Private Editing Using Untrusted Cloud Services

Yan Huang and David Evans
University of Virginia

http://MightBeEvil.com

UNIVERSITY of VIRGINIA

Motivation

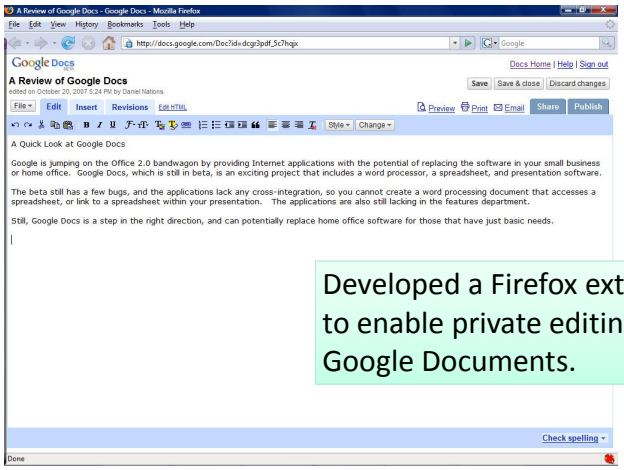
- To take advantage of existing cloud services without revealing private data to untrusted servers.
- We expect a solution that
  - is easy to deploy
  - and results in minimal negative interference with existing functionalities

Observation: Server Doesn't Need Data

Many cloud applications perform most **data-dependent** computation on the client:

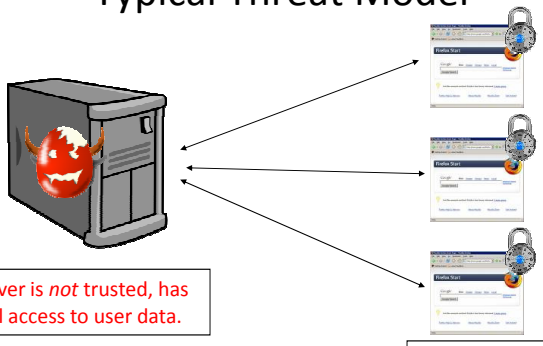
- Reduce server load
- Reduce laten

Computation needed at server-side:
- Protecting proprietary algorithms
- Greater computing power
- Large data needed



Developed a Firefox extension to enable private editing using Google Documents.

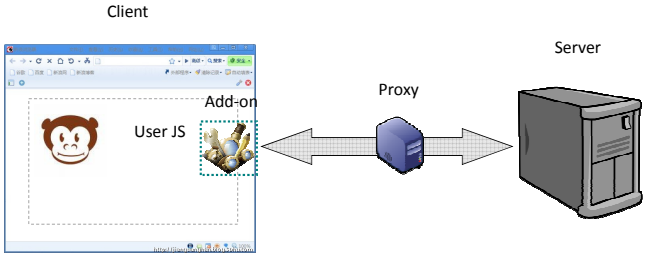
Typical Threat Model



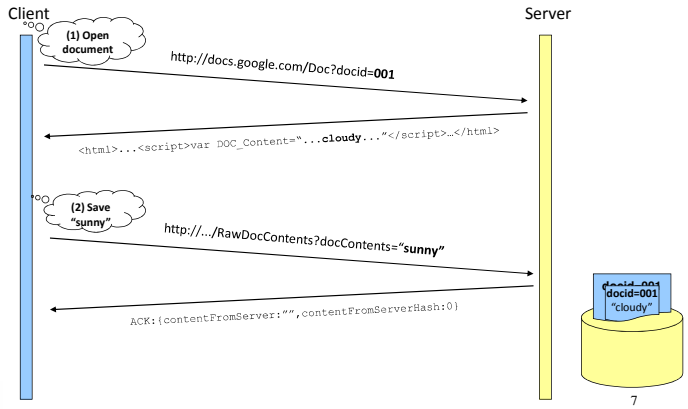
Server is *not* trusted, has full access to user data.

