COSIC

Traffic Analysis Attacks on a Continuously-Observable Steganographic File System

Carmela Troncoso

Claudia Diaz

Bart Preneel

ESAT/COSIC (KU Leuven)

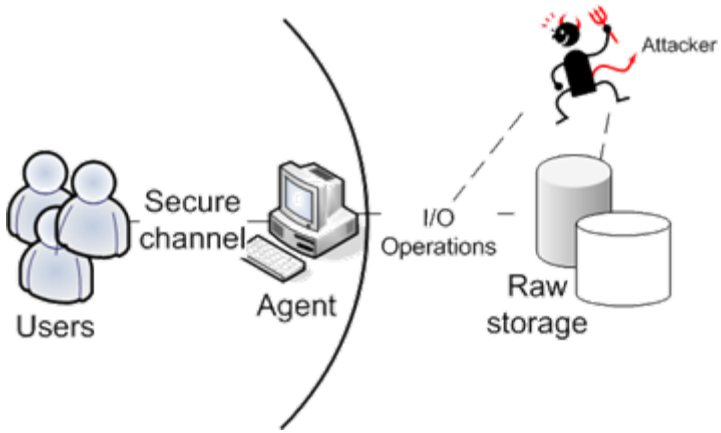
COSIC

Talk Outline

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

What are Steganographic File Systems?

- Protect the user from compulsion attacks
- Hide existence of files
  - Different levels of security
- Continuously observable: attacker monitors system

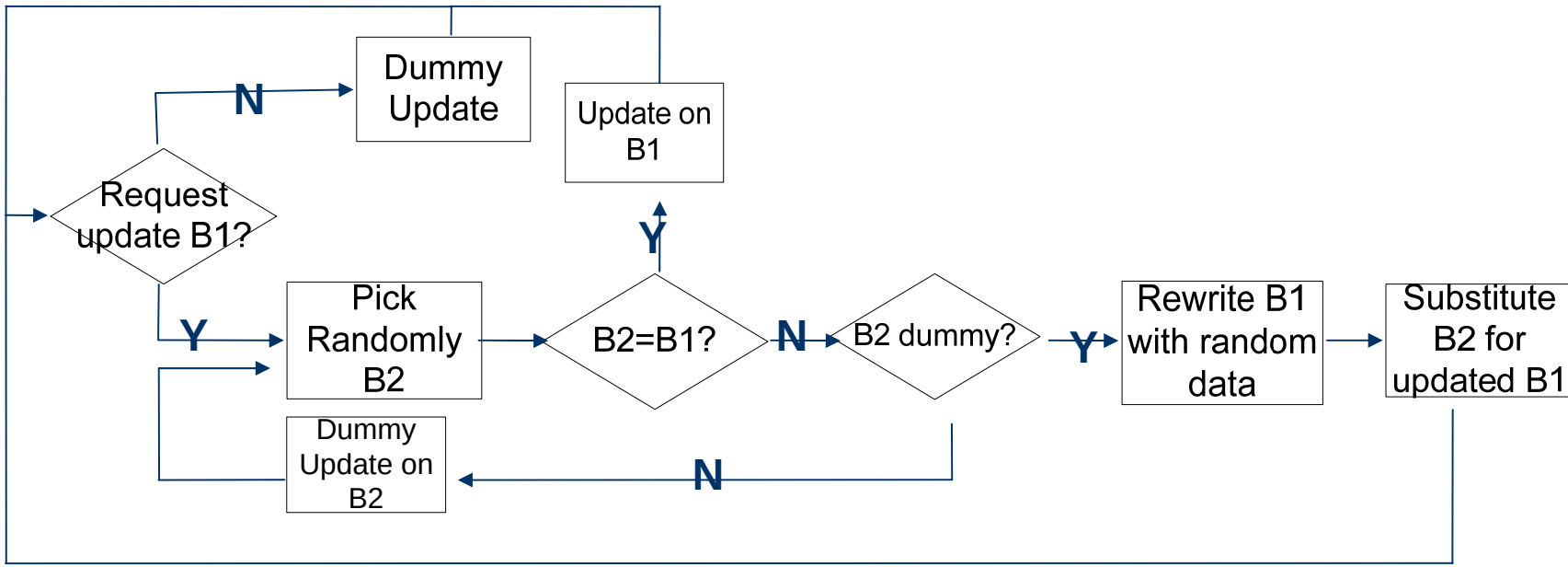


StegFS

- 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 accesed 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: 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 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!!

