Traffic Analysis Attacks on a Continuously-Observable Steganographic File System

Orr Dunkelman

Ecole Normale Superieure

Joint work with: Carmela Troncoso, Claudia Diaz, Bart Preneel (KU Leuven)

Talk Outline

- What are Steganographic File Systems?
- StegFS
- Attacking Multi-block Files
- Attacking One-Block Files
- Simulation results
- Conclusions

Privacy of your data

- We all have private information – financial data, health information, personal things, etc.
- Today, more and more of this data is stored in a digital manner.
- More and more of out interaction is done on networks (email, social networks, accessing banks/health care providers/insurance, etc.).
- An adversary that listens to your communications can learn quite a lot about you.

Privacy of your data (cont.)

- Besides communication patterns, we store sensitive information on our disks:
  - Bank reports
  - Health record
  - Whistle blowing information
  - Pro-democratic texts in non-democratic regimes.
- We should keep this data from leaking...

First Layer of Protection – Encryption of the Files

- Encryption of the data assures that if the hard disk is read by someone with no access to the secret key cannot obtain any information on the data inside the files.
- The encryption can be done in several layers:
  - Individual files
  - Directory level
  - The entire file system (by the operating system)
  - The entire drive or partition (by the hardware)

Individual files

- Advantages:
  - Most efficient (encrypt only what really matters)
  - Possibly different keys for different files
- Disadvantages:
  - Meta-data (size, name) is clear
  - Easy to identify encrypted files
- Upon update:
  - Either re-encrypt the entire file (not efficient)
  - Re-encrypt a smaller chunk (reveals information about the location of change)

A Whole Directory

- Advantages:
  - Relatively efficient
  - Possibly different keys for different directories
  - Allows for an encrypted directory inside an encrypted directory
- Disadvantages:
  - Easy to identify its existence
- Directory information – to encrypt?
  - If the map of the directory is not encrypted – we can get meta-data on encrypted files
  - If it is encrypted, less likely to recover corrupted directory

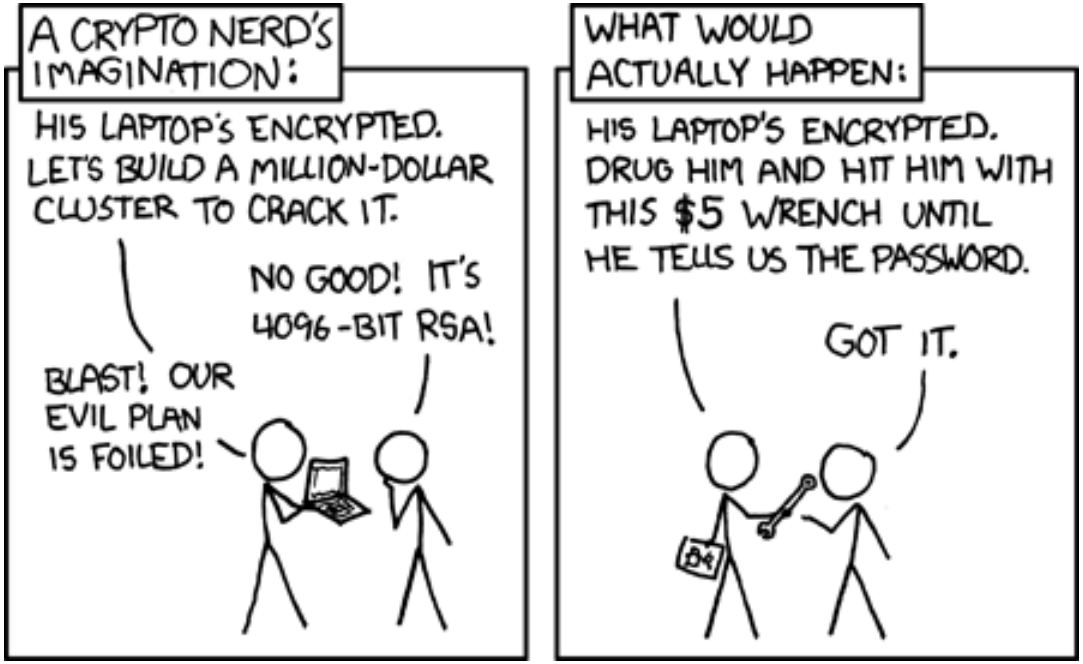
The Entire File System

- Advantages:
  - Very high security (assuming meta-data is also encrypted)
  - User do not need to specify exactly what he wishes to protect
- Disadvantages:
  - Resource consuming (Slowest method)
  - Easy to identify its existence
  - MBR (Master Boot Record) is out in the clear
  - May require human interaction at boot ...
- Should be done carefully, follow standards!
  - IEEE P1619 (has issues, has fixes)

