

A FRAMEWORK FOR THE ANALYSIS OF MIX-BASED STEGANOGRAPHIC FILE SYSTEMS

Claudia Diaz, Carmela Troncoso, Bart Preneel

K.U.Leuven COSIC

ESORICS – Malaga, October 8, 2008

MOTIVATION

- Problem: we want to keep stored information secure (confidential)
- Encryption protects against the unwanted disclosure of information
 - but... reveals the fact that hidden information exists!
- User can be threatened / tortured / coerced to disclose the decryption keys (“*coercion attack*”)
 - We need to hide the existence of files
- Property: **plausible deniability**
 - Allow users to deny believably that any further encrypted data is located on the storage device
 - If password is not known, not possible to determine the existence of hidden files

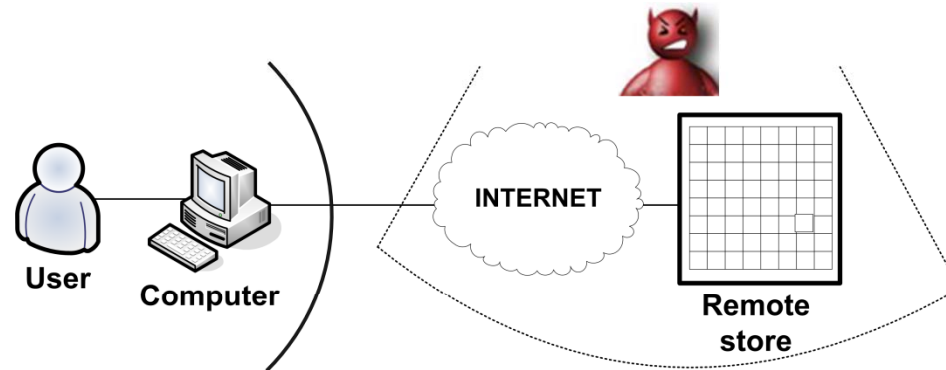
FIRST APPROACHES

- Anderson, Needham & Shamir (1998)
 1. Use cover files such that a linear combination (XOR) of them reveals the information
 2. Real files hidden in encrypted form in pseudo-random locations amongst random data (collisions due to birthday paradox)

- McDonald & Kuhn (1999)
 - 15 default security levels initialized with random keys, such that is not possible to know whether we have revealed the keys to all levels in use
 - User can show some “low” security levels while hiding “high” security levels

ATTACKER MODEL

- Previous schemes resist one/two store snapshots
- What if attacker can observe accesses to the store?
 - Remote or shared semi-trusted store
 - Distributed P2P system



- Monitors accesses to the store prior to coercion
- Ability to coerce the user at any point in time
 - User produces keys to some security levels
 - Attacker inspects user computer
- Game: If attacker is able to determine that the user has not provided all her keys, the attacker wins

