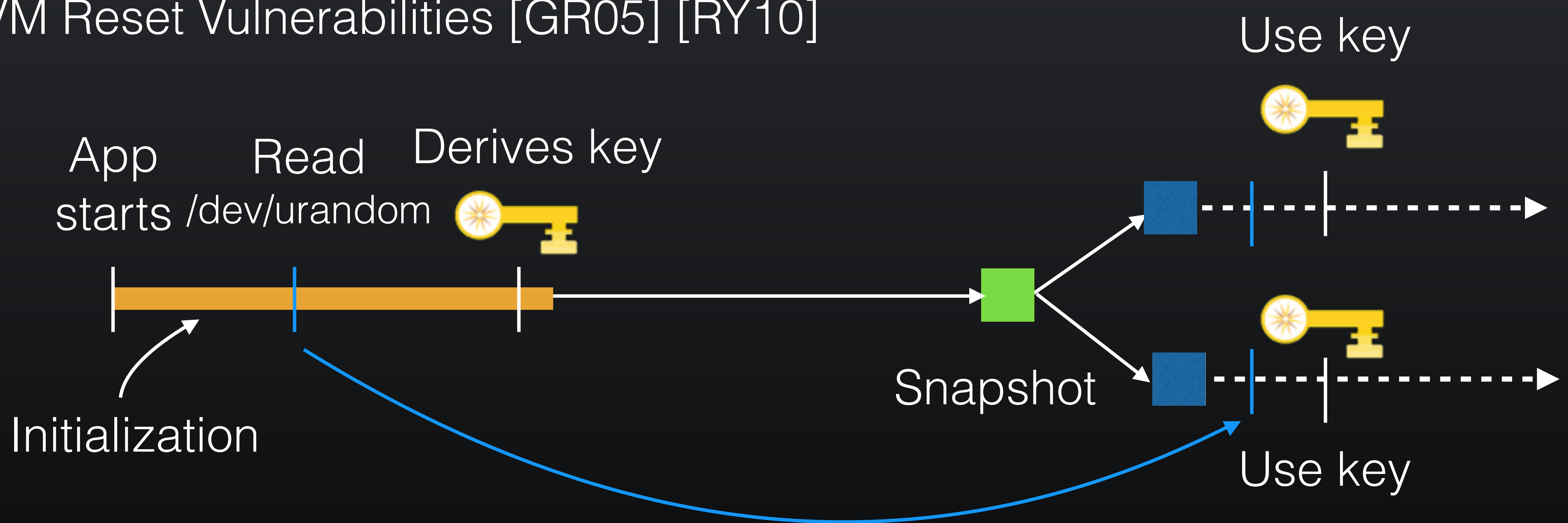


Resumption

vmware Xen Parallels

Security Problems with VM Resets

VM Reset Vulnerabilities [GR05] [RY10]



[RY10] Suggested countermeasure:
Narrow gap between deriving and using random numbers

Are system RNGs reset secure?

Linux RNG ***Not*** Reset Secure

One of our experiments:

- Boot VM in Xen, idle for 5 minutes
- Start measurement process, capture snapshot
- Resume from snapshot,
  read 512-bits from /dev/urandom every 500 us

Repeat for 8 distinct snapshots
Do 20 resumptions/snapshot

/dev/urandom outputs after resumption

Reset 1	Reset 2	Reset 3
21B8BEE4	21B8BEE4	21B8BEE4
9D27FB83	9D27FB83	9D27FB83
6CD124A6	6CD124A6	6CD124A6
E8734F71	E8734F71	E8734F71
111D337C	111D337C	111D337C
1E6DD331	1E6DD331	1E6DD331
8CC97112	8CC97112	8CC97112
2A2FA7DB	2A2FA7DB	2A2FA7DB
DBBF058C	DBBF058C	DBBF058C
26C334E7	26C334E7	26C334E7
F17D2D20	F17D2D20	45C78AE0
CC10232E	CC10232E	E678DBB2
...	...	...

