Undeniably Plausible Plausibly Deniable Storage

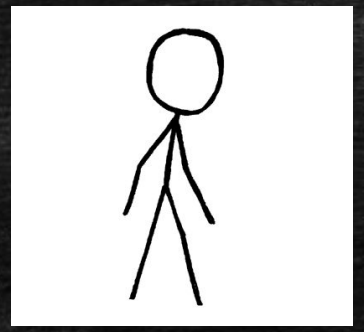
By: Swaminathan Ramesh and Dr. Ryan Henry

Outline

- Plausibly deniable filesystems (PDFS): what are they and why are they useful?
- How plausible deniability is defined in existing literature
  - HIVE
  - DataLair
- Shortcomings in existing definitions
- How can we fix these definitions?
  - Defining filesystem and filesystem operations
  - Plausibility as simulatability

What are plausibly deniable filesystems?

Alice

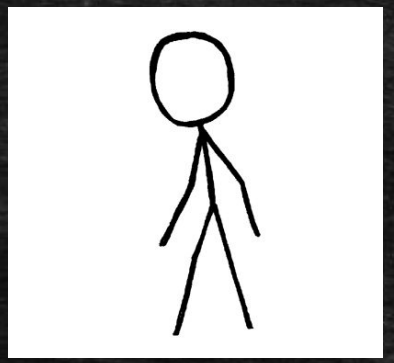


Proof of kicking puppies

Bob

Pictures of hugging puppies

Oscar



Proof of kicking puppies

Why are PDFS useful?

- Ensuring privacy in data storage settings like:
  - Journalists
  - Whistleblowers
  - Human rights activists

Formal model in literature - DataLair

- What is DataLair?
  - PDFS proposed by Sion et al. in CCS 2017
  - Uses write-only ORAM
  - Proposes PD-CPA to capture plausible deniability

Formal model in literature - DataLair

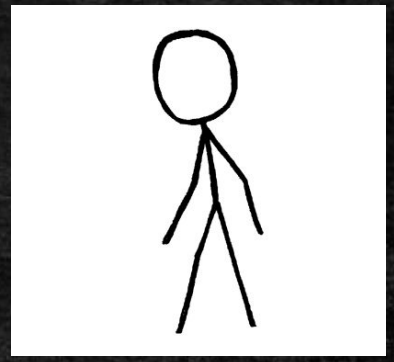
- Adversary model and capabilities
  - PPT adversary
  - Multi-snapshot

Formal model in literature - DataLair

- Security definition – PD-CPA(n,m). Security parameter – 1^r

Attacker

Challenger



1. Sends storage device

3. Sends public key

4. Sends P0, P1

5. Executes Pb; Sends snapshot

6. Outputs b'

2. Creates public and private keys and volumes; tosses a fair coin (b)

Attacker wins if b' == b

Formal model in literature - HIVE

- What is HIVE?
  - PDFS introduced by Blass et al. - CCS 2014
  - Uses write-only ORAM
  - Security notion - $G^{A-E}(n)$

Formal model in literature - HIVE

- Adversary model and capabilities
  - PPT adversary
  - Multi-snapshot

Formal model in literature - HIVE

- Security definition – $G^{A-E}(n)$; Security parameter – 1^n

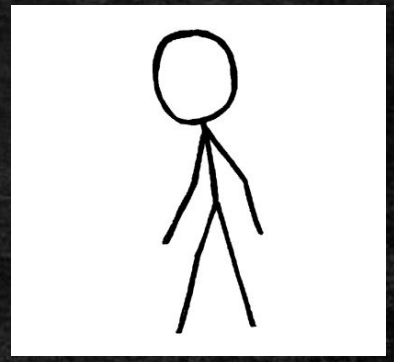
Attacker

1. Sends L

2. Sends initial snapshot

Sends P0, P1

Executes Pb; Sends snapshot



Challenger

Outputs b'

Uses L to create initial state of device; tosses a fair coin (b)

Attacker wins if b' == b

Shortcomings of HIVE and DataLair

- DataLair:
  - Artificial restriction on number of writes to private volumes – construction specific quirk
  - Has exactly one public and private volume
  - Does not talk about filesystem state changes from "reads"

- Common drawbacks:
  - Does not account for partial revelation of private volumes
  - Definitively expose the existence of private volumes
  - Do not explore relation between PDFS and secure deletion

Fixing definitions - 1

- Formal model of a filesystem based on Turing machines
  - Epoch-driven
  - Tapes
  - Operations
    - Traces and access patterns

Fixing definitions - 2

- Plausibility as simulatability

  - Real-world application: OTR deniability in Signal
  - Adversary cannot distinguish between "real" and "ideal" worlds

  - Adversary scenarios:
    1. Explicit knowledge of hidden volumes
    2. No knowledge of hidden volumes but non-simulated transcript
    3. No or partial knowledge of hidden volumes and simulated transcript

Fixing definitions - 3

- Hiding operations and hidden volumes
  - Operation hiding:
    - Adversary has full knowledge of private volumes
    - Can supply operation traces to challenger
    - Cannot distinguish between different snapshots of filesystem based on knowledge of operation traces
  - Volume hiding:
    - Adversary does not know what volumes exist
    - Can specify operation traces
    - Cannot infer existence of hidden volumes from knowledge of snapshots and operation traces

Thank you!