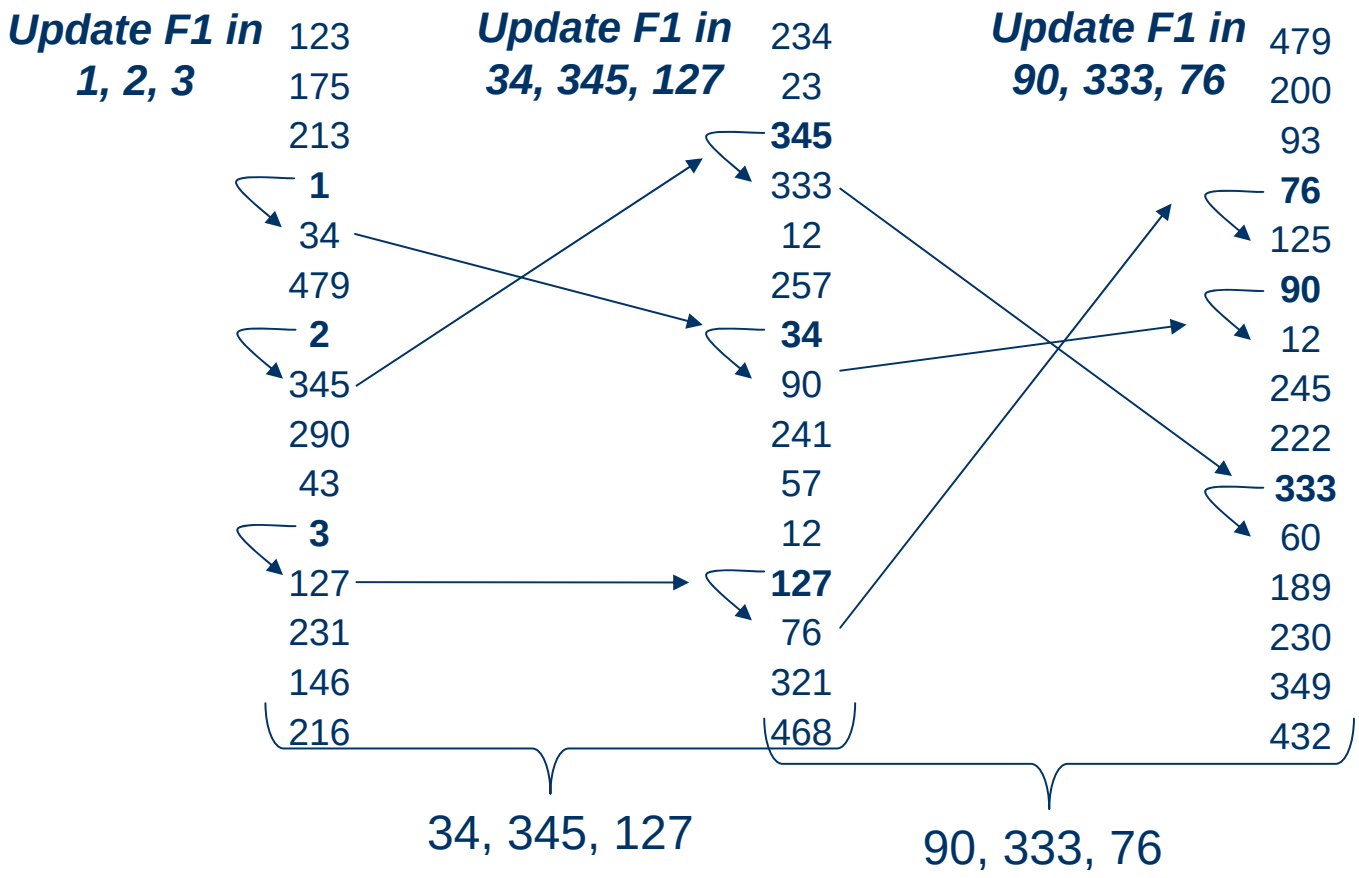
Attacking multi-block files: Update one block