/dev/urandom outputs after resumption

Linux RNG is ***not*** reset secure:
7/8 snapshots produce mostly identical outputs

Reset 1	Reset 2	Reset 3
1E6DD331	1E6DD331	1E6DD331
8CC97112	8CC97112	8CC97112
2A2FA7DB	2A2FA7DB	2A2FA7DB
DBBF058C	DBBF058C	DBBF058C
26C334E7	26C334E7	26C334E7
F17D2D20	F17D2D20	45C78AE0
CC10232E	CC10232E	E678DBB2
...	...	...

Reset insecurity and applications

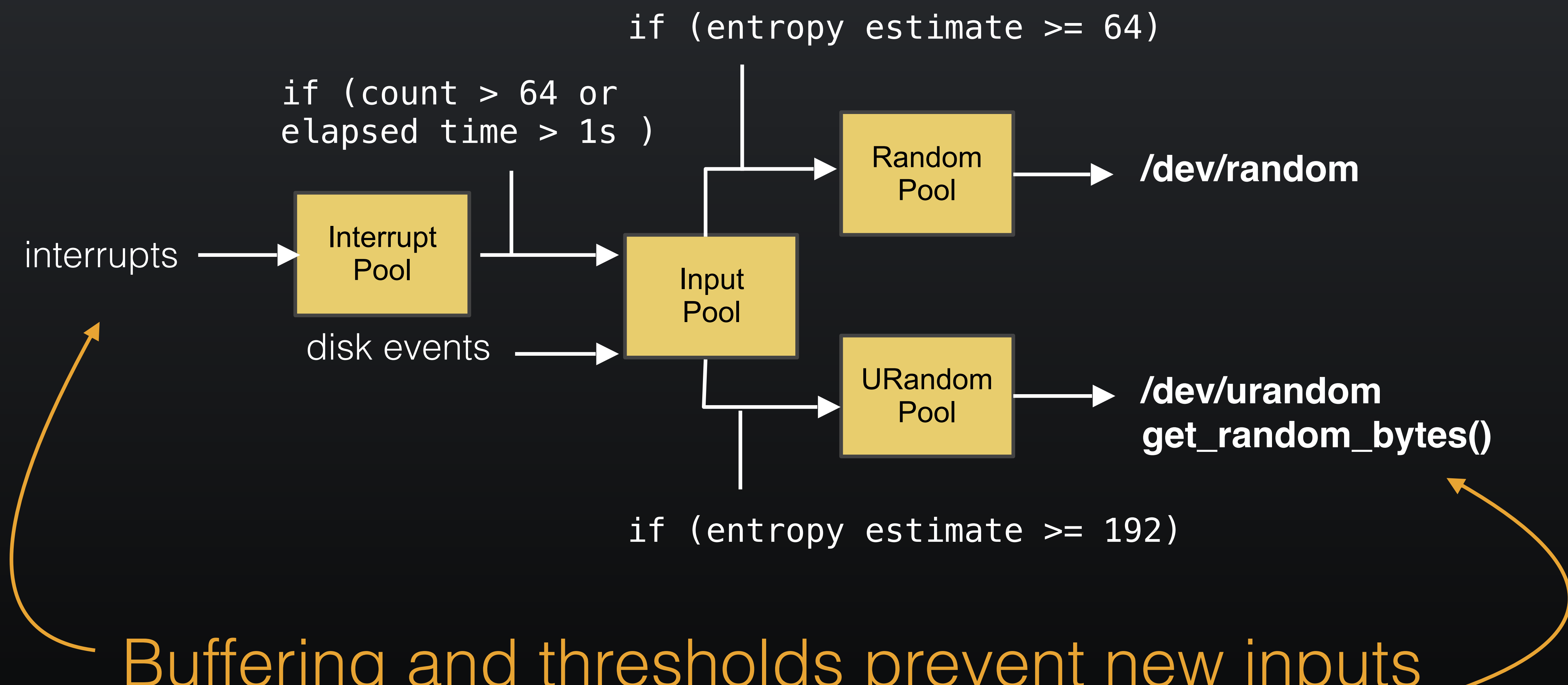
Generate RSA key on resumption:

```
openssl genrsa
```

30 snapshots; 2 resets/snapshot (ASLR Off)

- 27 trials produced identical private keys
- 3 trials produced unique private keys

