Keypad: Auditing Encrypted Filesystem for Theft-prone Devices

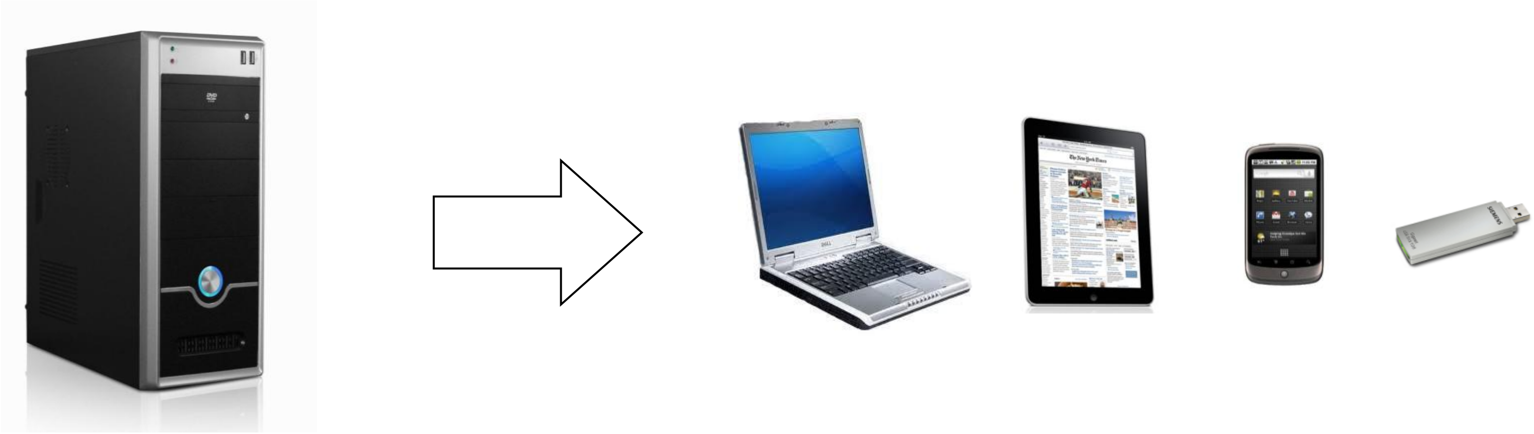
Roxana Geambasu
John P. John
Steve Gribble
Yoshi Kohno
Hank Levy

University of Washington

The Move to Small, Powerful, Mobile Devices

- Small, powerful mobile devices are replacing desktops
- Mobile devices bring important advantages:
  - Location-based services, mobile web
  - Constant connectivity, data access, email

