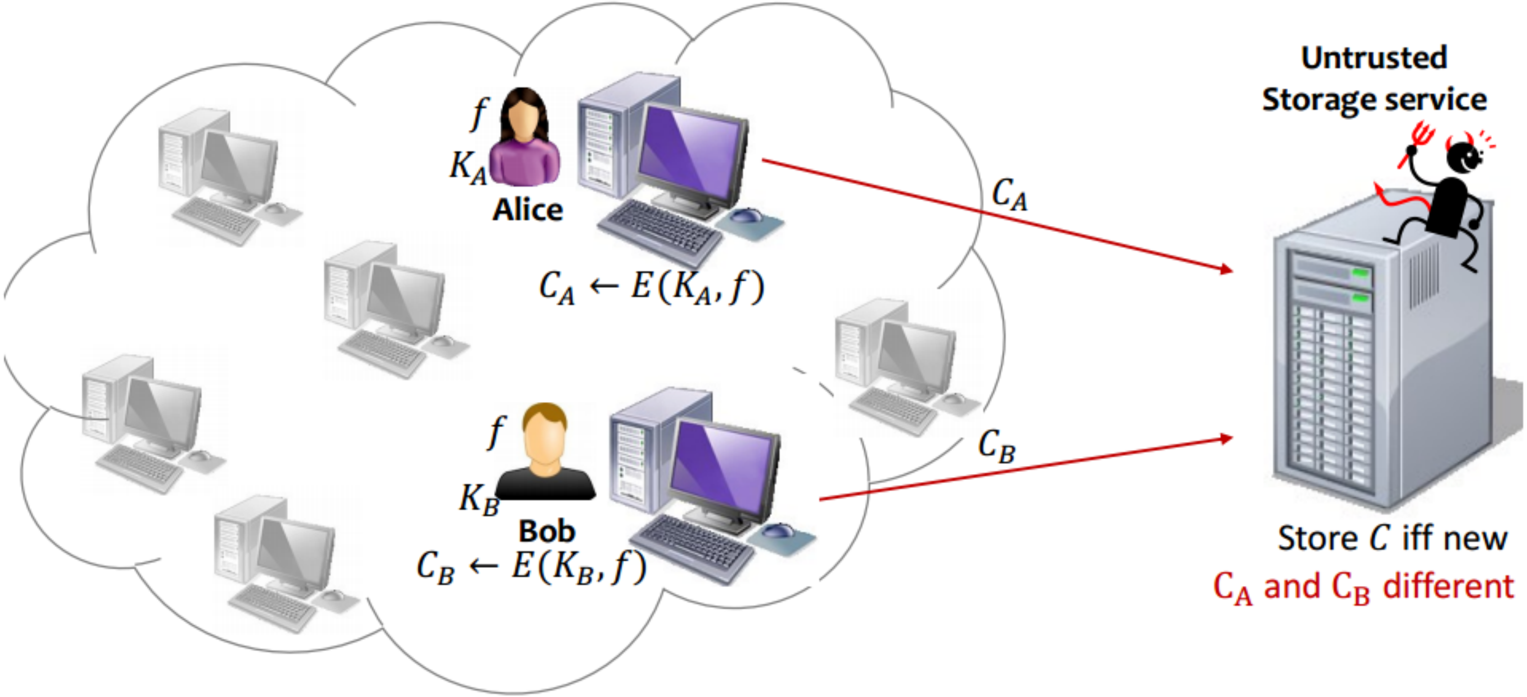
Secure Deduplication and Message Locked Encryption

Vinson Young
Eswar Natarajan

Outline

- Survey of existing techniques and motivation
- Problems of existing schemes and side channels