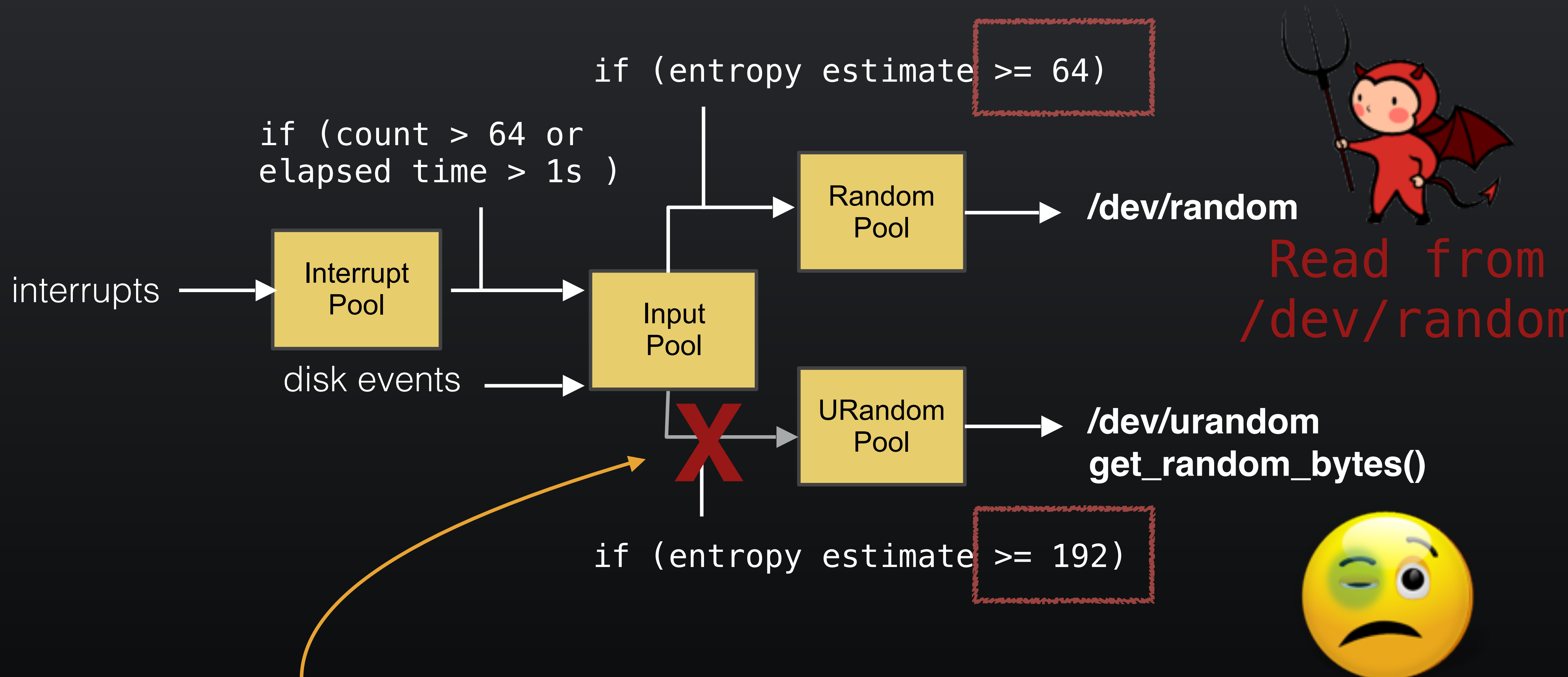
Why does this happen?



Buffering and thresholds prevent new inputs from impacting outputs

Linux /dev/(u)random

An even more extreme case: Entropy Starvation Attack



Prevents new inputs from ***ever*** reaching /dev/urandom

No entropy in /dev/urandom outputs

Reset Vulnerabilities Effect Other Platforms

FreeBSD
/dev/random produces identical output stream
Up to 100 seconds after resumption

Microsoft Windows 7
Produces repeated outputs indefinitely

`rand_s` (stdlib)
`CryptGenRandom` (Win32)
`RngCryptoServices` (.NET)

Reset vulnerabilities summary

Using snapshots can compromise security of applications relying on system RNGs

Many different VM platforms / operating systems

Infrastructure-as-a-service providers don't yet support full-memory snapshots, but could in the future.