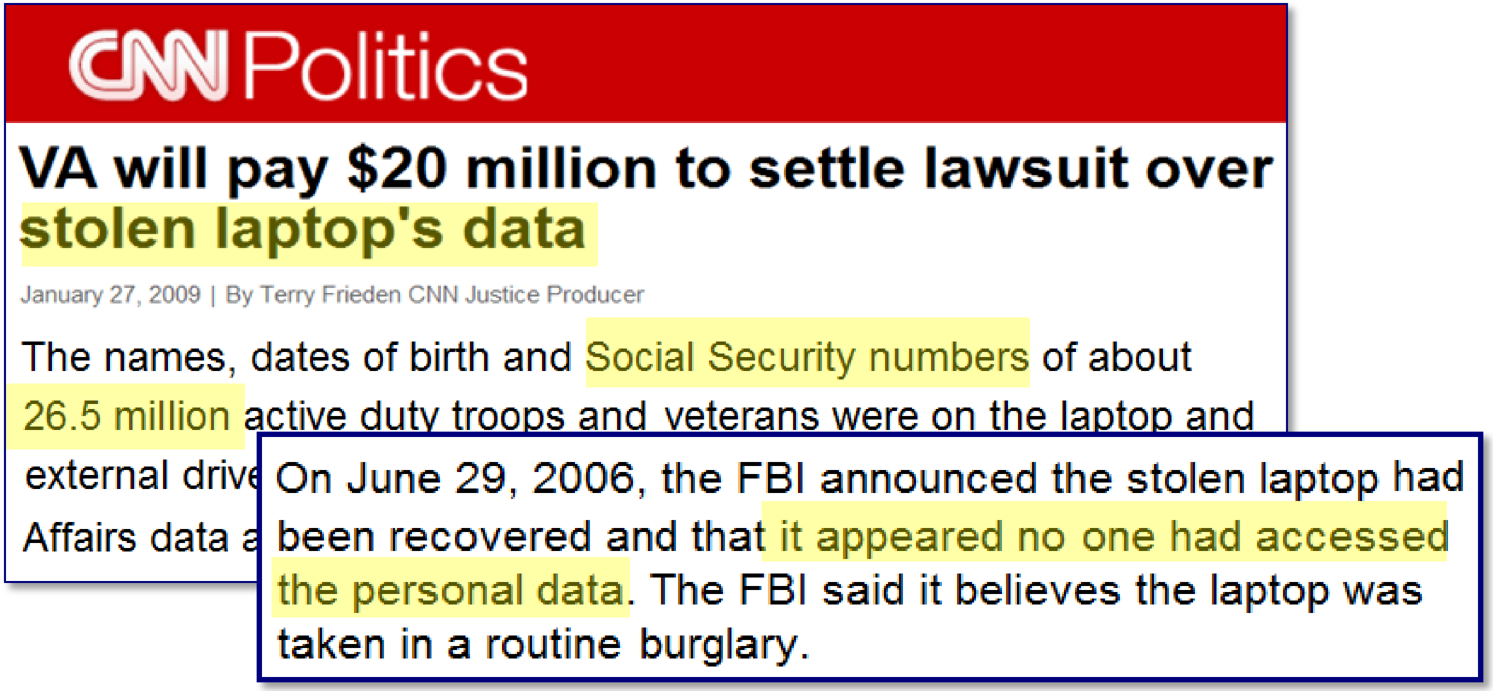
Desktop *Small mobile devices*



The Problem with Mobile Devices

- Mobile devices are prone to theft and loss
  - □ 500K laptops per year are lost in US airports [Ponemon Institute '09]

- Mobile device theft/loss exposes sensitive data
  - □ SSNs, financial data, health data, trade secrets, state secrets, …



CNN Politics

VA will pay $20 million to settle lawsuit over stolen laptop's data

January 27, 2009 | By Terry Frieden CNN Justice Producer

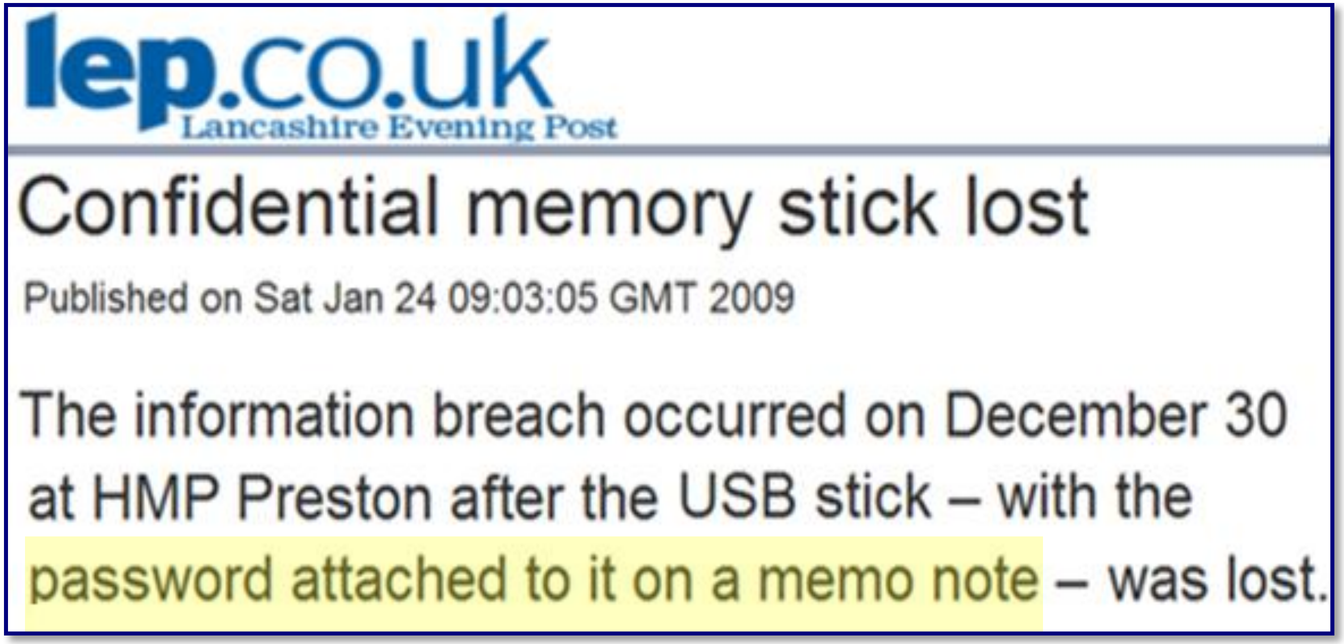
The names, dates of birth and Social Security numbers of about 26.5 million active duty troops and veterans were on the laptop and external drive

Affairs data a

On June 29, 2006, the FBI announced the stolen laptop had been recovered and that it appeared no one had accessed the personal data. The FBI said it believes the laptop was taken in a routine burglary.