Browser is not compromised

Design Choices

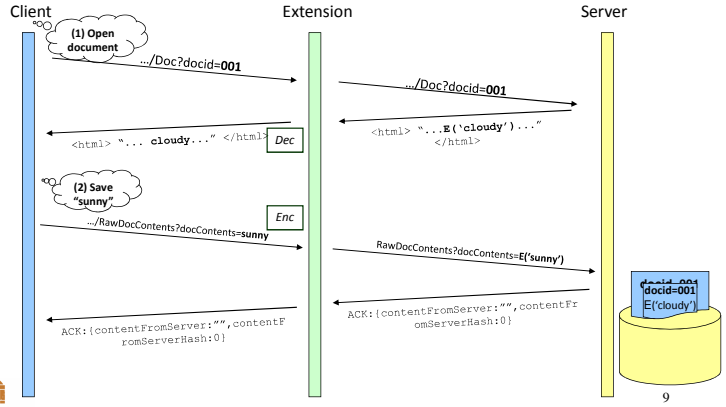


Protocol Without Extension

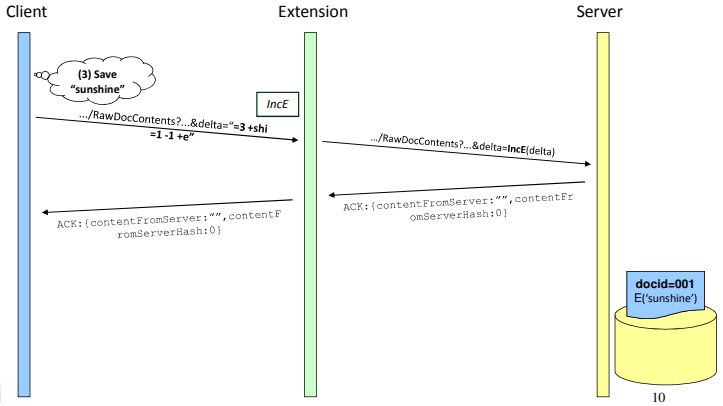


Protocol Without Extension

Protocol With Extension



Protocol With Extension

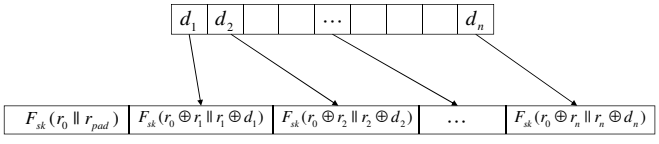


Incremental Encryption

Gen	key generation algorithm
$c = Enc_k(m)$	encrypt whole message m
$m = Dec_k(c)$	decrypt ciphertext c
$c' = IncE_k(op, m, c)$	given a key k, an edit operation op, previous message m, and previous ciphertext c, compute an updated ciphertext c'.

[BKY01] – Buonanno, Katz, and Yung. *Fast Software Encryption* 2001

Privacy Only Mode



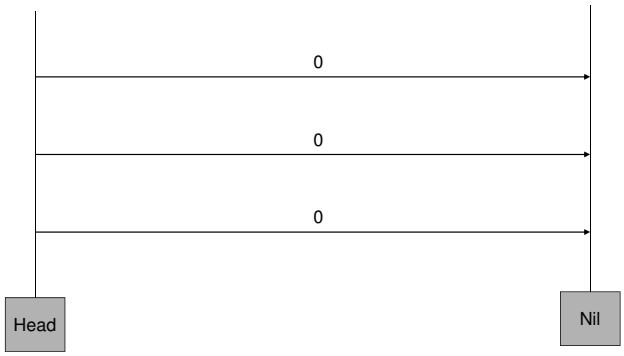
F	Trapdoor psuedorandom permutation
r_i	Random numbers
d_i	Document segments
$\|$	Concatenation

Multiple Characters per Block

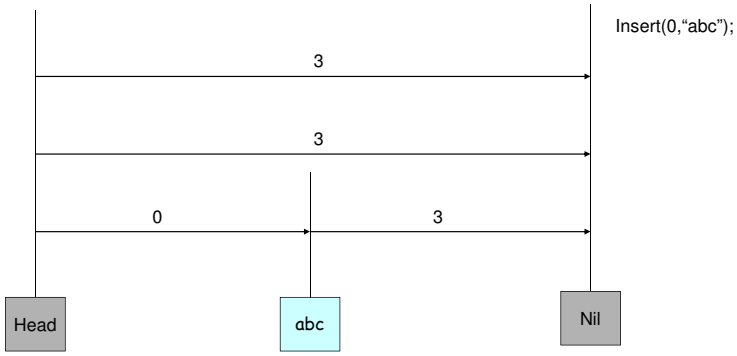
Motivation: reduce the ciphertext blow-up

Challenge: the *index* of each character will change so that naïve implementation won't work