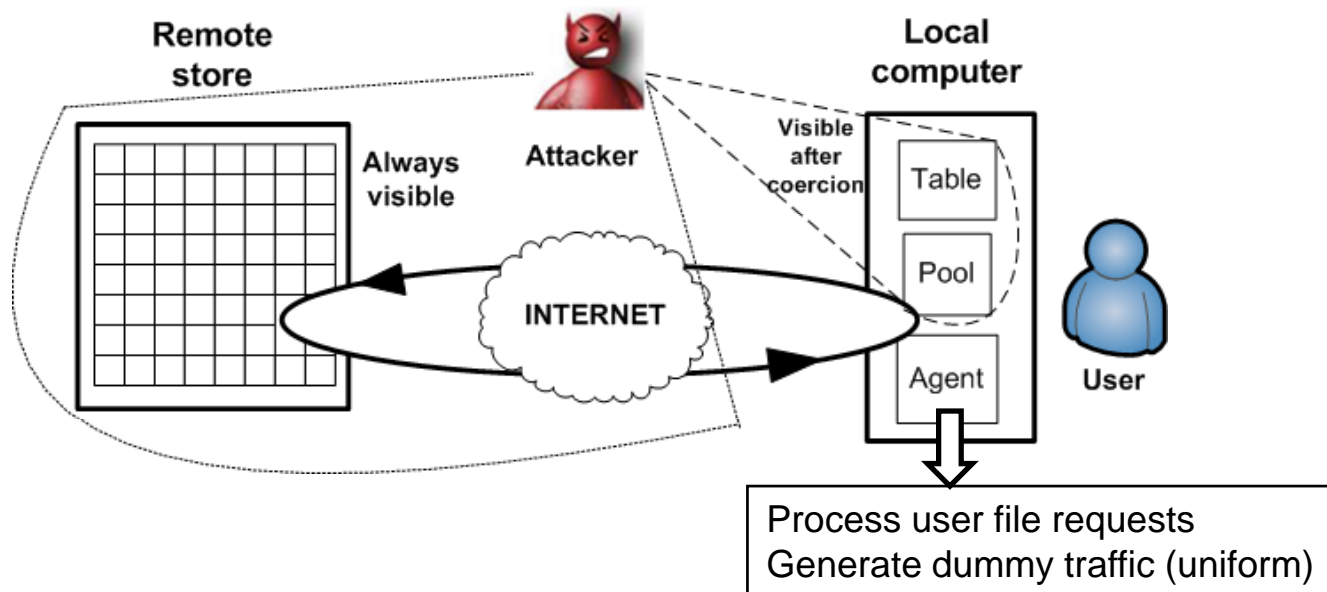
RELATED WORK

- Semi-trusted remote store: Zhou et al. (2004)
 - Use of constant rate cover traffic (dummy accesses) to disguise file accesses
 - Re-encryption and (low-entropy) relocation of file blocks
 - Protects against simple access frequency analysis
 - Broken by Troncoso et al. (2007) with traffic analysis attacks that find correlations between sets of accesses (files in the system can be found prior to coercion)

- Distributed (P2P) steganographic file systems:
 - Mnemosyne: Hand and Roscoe (2002)
 - Mojitos: Giefer and Letchner (2002)
 - Propose dummy traffic to hide access patterns (no details provided)

SYSTEM MODEL

- Files are stored on fixed-size blocks
- Blocks containing (encrypted) file data are undistinguishable from empty blocks containing random data
- Several levels of security
 - User discloses keys to some of these levels while keeping others hidden
 - Data persistence: redundancy (impact on traffic analysis)
- Traffic analysis resistance
 - High entropy block relocation
 - Constant rate dummy traffic