Our Contributions

- First study of system RNGs in modern virtualized settings
- Snapshots cause problems? → **YES**
- Entropy-poor inputs?
- New clean-slate RNG design — Whirlwind

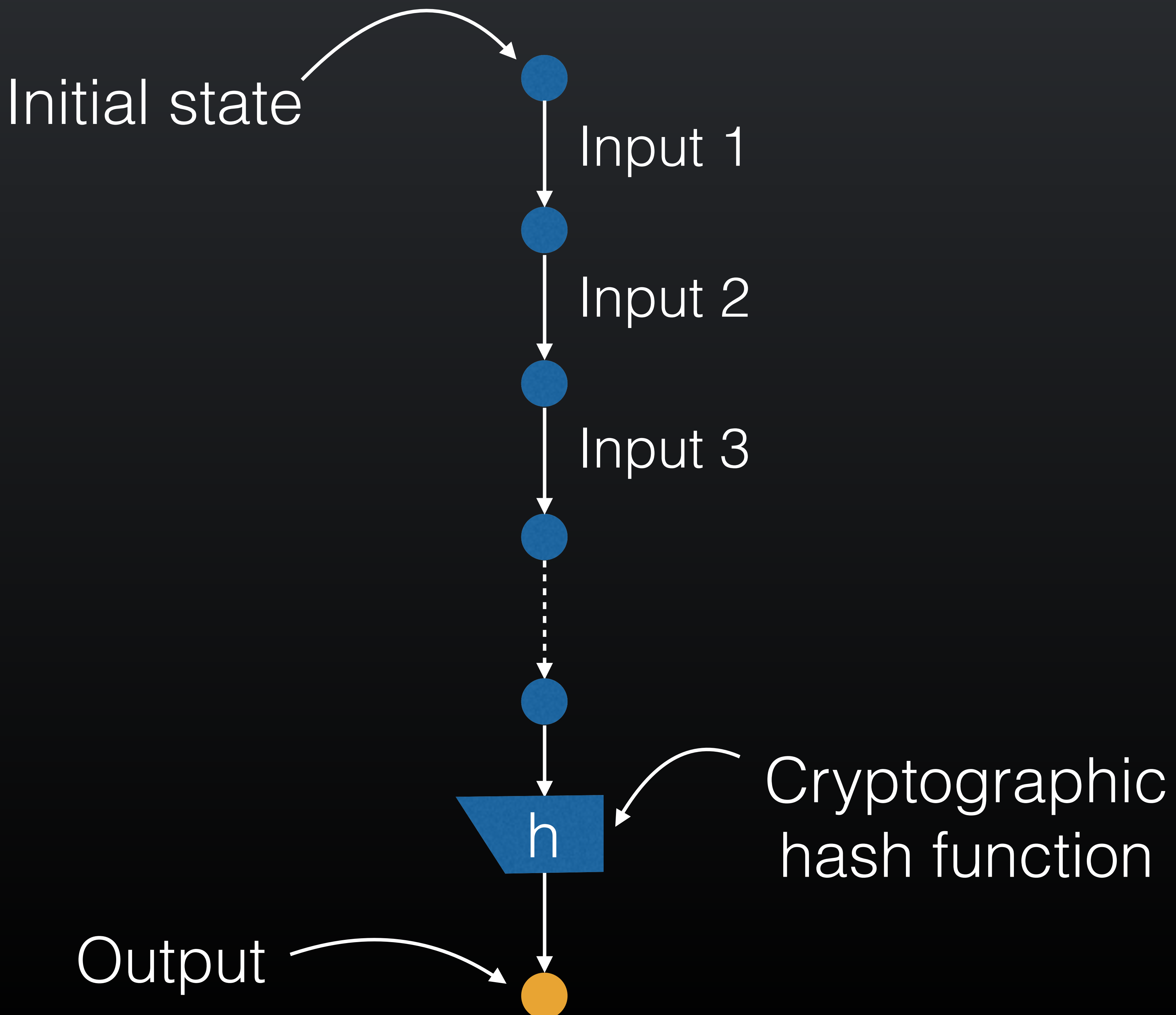
Our Contributions

- First study of system RNGs in modern virtualized settings
- Snapshots cause problems? → YES
- Entropy-poor inputs?
- New clean-slate RNG design — Whirlwind