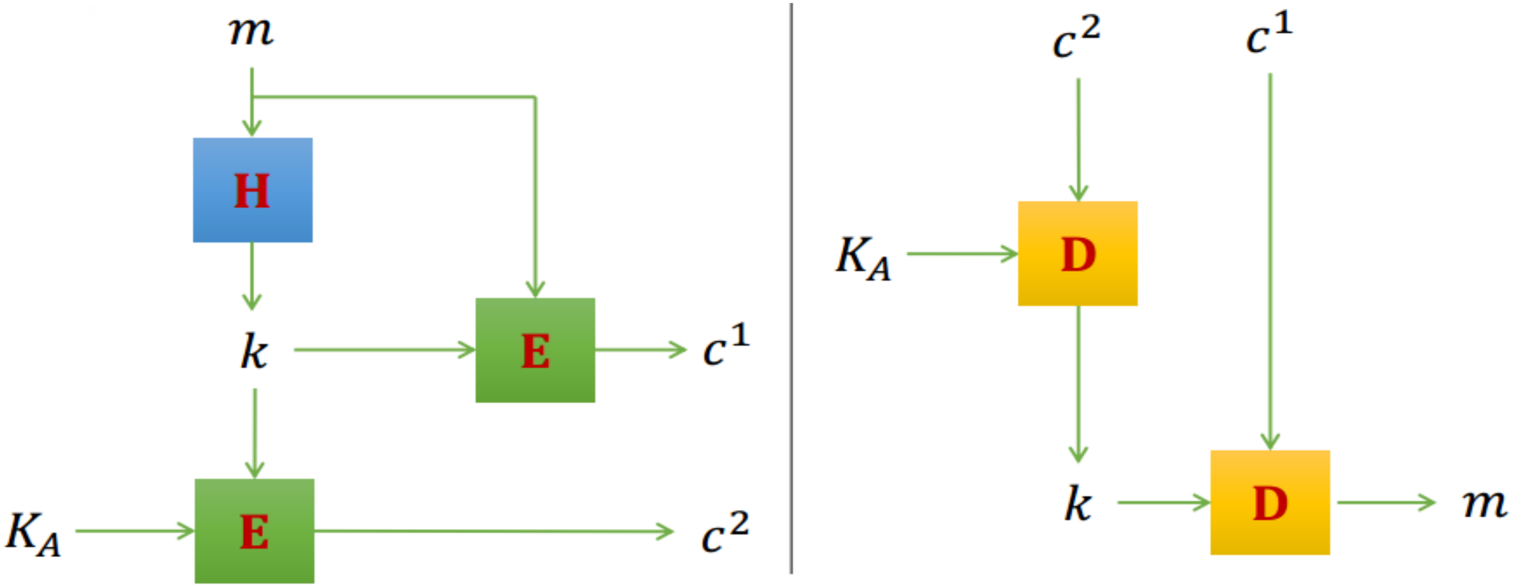
Encryption Worsens Deduplication



Store C iff new

C_A and C_B different

Message Locked Encryption (Enc+Dec)



- Same plaintext encrypts to same ciphertext
- Allows for deduplication at server