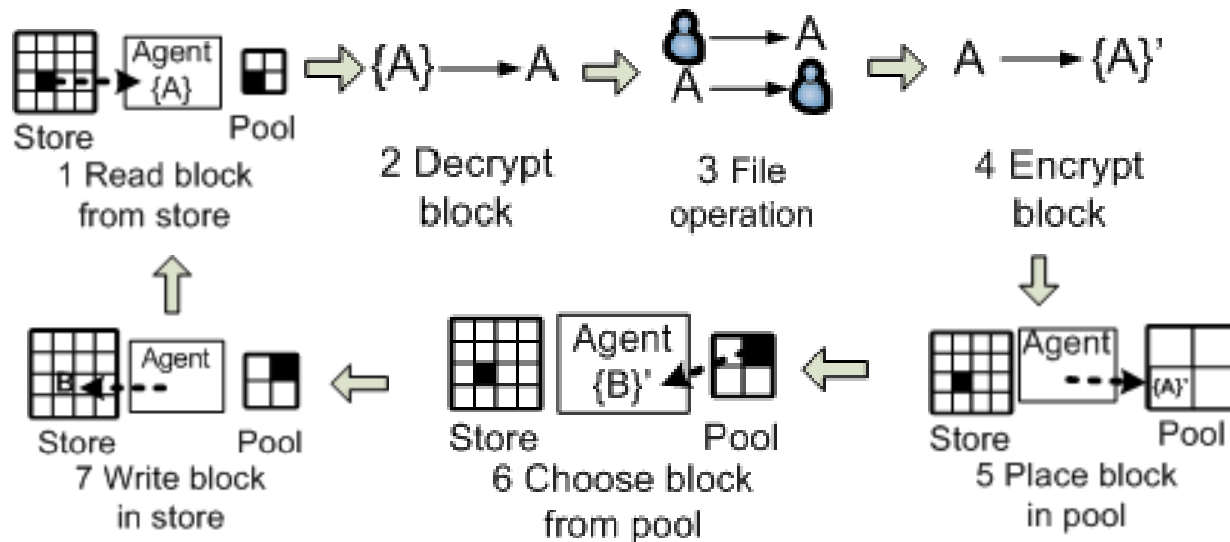
ACCESS CYCLE

Table

Block location	H	Block key	Metadata	$\#$
...
i	H(A)	bk _A	$\mathcal{M}_A = E_{uks}(\text{metadata}_A r_A)$	$\#_A = H(D_{uks}(\mathcal{M}_A))$
...
j	H(Z)	bk _Z	$\mathcal{M}_Z = \text{random}$	$\#_Z = \text{random}$
...

$\{A\}_{b_kA}$
 Location i
 $\{Z\}_{b_kZ}$
 Location j

metadata = (file_name, s, ...) $A = E_{uks}(\text{File data})$ $Z = \text{Random data}$



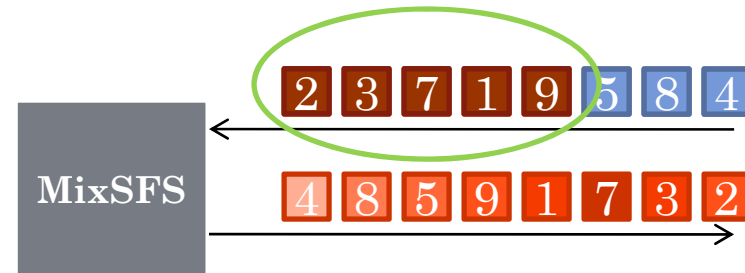
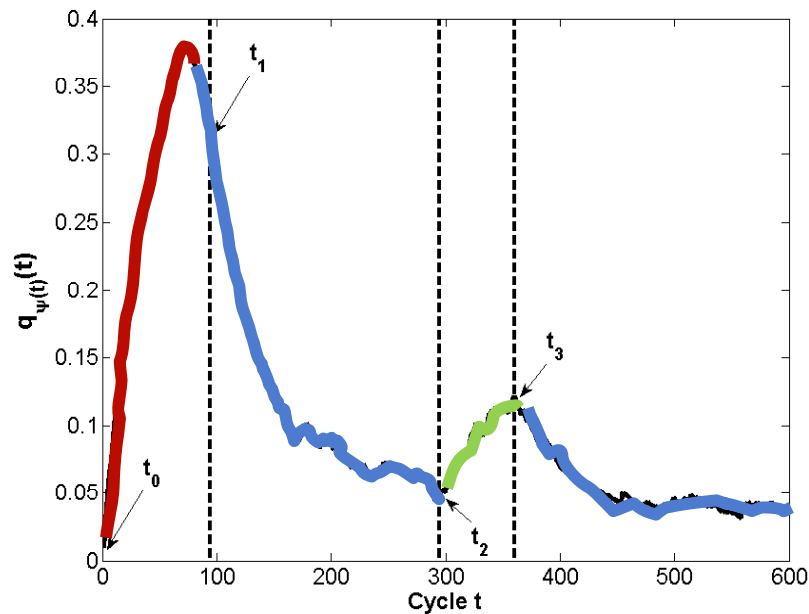
ATTACK METHODOLOGY

1. Attacker profiles the system to extract:
 - Typical access sequences when the user is idle (dummy traffic)
 - Typical access sequences when the user is accessing a file
2. Attacker monitors accesses and looks for sequences that look like file accesses
3. Attacker coerces the user when sequence indicates possible file access (worst case scenario)
4. Attacker obtains some user keys and inspects computer
5. Attacker combines the evidence obtained before and after coercion to try to determine if there are more user keys the user has not provided
6. If the probability of undisclosed keys is high, deniability is low, and vice versa.