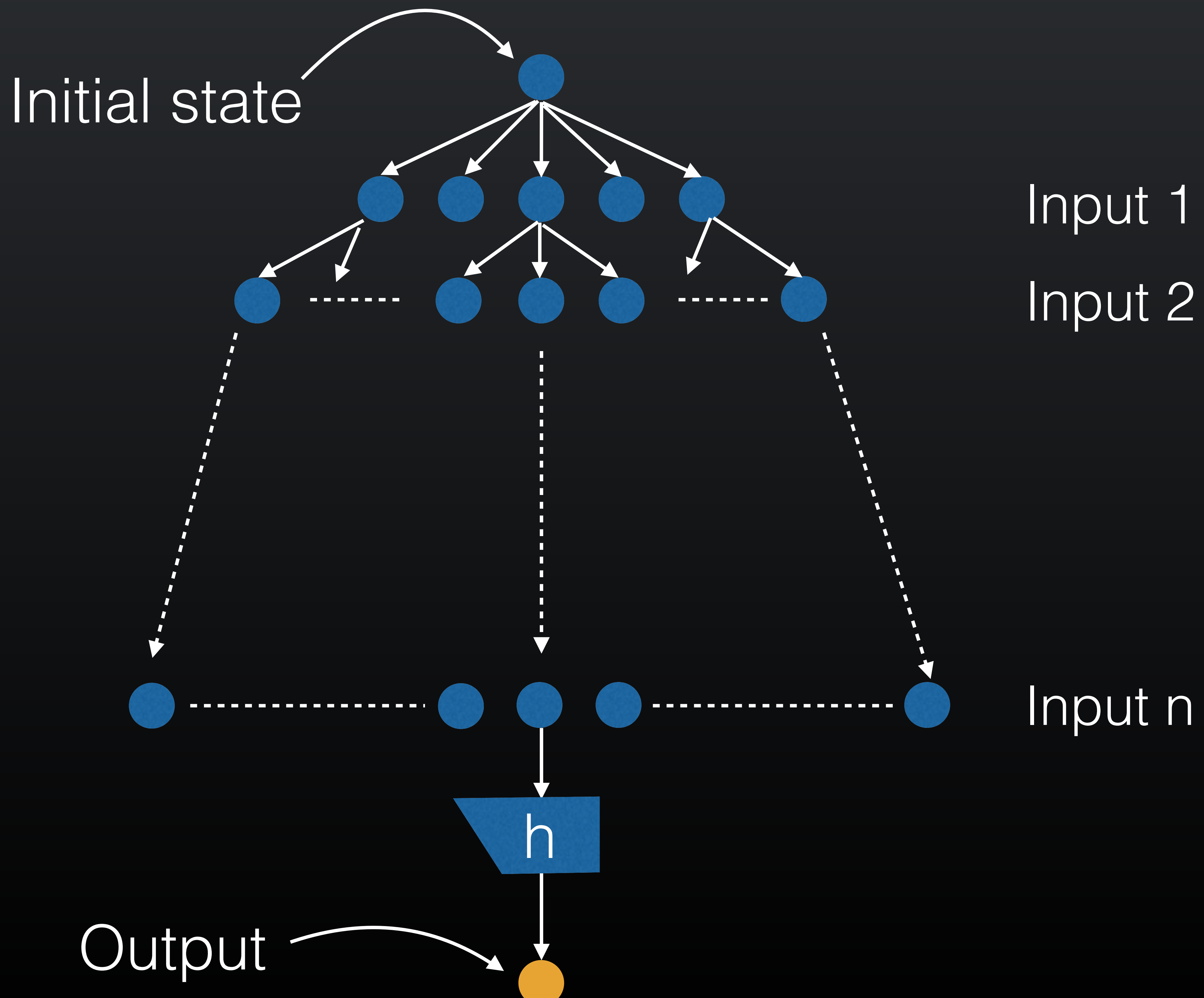
Estimating Input Entropy

- Instrumented Linux RNG
- Collected all inputs, outputs on boot
- Gathered data from: native, Xen, VMware, and EC2
- Statistical hypothesis testing to estimate entropy of each input
- Use input entropy to estimate security of each output