Is Encryption Sufficient?

- Encrypting files on a mobile device increases security
  - E.g.: BitLocker, PGP Whole Disk Encryption, TrueCrypt, …
- But is encryption enough?



lep.co.uk

Lancashire Evening Post

Confidential memory stick lost

Published on Sat Jan 24 09:03:05 GMT 2009

The information breach occurred on December 30 at HMP Preston after the USB stick – with the password attached to it on a memo note – was lost.

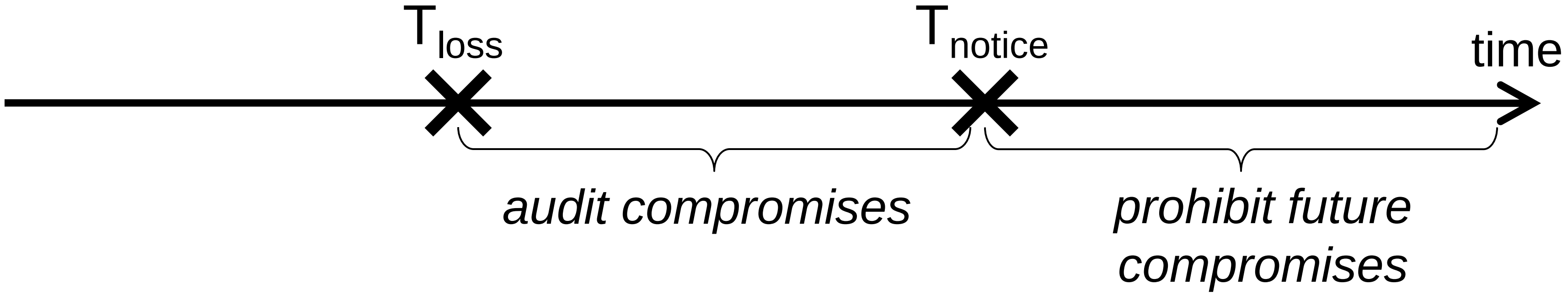
Problems with Encryption

- Problem 1: Encryption can and does fail
  - Security and usability are at odds
    - "Johnny can't encrypt" [Whitten, Tygar '99]
    - Users set guessable passwords, reuse them [Gaw, Felten '05] , [Imperva '10]
    - Users leave smartcards inside laptops [Caveo '03]
  - Hardware attacks are possible
    - Cold-boot attacks [Halderman , Schoen, Heninger, et.al. '08]
    - TPM attacks [Anderson, Kuhn '96]

- Problem 2: When encryption fails, it fails silently
  - User cannot know whether or not the data was compromised

Our Goals

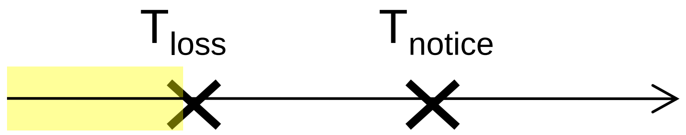
- After a device is stolen or lost, we want to:
  - □ know whether or not the data was compromised
  - □ know exactly what data was compromised
  - □ prohibit future compromises once the user detects theft