EXTRACTING INFORMATION FROM THE SEQUENCE OF ACCESSES TO THE STORE I

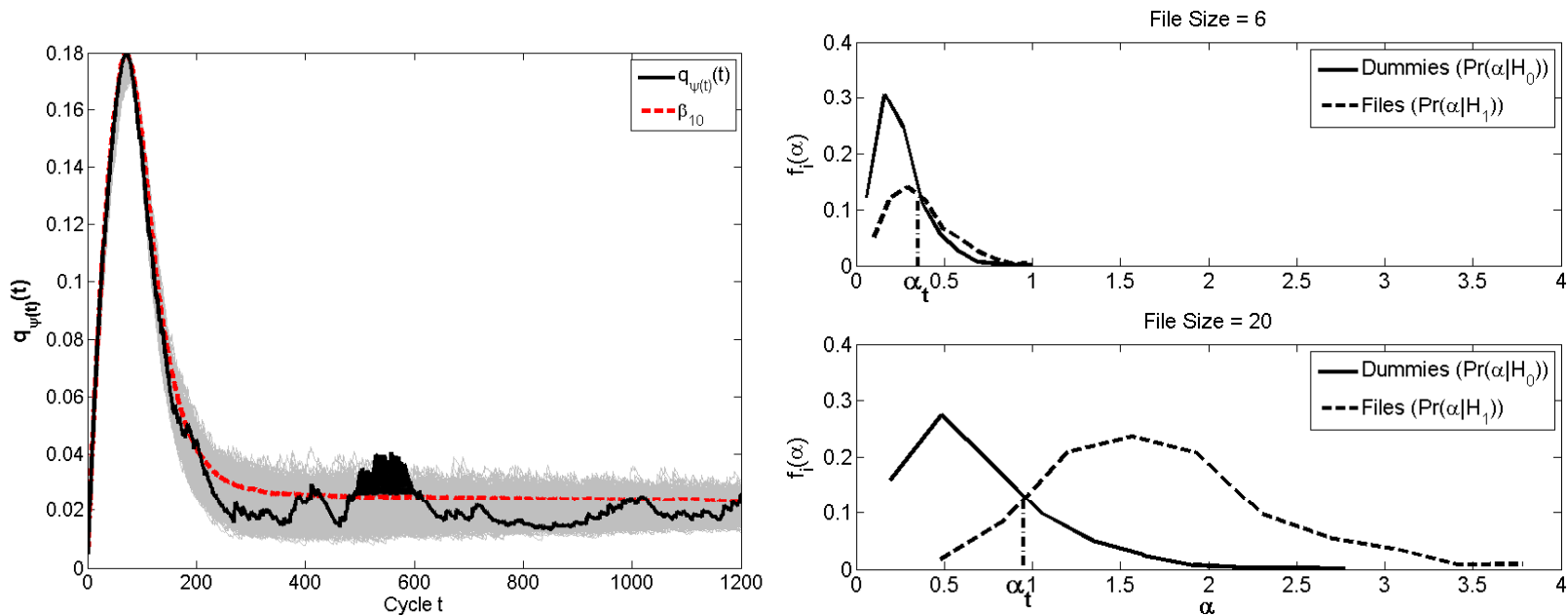
- Attacker profiles the system to extract typical access sequences when the user is accessing a file

$$q_{\psi(t)}(t) = \frac{1}{P} [E_{pool}(t-1) + q_{\psi(t)}(t-1)] \quad E_{pool}(t) = E_{pool}(t-1) + q_{\psi(t)}(t-1) - q_{\psi(t)}(t)$$