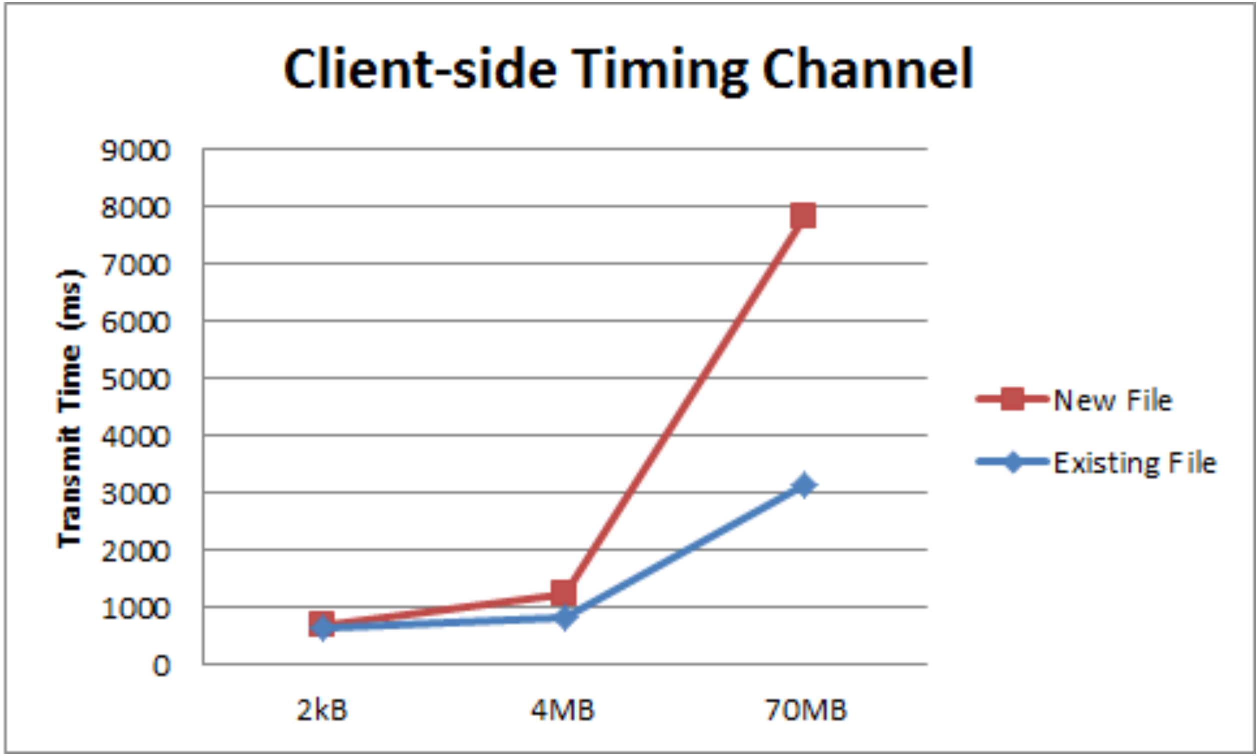
Survey + Background

- Client-based Encryption
  - Bitcasa, Attic backup
  - Efficient on computation and bandwidth for server
  - Side-channels
- Key-server-based Encryption
  - ClouDedup (block-based)
  - DupLESS (file-based)
  - To be analyzed

Client-based Encryption

- Client Steps:
  - Key = Hash of file
  - Encrypt data with message-derived key
  - Check if file is already on server by checking hash
  - If unique, send encrypted file
- Strength -- Encryption + Deduplication
- Weakness
  - Vulnerable to Confirmation/Timing attacks