The Entire Drive

- Advantages:
  - Very high security
  - User do not need to specify exactly what he wishes to protect
  - Everything is encrypted if done properly
  - Should be easy now that the Trusted Computing Group agreed what to do
- Disadvantages:
  - Resource consuming (even though encryption done in hardware)
  - May require human interaction at boot (or not secure)

Steganographic File Systems?



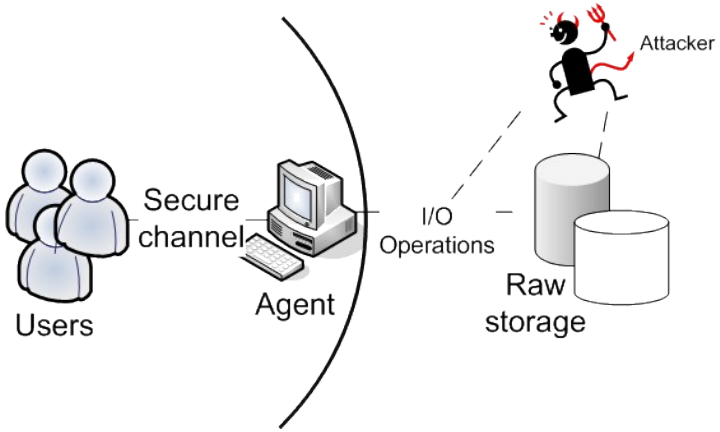
Taken from http://www.xkcd.com/538/

Steganographic File Systems?

- Encryption protects against the unwanted disclosure of information but… does not hide its very existence!
- User can be threatened (physically, economic sanctions,...)
- Steganographic File Systems provide “Plausible deniability”
- Protect the user from compulsion attacks
- Hide the existence of files
  - Different levels of security

What are Steganographic File Systems?

- Protects the user from compulsion attacks
- Hides the existence of files
  - Different levels of security
- Continuously observable: attacker monitors system all the time! (strongest model)