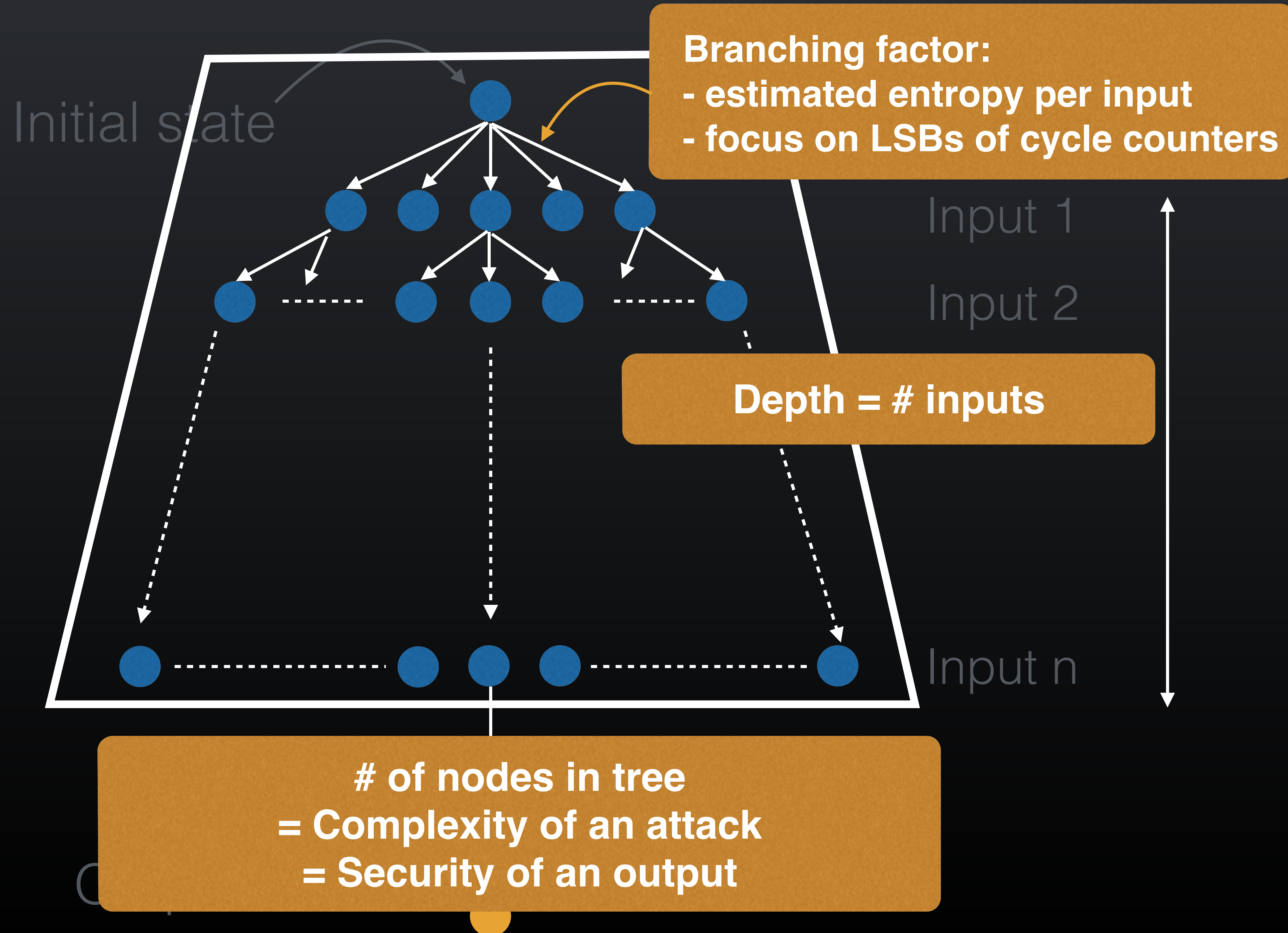
RNG Operation



Attacker's View



Attacker's View



Results: Boot Security

No inputs before first output:
constant value

Output #	Native	Xen	VMware	EC2
1	0	0	0	0
2	129	129	784	134
15	129	1024	1024	1024

Search space ($\log_2$) required of adversary for outputs of Linux /dev/(u)random during boot

