cryptar: Secure, Untrusting, Differencing Backup

Jeff Abrahamson, jeffa@cs.drexel.edu
Drexel University, Department of Computer Science
Object Recognition and Applied Algorithms Lab

Adam O’Donnell, adam@io.ece.drexel.edu
Drexel University, Department of Electrical and Computer Engineering
Computer Communications Laboratory

January 22, 2004
The Problem

Archive files

- No eavesdroppers
- No vandals

Stay practical (network storage).
Picture of the Problem

Alice

Eve

Server
Assumptions

• Expensive network

• Cheap computation

• Simple and cheap

• Hard drives are cheap (but fragile)
Things that Don’t Work

• tape — *too expensive*

• rsync — *cleartext*

• encryption and rsync — *long deltas*
Rsync Strategy

- Block differencing algorithm

But data vulnerable!
Rsync Strategy

Alice

Alice’s data

Server

Store data

Compute checksums

Eve
Cryptar Strategy

- Alice
  - Store data (encrypted!)
  - Compute (and store!) checksums

- Server

- Eve
Cryptar Strategy

- Block differencing algorithm

- Simulate server with a database that knows answers to questions we might ask

- Use a remote block store
Block Cover
Block Cover
Imaginary Client-Server

server

client
Compute Edit Script

server

client
This Doesn’t Work

• This was rsync

• But. . .
  – Gives away secrets
  – Subject to vandals
Thought Experiment

- Simulate server with a database
- Remember only checksums
- Encrypt and store in a block server somewhere!

Then we can answer questions for an imaginary server.
Enter cryptar

For each block, remember

- offset in file
- checksum
- SHA-1
- identifier in block store

Together these are the block list.
For each file, remember

- length, modification date, and SHA-1 of whole file

- offset, checksum, SHA-1, and identifier for blocks in simple block cover (block list)

Put blocks in remote block store.
Enter cryptar

For each file, remember

- length, modification date, and SHA-1 of whole file

- offset, checksum, SHA-1, and identifier for blocks in simple block cover (block list)

Put blocks in remote block store. What about block list?
Enter cryptar

For each file, remember

- length, modification date, and SHA-1 of whole file
- SHA1 and identifier for block list

Put blocks and block list itself in remote block store.
No Eavesdroppers

How do we know no one is reading our files?

- They are encrypted by cryptar, only encrypted data go to the remote block store.
- Even channel is easily encrypted (ssh, IPsec).
No Vandals

How do we know no one has modified our data on the remote block store?

• We have 160 bit SHA-1 cryptographic hashes for every block we store.

• We have SHA-1 hashes for every file, too.
• Consider a 10 MB file stored in 10,240 blocks of size 1K each.

• The probability that no bad block passes the hash is

\[
(1 - 2^{-160})^{10240} \approx 1 - 10240 \left(2^{-160}\right)
\approx 1 - 2^{13}2^{-160}
= 1 - 2^{-147}
\]
Doubting Thomas

Suppose we transfer 1000 such files per second without stop for 100 years. Then the number of blocks transferred would be

\[(1000)(10240)(60 \cdot 60 \cdot 24 \cdot 365.25 \cdot 100) \approx 2^{55}\]

Then the probability that no bad block will pass the hash at some point during that century of work is

\[\left(1 - 2^{-160}\right)^{2^{55}} \approx 1 - 2^{55}2^{-160} = 1 - 2^{-105}\]
Encryption notes

- AES
  - Advanced Encryption Standard
  - a.k.a. Rijndael
  - Replaces DES
Encryption notes

- AES
- Can use multiple keys
- Currently use single key and multiple IV’s (stored remote and clear)
Applications

- Backup
Applications

- Backup
- Pervasive version control
Applications

- Backup
- Pervasive version control
- Poor-man’s Netapp
Availability

Code available, GPL, etc.

- Sourceforge (hopefully)

- http://www.cs.drexel.edu/~jeffa/cryptar/

Contact: jeffa@cs.drexel.edu
        adam@io.ece.drexel.edu