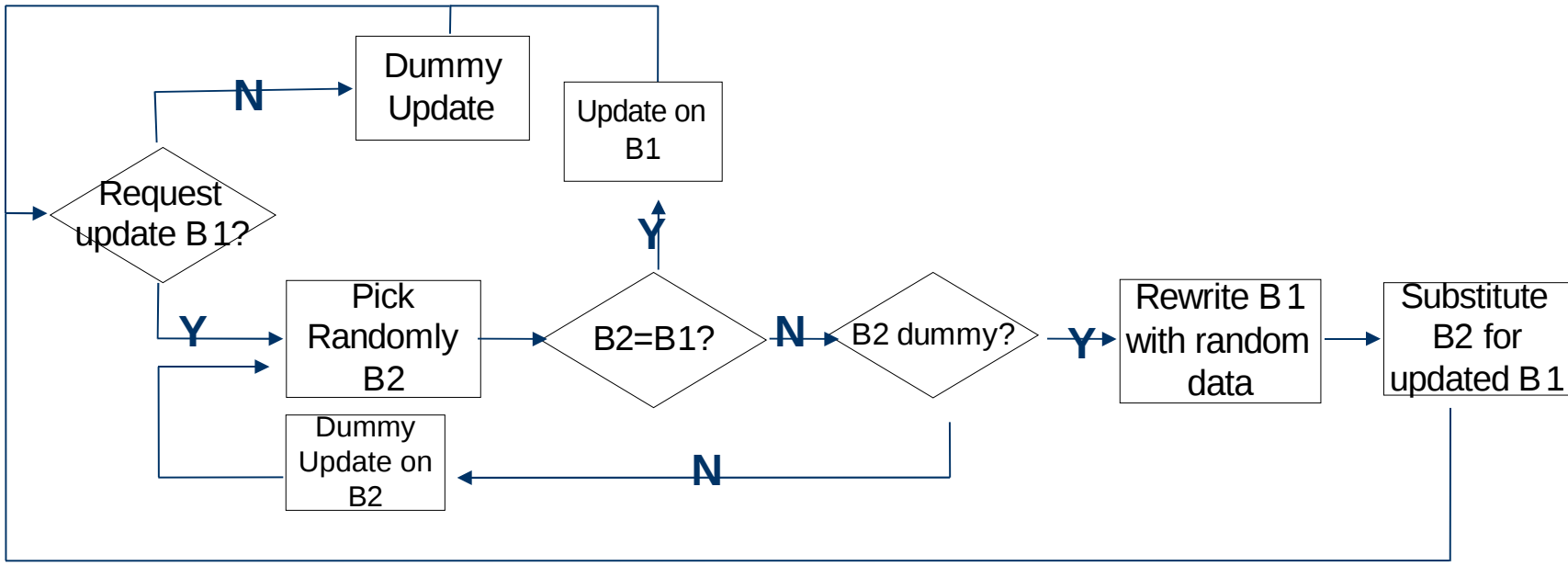


StegFS [Zh+03]

- Proposed by Zhou, Pan &Tan in 2003
  - Bitmap for blocks: free or allocated (real files, dummy files)
  - Files encrypted with a CBC block cipher
  - Key per file
  - Files accessed through headers
- Types of blocks:
  - **File blocks**: contain encrypted user data
  - **Dummy blocks**: free. Contain random data
- Update operations:
  - **Data update**: change content of block
  - **Dummy update**: change IV of CBC

StegFS [Zh+04]: Update Algorithm

- 2004: Algorithm to provide resistance against continuous observation: algorithm for reads and writes
- Update algorithm



StegFS: proof of security

"*For a data update,* ***each block in the storage space has the same probability of being selected to hold the new data****. Hence the* ***data updates produce random block I/Os, and follow exactly the same pattern as the dummy updates****. Therefore, whether there is any data update or not, the* ***updates on the raw storage follow the same probability distribution as that of dummy updates****."*

- All the locations have the same probability of being selected
- BUT:
  - I/Os produced by file updates follow different patterns than dummy updates.
  - The probability distributions of location updates are different depending on user activity.

Traffic analysis attacks on accessed locations!!

What is Traffic Analysis?

- By studying the frequency of the communication or its volume, one can learn many things:
  - Day before the attack on pearl harbour, the Japanese embassy received a (unusually) long transmission
  - Unusual large amount of short options being purchased just before the 11/9 attacks
  - A TCP/IP connection which is being open towards a known hacker address may reveal infection
- Depending on the model, the adversary may tap all communication lines, or even control some of the routers.