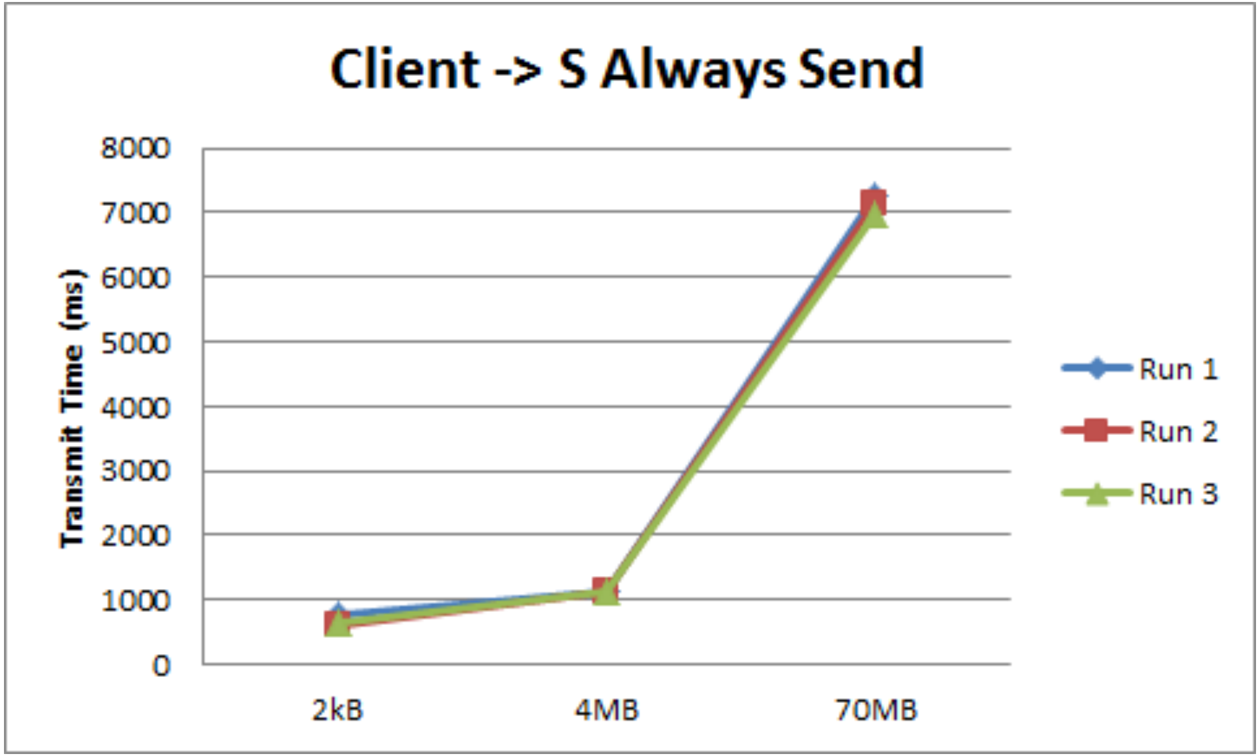
Confirmation / Timing Attack



Modeling Method

- Hash with RSA
- Encrypt with AES
- Send/confirm file via scp
- Time with ‘date’

Close channel by always sending

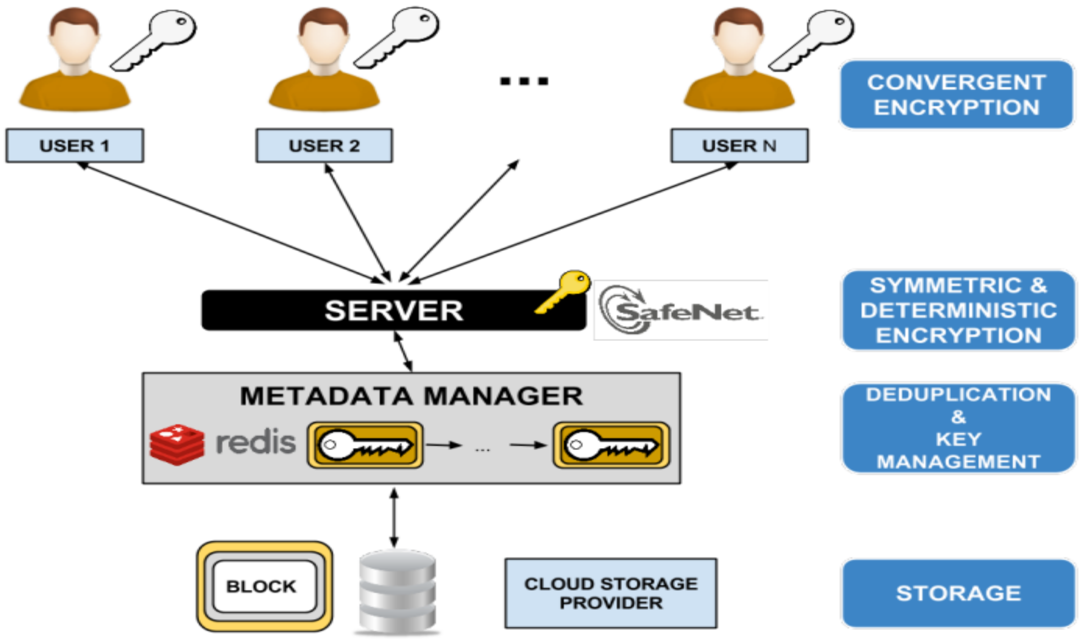


- Client knows encryption method
- Can use brute-force

- Protect by separating entity that handles *encryption/hashing* and *deduplication*