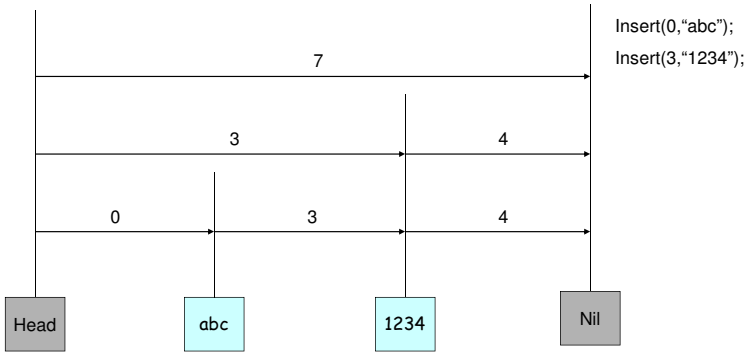
IndexedSkipList



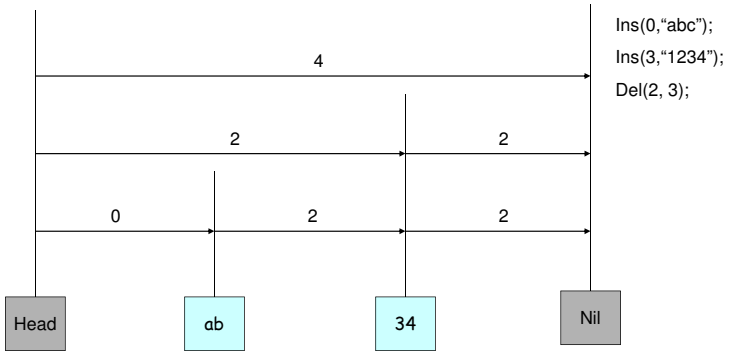
IndexedSkipList



IndexedSkipList



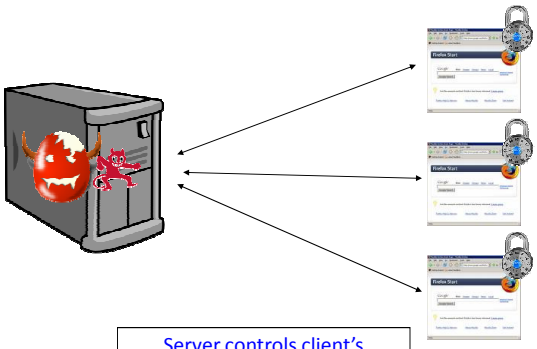
IndexedSkipList



Security Analysis

- Server knows the document ciphertext
- Can infer the length of original document
- Knows editing positions and edit operation types
- Can deny service

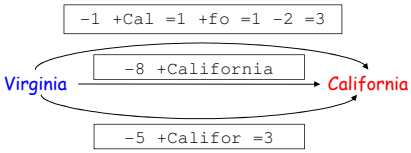
Extreme Threat Model



Server controls client's application content software.

Security Analysis

- Covert Channels such as timing or delta



- Using obfuscated protocol
- Dynamically generated client/server protocols

Functional Evaluation

- Basic editing functions are supported
- Features disrupted:
  - Translation
  - Spell checking
  - Drawings
  - Export
  - Collaboration

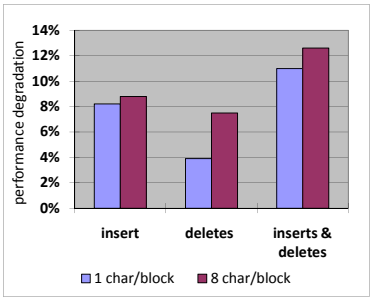
Performance Evaluation Micro-Benchmarks

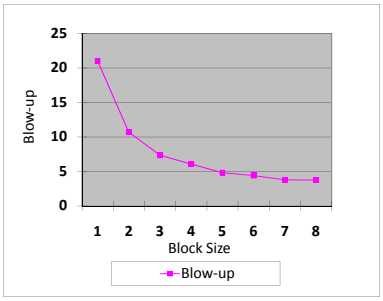
	Average Time (per char)
encryption	**.091 ms**
decryption	**.085 ms**
transform delta	**.110 ms**

Microbenchmark of privacy-and-integrity mode

(avg. of 1000 tests)

Macro-Benchmarks





Conclusion

- Editing encrypted data can be practical
- Practical secure computation under relaxed security definitions can be very useful
- Would be a very challenging problem should the service provider chooses not to cooperate

Download from: www.MightBeEvil.com/securedocs