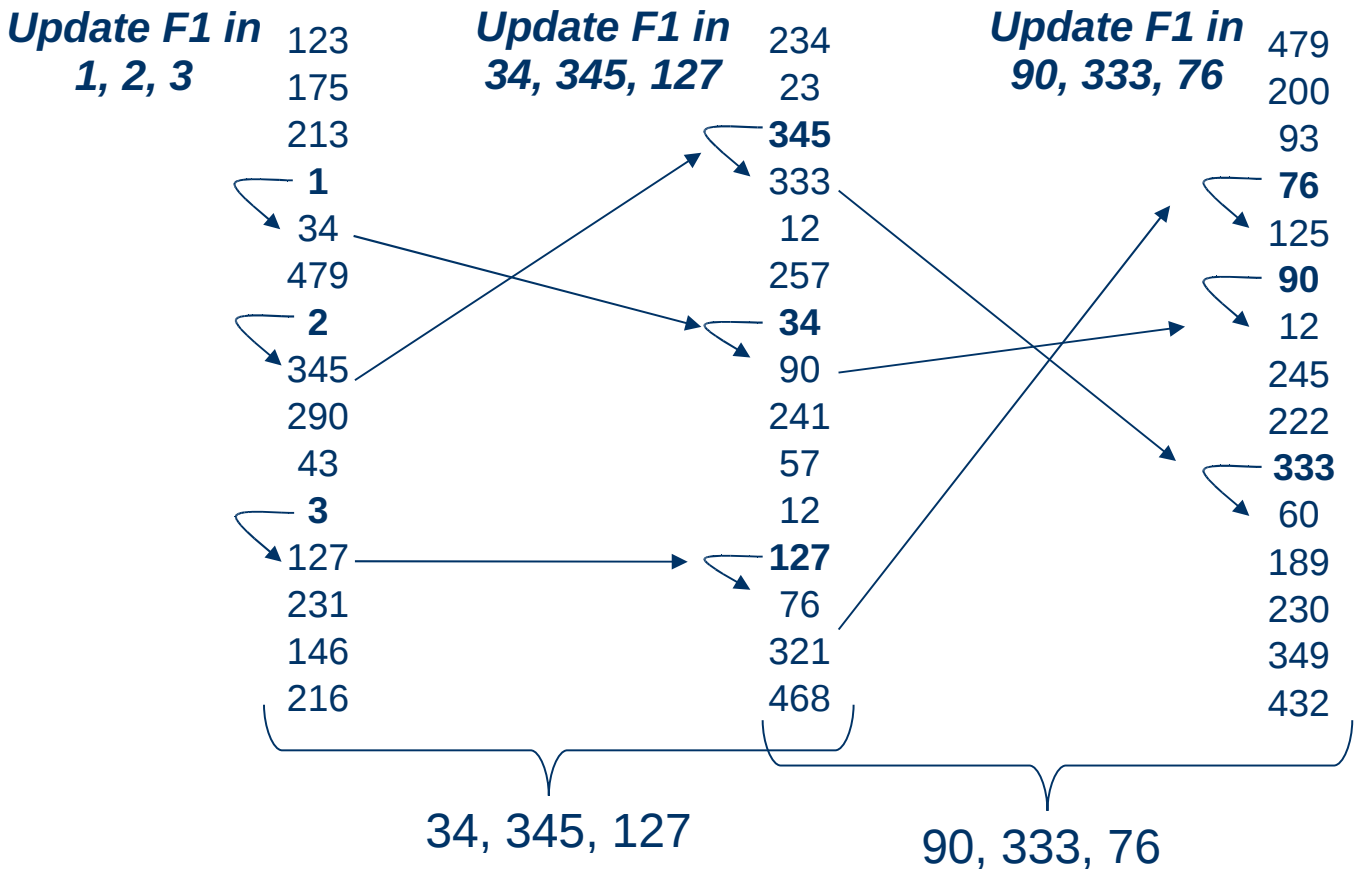
Attacking multi-block files: Update one block

- Pattern (updating B1):
  1. As many dummy updates on data blocks as data B2's are chosen
  2. Overwrite file block B1 with random data
  3. Write dummy block B2 with the updated data