Key-server-based Secure Dedup: #1

ClouDedup



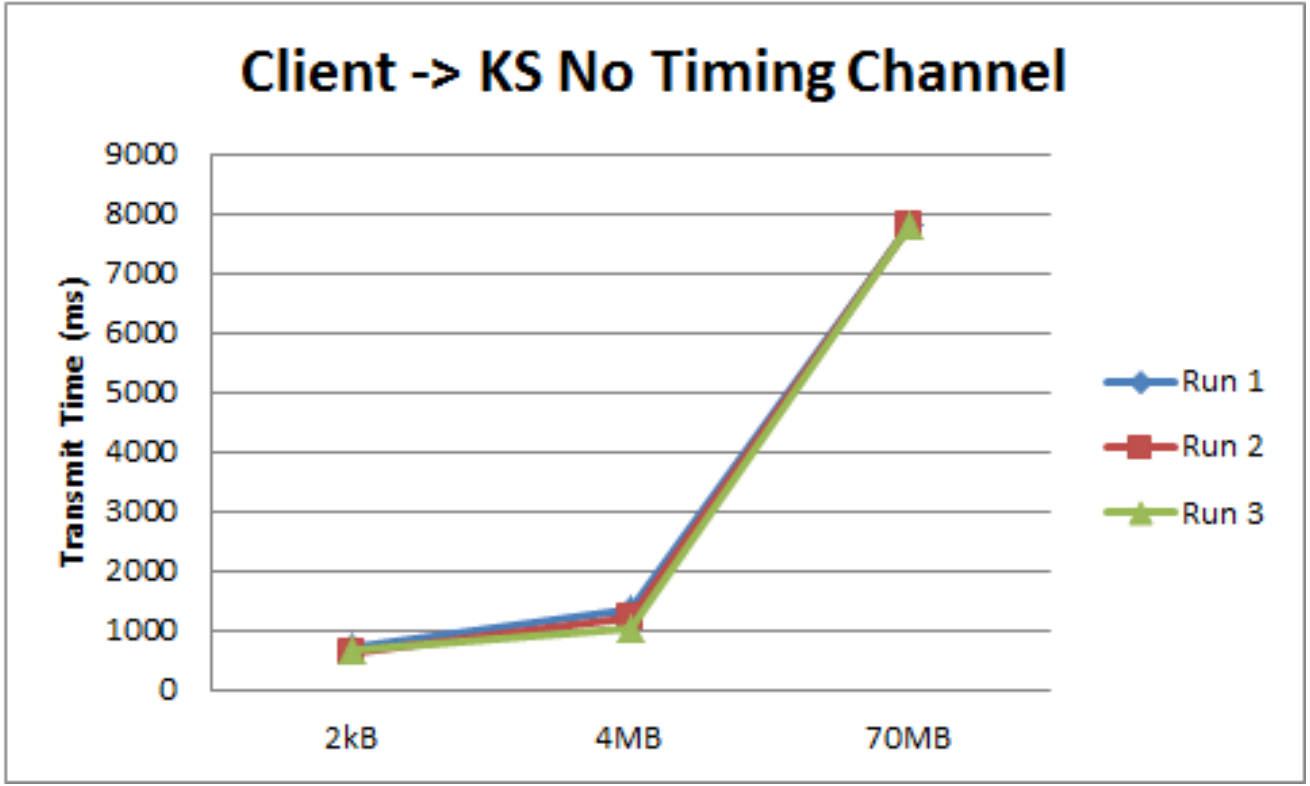
Security Model

- ***Client encrypts*** using convergent encryption
- Send to ***Key server*** to ***manage deduplication***
- Cloud server has no knowledge of organization
  - No individual component can reconstruct file
- 8KB-sized blocks for more deduplication