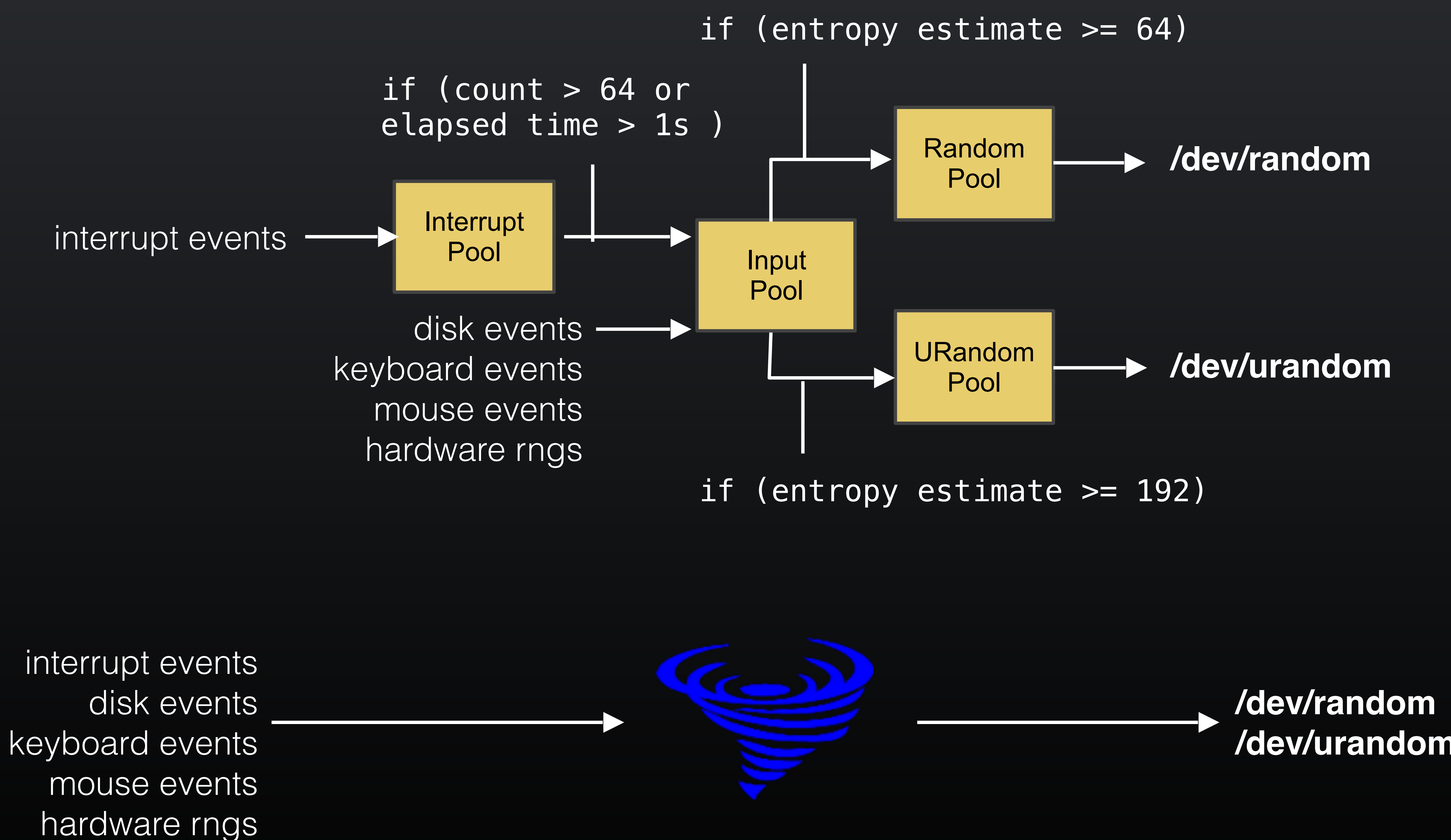
Our Contributions

- First study of system RNGs in modern virtualized settings
- Snapshots cause problems? → YES
- Entropy-poor inputs? → **NO**
- New clean-slate RNG design — Whirlwind