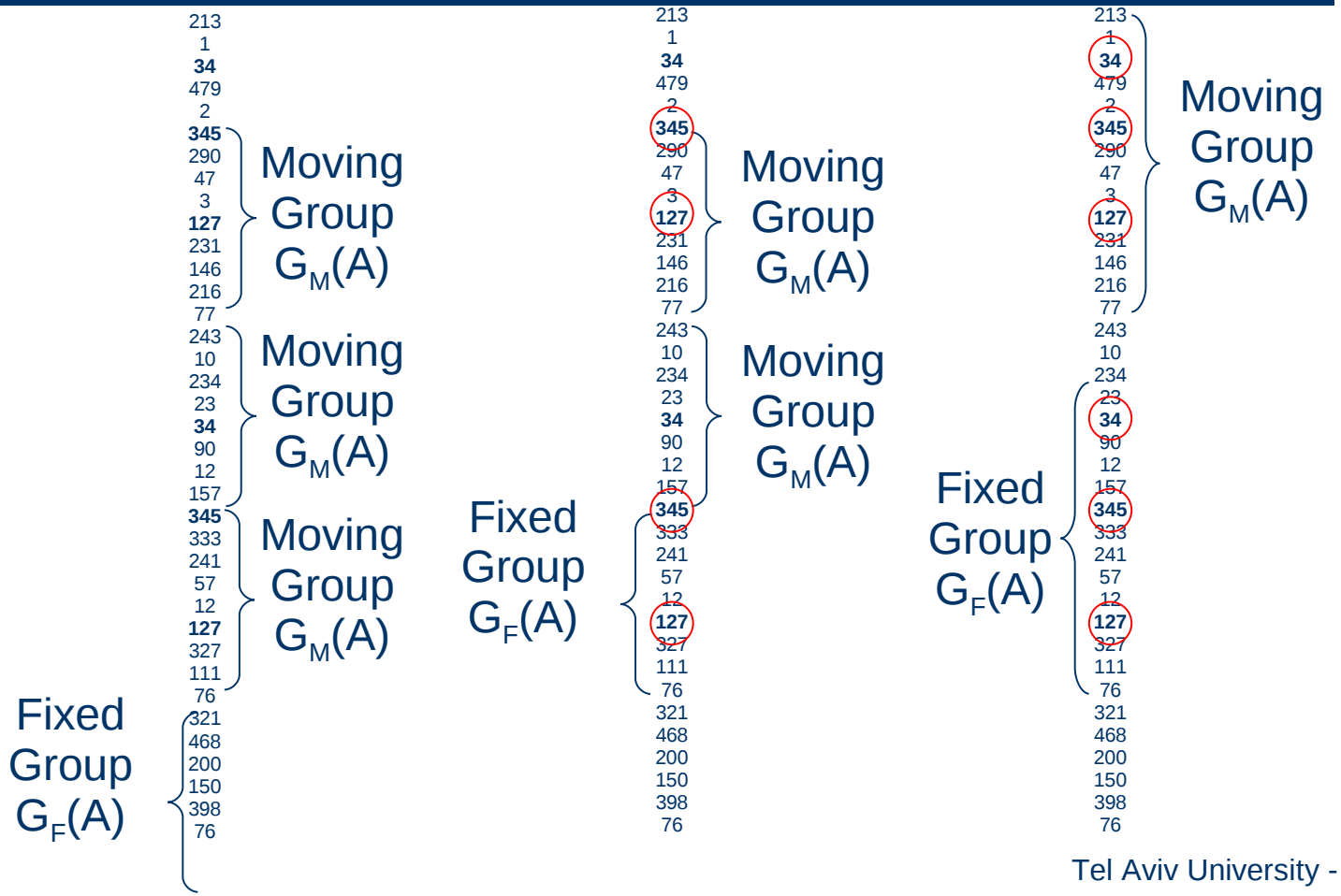
Example: Update block B1=3

Block selected	Access list	Operation performed
290 (F) 47 (F) 127 (D) -	290 47 3 127	Dummy update on data block B2=290
		Dummy update on data block B2=47
		Overwrite file block B1=3 with random data
		Write B2=127 with updated content of B1