Timing Channel Closed on Client

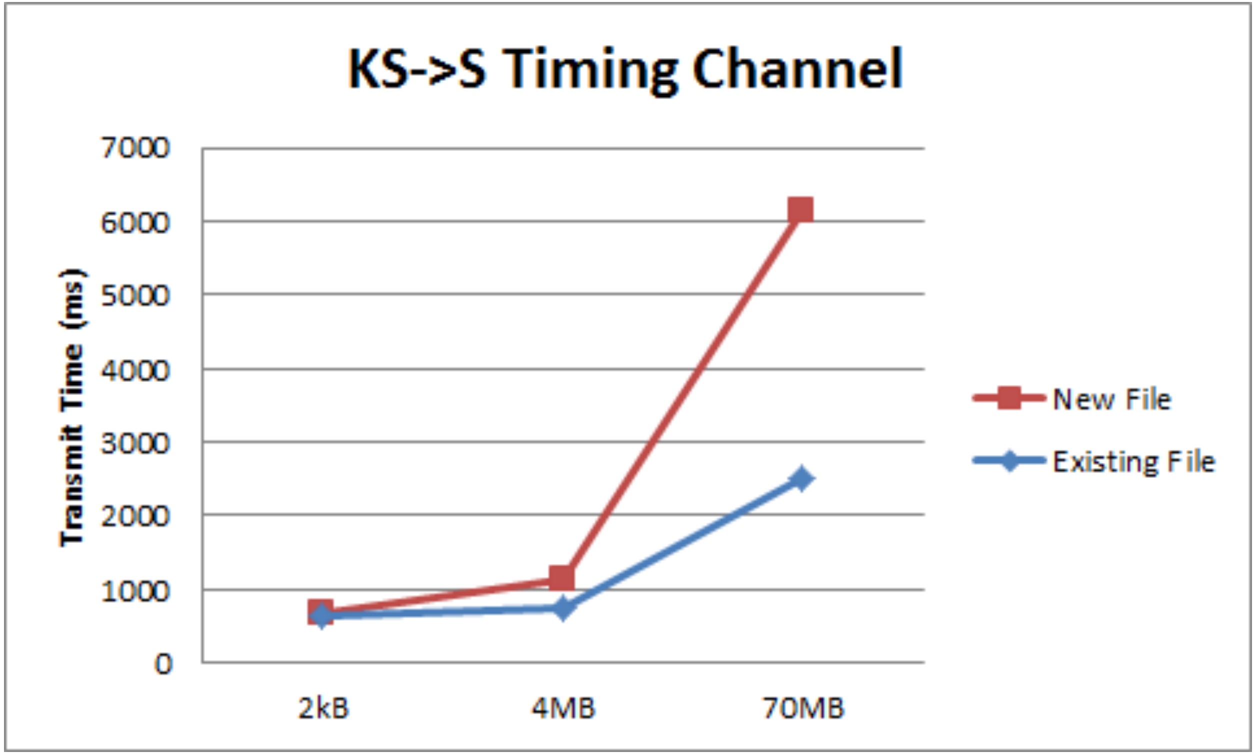
- C->KS (only segment available to the attacker.)
- Getting access to the side channels on the keyserver will require additional vulnerabilities

Client -> KS No Timing Channel



However!

KS->S Timing Channel



Snoop KS->S timing channel?

- Timing channel can be found using indirect means
  - Congestion / bandwidth of KS->S