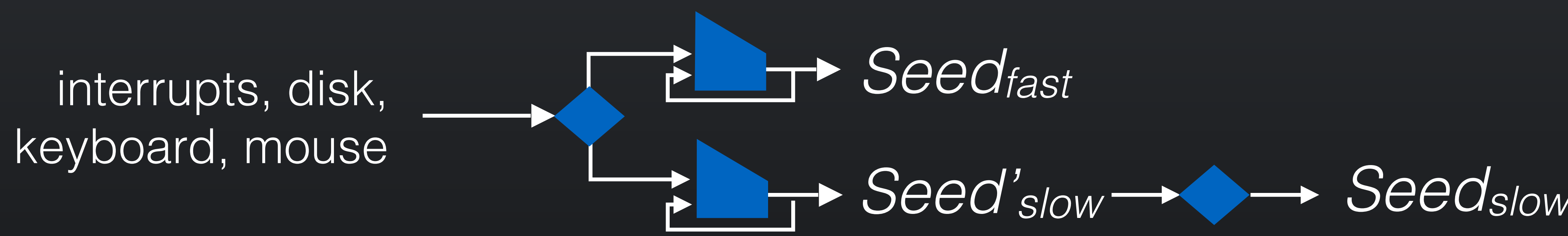
Our Contributions

- First study of system RNGs in modern virtualized settings
- Snapshots cause problems? → YES
- Entropy-poor inputs? → NO
- New clean-slate RNG design — Whirlwind

Same interfaces - new design



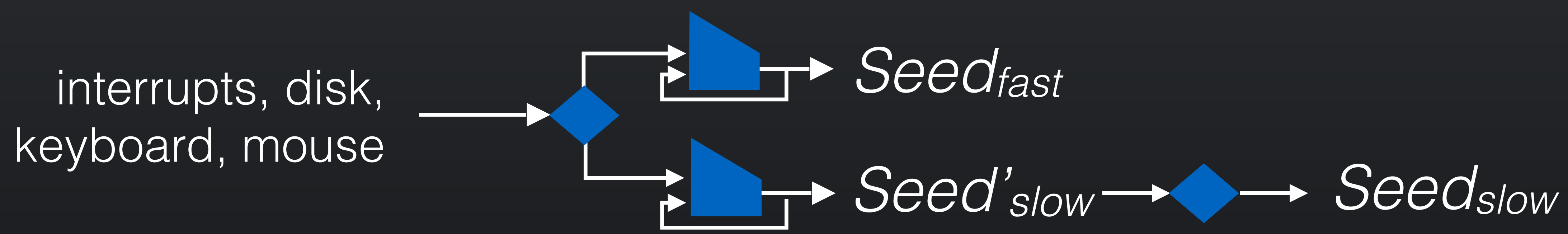
Whirlwind RNG Design



Online hashing input process [DPRVW13]
Two-state structure (fast-slow) borrowed from Yarrow (FreeBSD) [SF99]

Seed$_{fast}$ ensures secure state change *rapidly*
Seed$_{slow}$ prevents checkpoint (aka tracking) attacks [SF99]

Whirlwind RNG Design



On boot: securely initialize RNG with large number of inputs

On snapshot resumption: inject randomness from Xen hypervisor

Whirlwind RNG

1. Drop-in replacement for legacy Linux RNG
2. Simple
3. Cryptographically sound
4. First output is not predictable
5. Reset security *by design*

Conclusions

- Linux, FreeBSD, and Windows are vulnerable on snapshot resumption
- First output of Linux /dev/(u)random on boot is a constant value
- Virtual settings have sufficient entropy
- Whirlwind RNG gives reset security by design

More information:

http://pages.cs.wisc.edu/~ace/