- 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

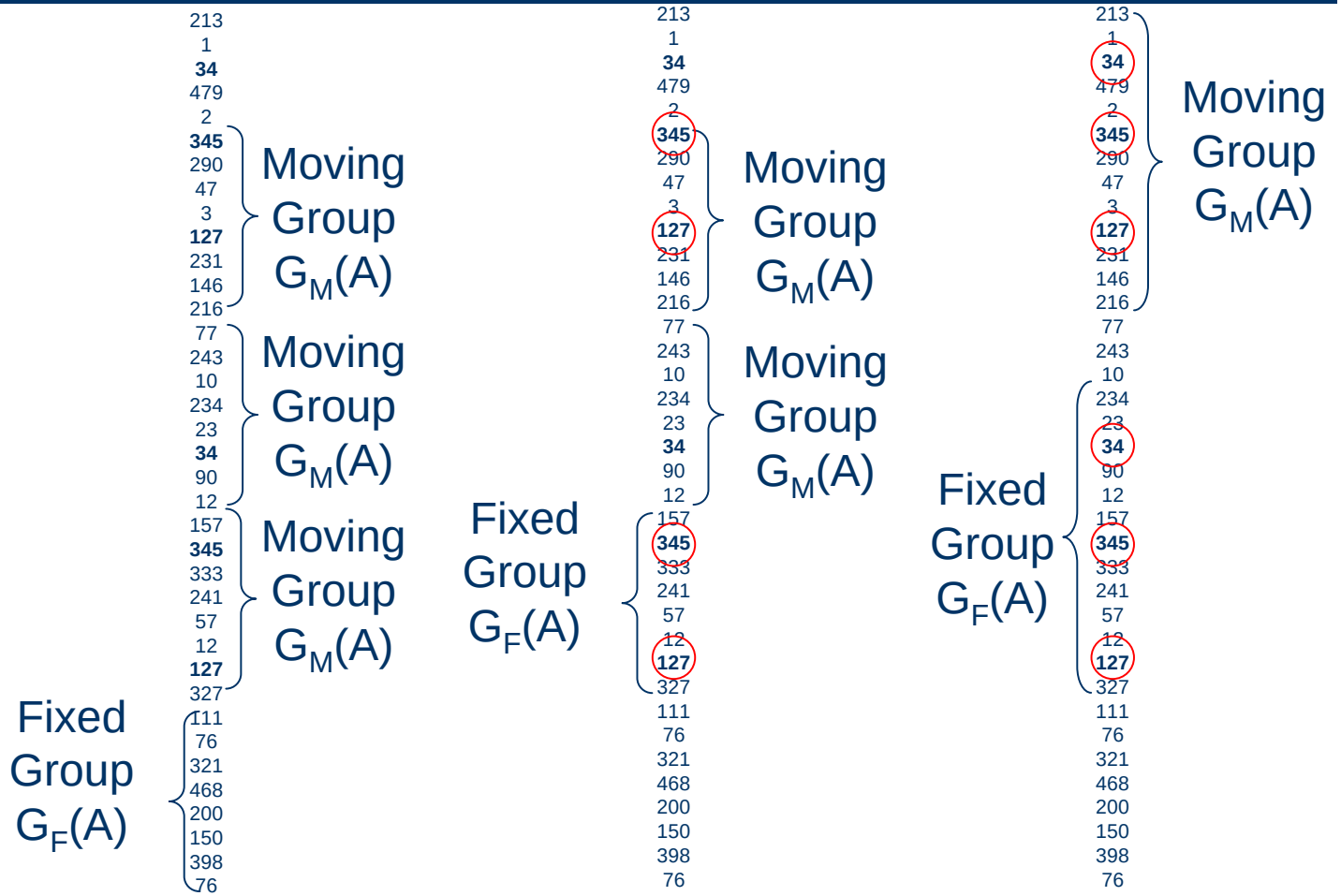
Example: Update block B1=3

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