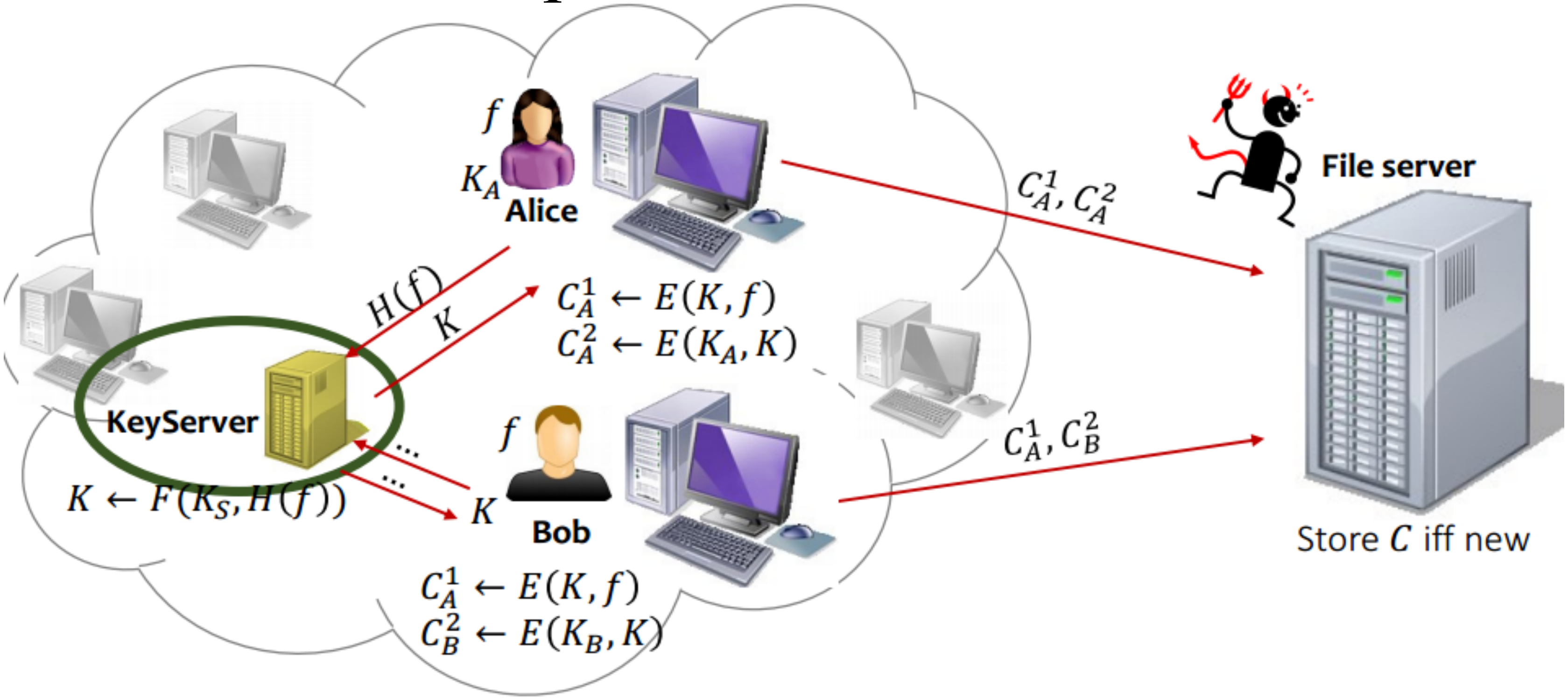
Alternatively: Attack block-based dedup?

Single Book, % deduplicated	70MB-size DNA sequence	4MB Shakespeare
8KB chunks	**0%**	**0%**
1KB chunks	0%	0%
128B chunks		0%

- ClouDedup: Single-file not likely to deduplicate
- Note: “Similar” files however can

Key-server-based Secure Dedup: #2

DupLESS



f
K_A
Alice
$H(f)$
K
$C_A^1 \leftarrow E(K, f)$
$C_A^2 \leftarrow E(K_A, K)$
C_A^1, C_A^2
File server
KeyServer
$K \leftarrow F(K_S, H(f))$
...
...
f
K
Bob
$C_A^1 \leftarrow E(K, f)$
$C_B^2 \leftarrow E(K_B, K)$
C_A^1, C_B^2
Store C iff new

Security Model

- Secure
- Uses Oblivious PRF to create keys
- Defaults to the current web security standards like rate limiting, fraud detection algorithms and limiting access.
- Assumes the keyserver is part of the protected network and is secured with firewalls etc. Not perfect but is standard.

DupLESS key features

- Close timing channels
  - Always send C->KS and C->S
- Mitigate brute force chances
  - Hashing is done by KS

Conclusion

- Surveyed of existing techniques and surveyed weaknesses in current dedup schemes.
- Demo of existing attack against weaker schemes
- Proposal of new attack vectors

References

- “DupLESS: Server-Aided Encryption for Deduplicated Storage”;Mihir Bellare and Sriram Keelveedhi, UCSD;Thomas Ristenpart, UW-Madison
- "Message-locked encryption and secure deduplication";BELLARE , M., KEELVEEDHI , S., AND RISTENPART , T. EUROCRYPT 2013
- “ClouDedup”; http://elastic-security.com/2013/12/10/cloudedup-secure-deduplication/

To do: Block-based Timing Attack

- Security sensitivity to blocking size. ClouDedup uses 8kB size, which is large enough to not see deduplication in single file.
- Sensitivity graph: Dedup vs. block size.