EXTRACTING INFORMATION FROM THE SEQUENCE OF ACCESSES TO THE STORE II

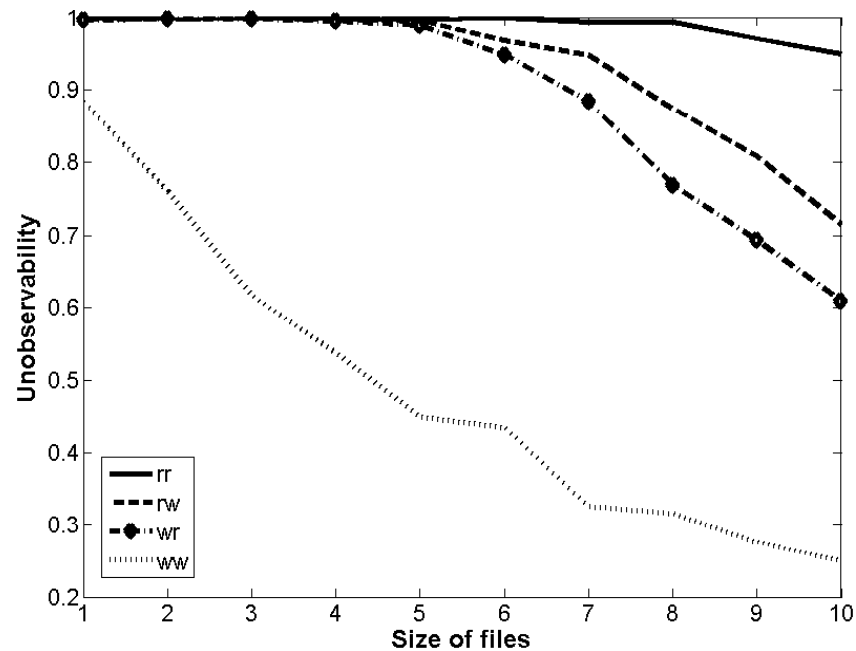
- Attacker profiles the system to extract:
 - Typical access sequences when the user is idle (dummy traffic)
 - Establish a baseline for dummy traffic
- Analyze accesses to store and find strong correlations (unlikely to be generated by dummy traffic)
- For big files, the area that goes over the baseline is much bigger than for dummy traffic (i.e., distinguishable)



SECURITY METRICS: UNOBSERVABILITY

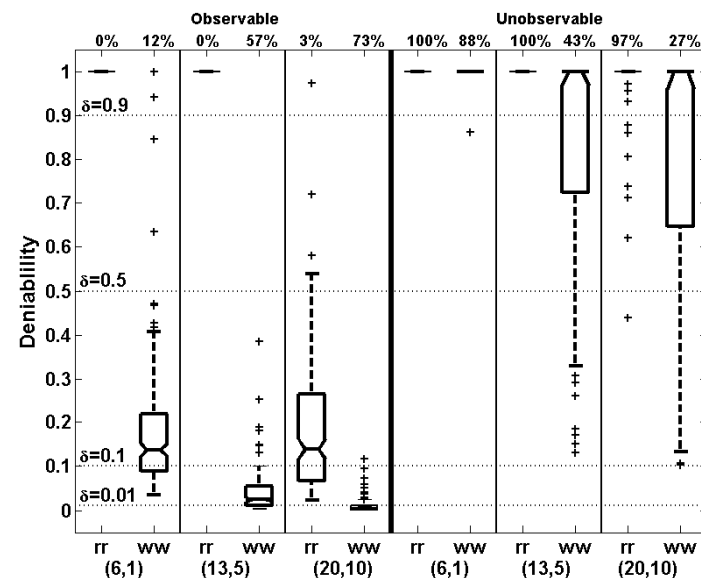
○ Prior to coercion:

- we define **unobservability (U)** as the probability of a file operation being undetectable by the adversary; i.e., the sequence of store accesses generated by a file operation is considered by the adversary as dummy traffic



SECURITY METRICS: DENIABILITY

- After coercion
 - Percentage of empty blocks in pool compared to the percentage in the whole store
 - Worst case scenario: coercion occurs immediately after a file access
- We define **deniability (D)** as the probability that the evidence collected by the adversary (before and after coercion) has been generated by dummy traffic.



CONCLUSIONS AND FUTURE WORK

- We have:
 - Presented an architecture for traffic analysis resistant SFS, of relevance in scenarios where the adversary can observe accesses to the store
 - Formalized security goals and metrics
 - Developed traffic analysis methods to evaluate the system
 - Performed tests to validate the security of the system
- Hard to protect against traffic analysis, even using constant rate dummy traffic
- Hard to conceal file accesses with dummy traffic that selects locations uniformly at random
- Design of smarter traffic analysis strategies



Thank you