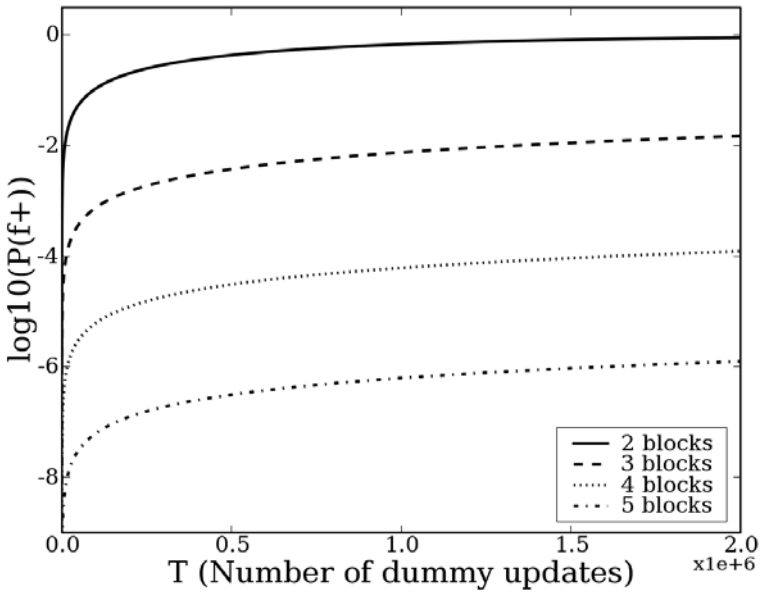


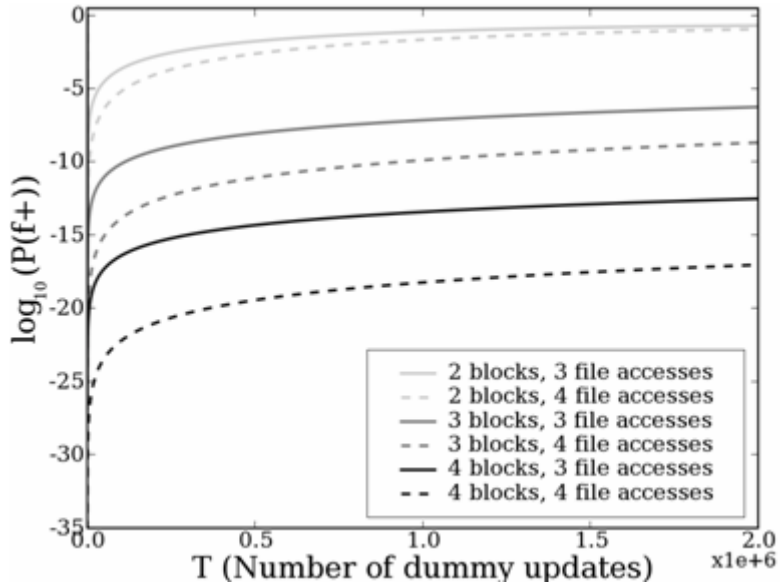
Attacking multi-block files: False positives

- 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:

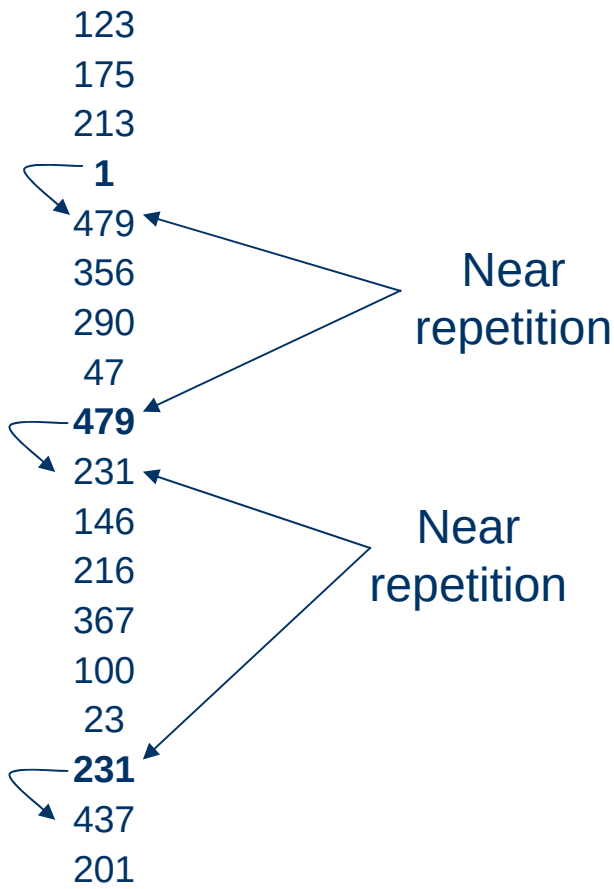
- File blocks updated more frequently than dummy blocks

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

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

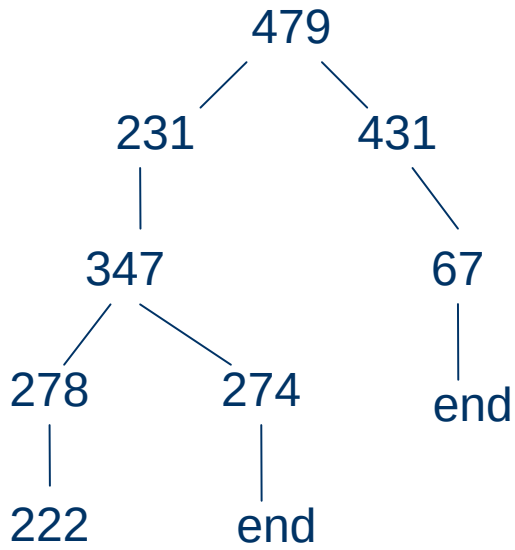
$$f > \frac{1}{B}$$

Need more than one update (hops)



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

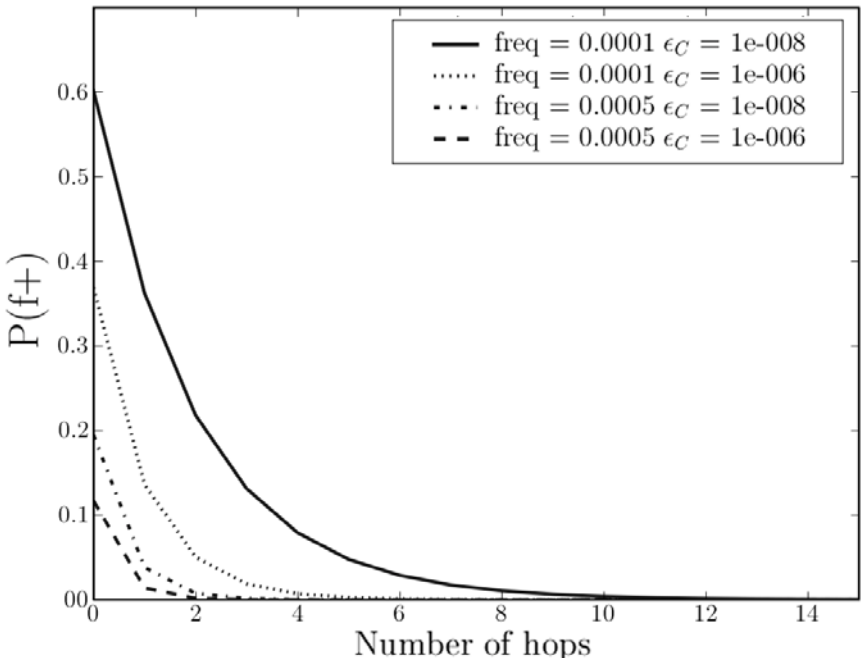
123	431
175	67
213	98
1	239
479	347
356	278
290	95
47	467
479	109
231	21
146	263
216	278
367	222
100	417
23	322
479	347
431	274
231	87
347	9
67	123
12	321