Attacking multi-block files: Update a file



Attacking multi-block files: Algorithm

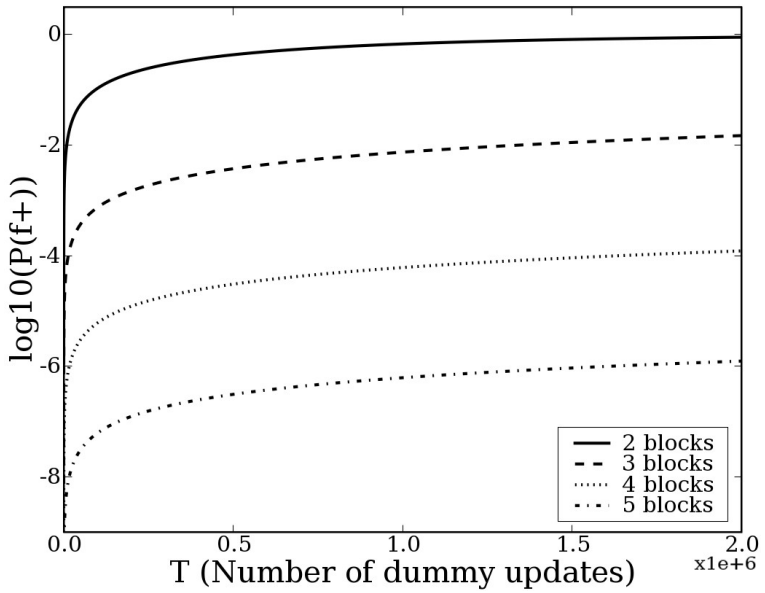


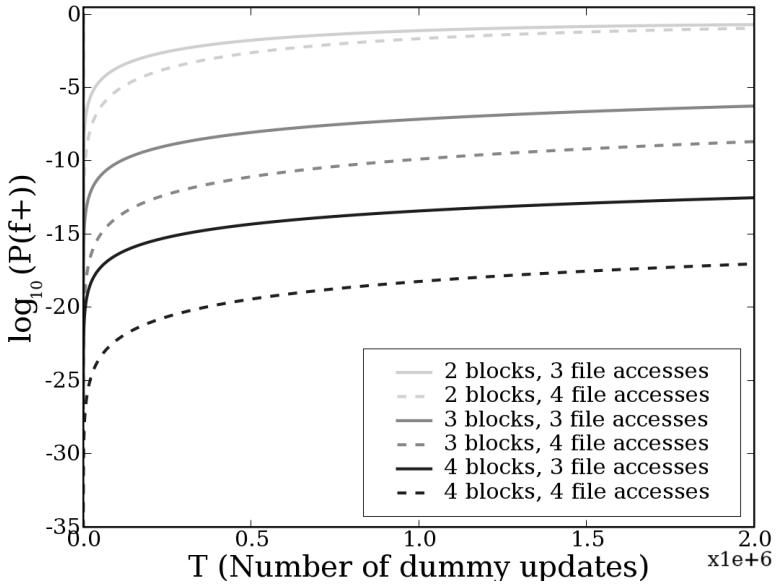
Attacking multi-block files: False positives

- The attacker thinks he has found a file but the patterns have been randomly produced

B = size of storage
b = size of file searched
Number of file accesses ↓

T = total accesses ↑





Attacking one-block files:

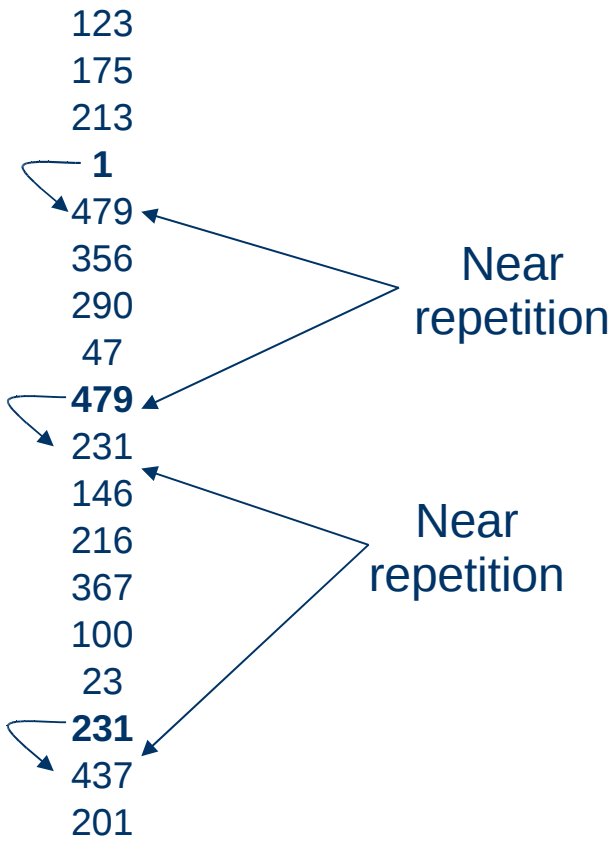
- File blocks updated more frequently than dummy blocks

$$P_D(i) = \left(1 - \frac{1}{B}\right)^{i-1}\left(\frac{1}{B}\right)$$

$$P_F(i) = (1 - f)^{i-1} f$$

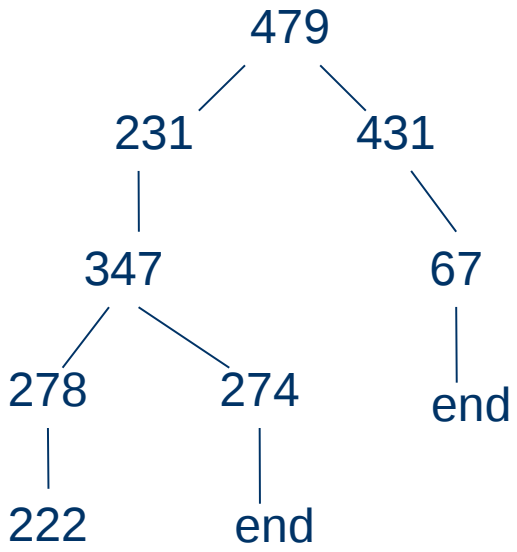
$$f > \frac{1}{B}$$

Need more than one update (hops)



Attacking one-block files: Algorithm (h=5)

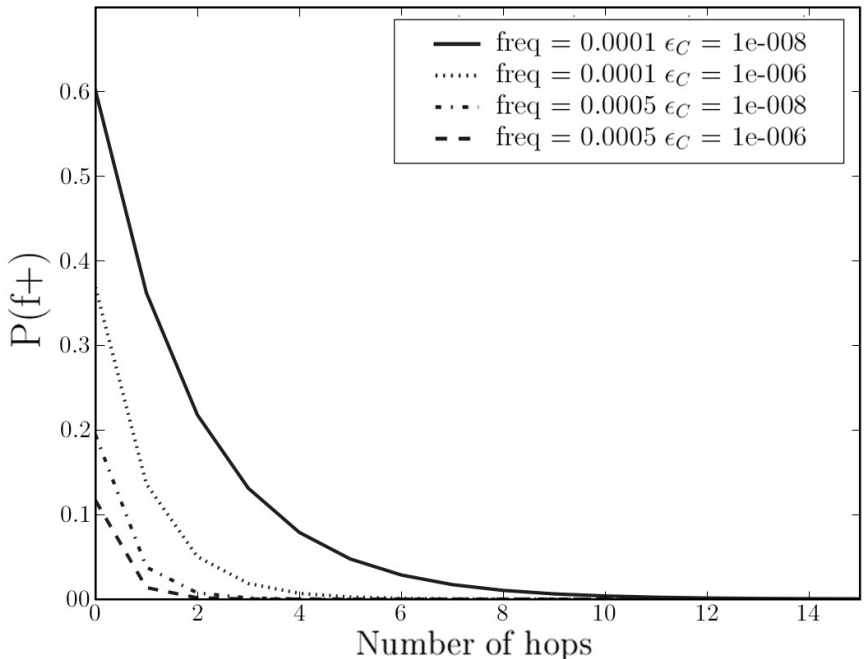
123	431
175	67
213	98
1	239
479	347
356	278
290	95
47	467
479	109
231	21
146	263
216	278
367	222
100	417
23	322
479	347
431	274
231	87
347	9
67	123
12	321



Attacking one-block files: False positives

- Dummy block updates appear *near* (D_C such that $P_F(D_C) < \varepsilon_C$) in the h hops considered

$$P_{f+}(1) = \sum_{i=0}^{D_C} P_D(i) \Rightarrow P_{f+}(h) = \left(P_{f+}(1)\right)^h$$



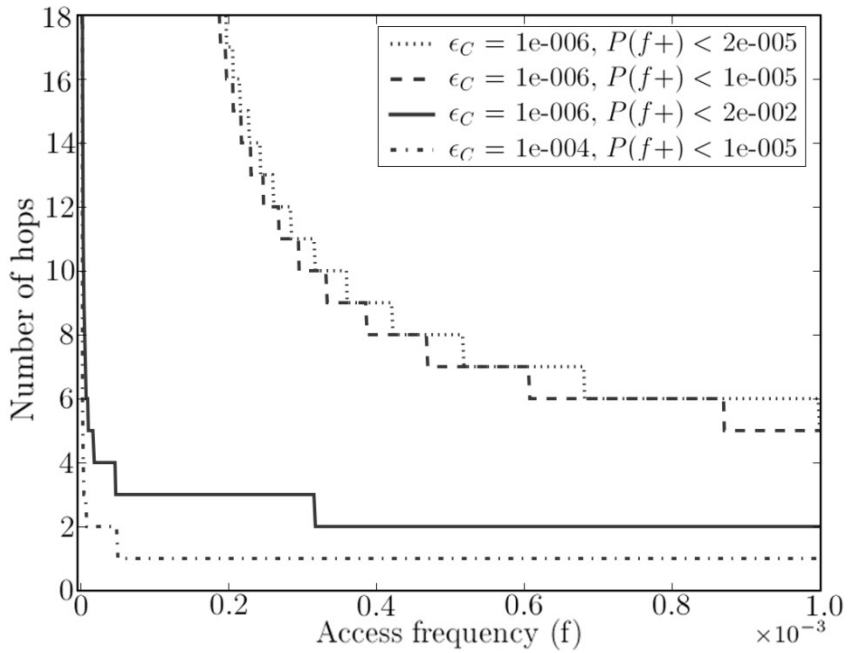
$$\varepsilon_C \uparrow \Rightarrow D_C \downarrow \Rightarrow P(f+) \downarrow$$