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

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



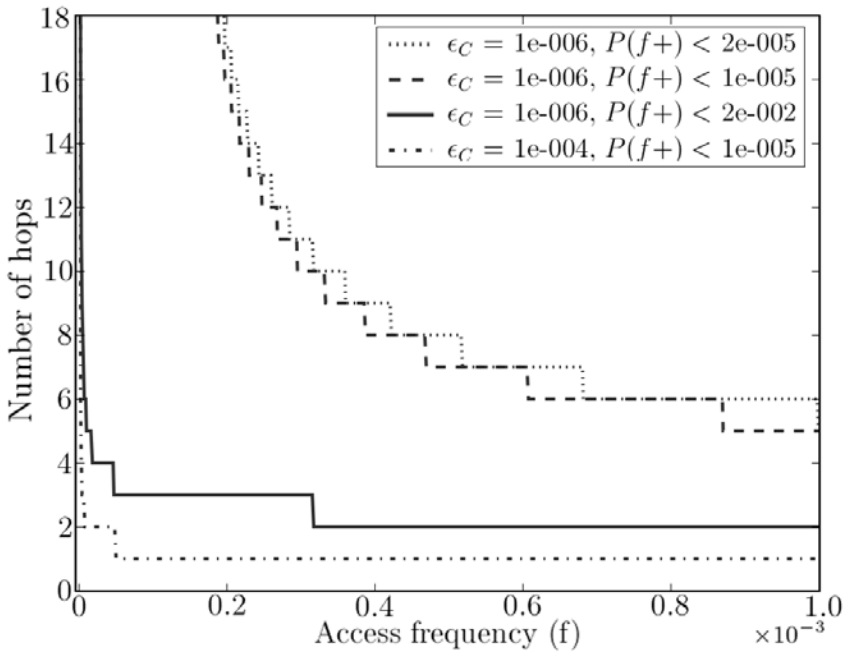
$$\varepsilon_C \uparrow \Rightarrow D_C \downarrow \Rightarrow P(f+) \downarrow$$

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

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

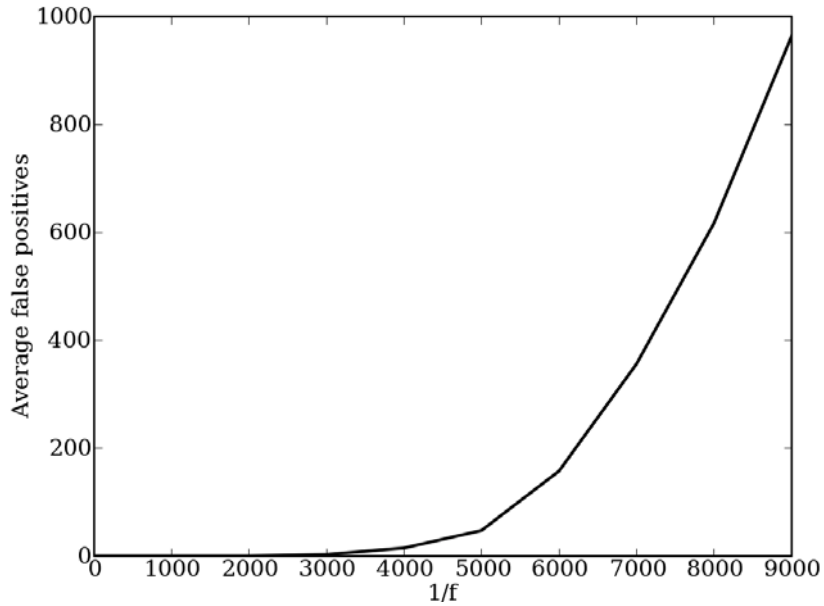
Size of files (blocks)	Number of files per size	File update frequency	Size of storage space
2-3	10	3%	10,000
4-10	10	3%	10,000

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 succes f(f)
- $\downarrow f \Rightarrow \uparrow$ *false positives*



Conclusions

- Claims unsubstantiated
  1. The algorithms do **not** produce same pattern for dummy and user updates
  2. 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