$$f \uparrow \Rightarrow D_C \downarrow \Rightarrow P(f+) \downarrow$$

Attacking one-block files: False negatives

- A file update happens *far* ($\geq D_C$) in one of the *h* hops

$$P_{f-}(1) = \sum_{i=D_C}^{\infty} P_F(i) \Rightarrow P_{f-}(h) = \sum_{i=1}^{h} \binom{h}{i} P_{f-}(1)$$



$$\varepsilon_C \uparrow \Rightarrow D_C \downarrow \Rightarrow h \downarrow$$

$$P(f+) \uparrow \Rightarrow D_C \uparrow \Rightarrow h \uparrow$$

Simulation results
Multi-block files

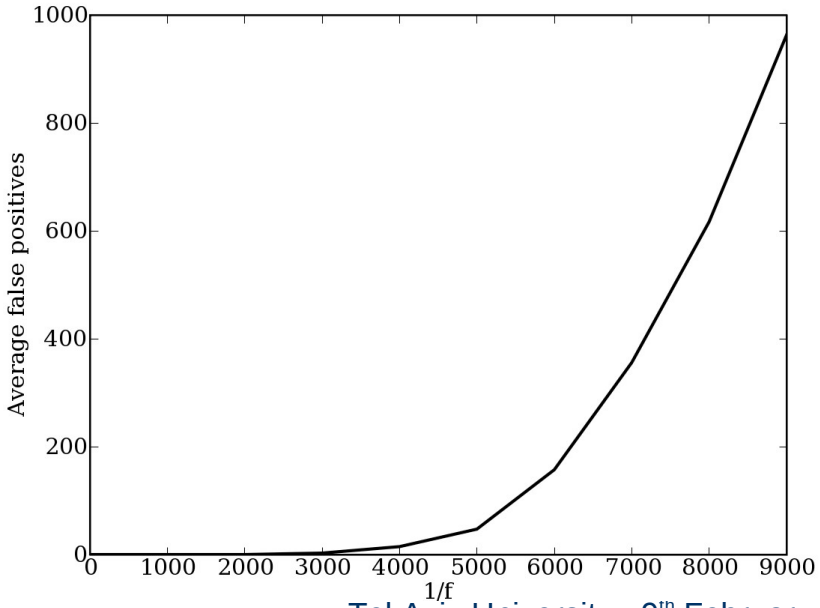
Size of files (blocks)	Number of files per size	File update frequency	Size of storage space
2-3	10	3%	10,000
4-10	10	3%	10,000

Size of files	Files found	Wrong size	False positives
2-3	>99%	<2%	<2%
4-10	>99%	<1%	0

Simulation results
One-block files

Number of files	Accesses to each file	Hops threshold	ε_C	Size of storage space
10	12	10	10^{-8}	100,000

- Rate of success f(f)
- $\downarrow f \Rightarrow \uparrow false - positives$



Conclusions

- Claims unsubstantiated
  - The algorithms do **not** produce same pattern for dummy and user updates
  - The distribution of updated locations is **different** depending on whether there is user activity or not
- Blocks are rarely relocated, and when they are, their new location is known
- Multi-block file updates generate correlations between accessed locations
- Very easy find multi-block files and easy one-block files

A *bit* of randomness is not enough

Thank you for your attention