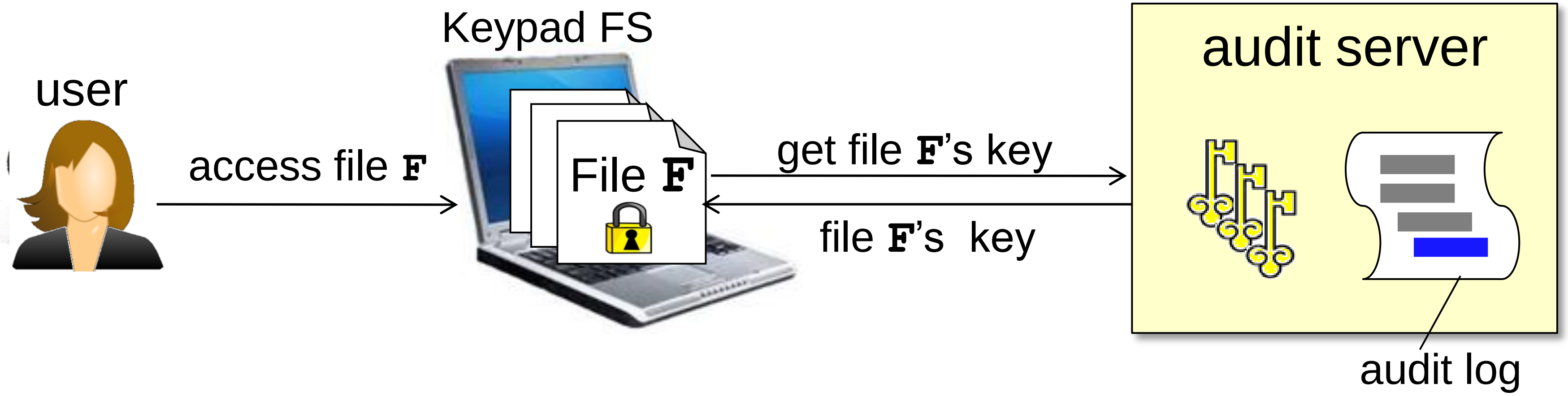


- We want strong auditing guarantees:
  - □ Even if thief turns off network (unlike Apple MobileMe, Intel AT)
  - □ Even if thief tampers with the device
  - □ Without impacting usability

Keypad: An Auditing Encrypted File System

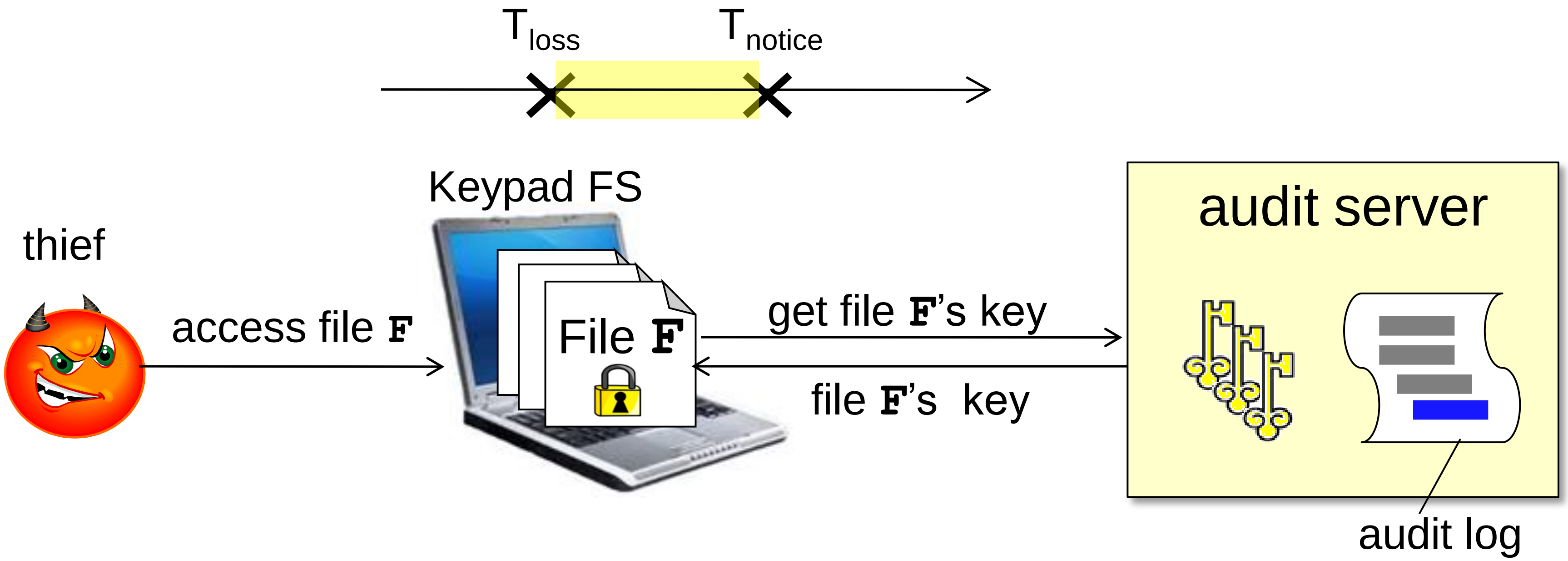
- Provides fine-grained remote access auditing and control
- Core idea: Force remote access auditing with encryption
  - □ Encrypt each file with its own random key
  - □ Store the keys on a remote server, which logs all accesses





Keypad: An Auditing Encrypted File System

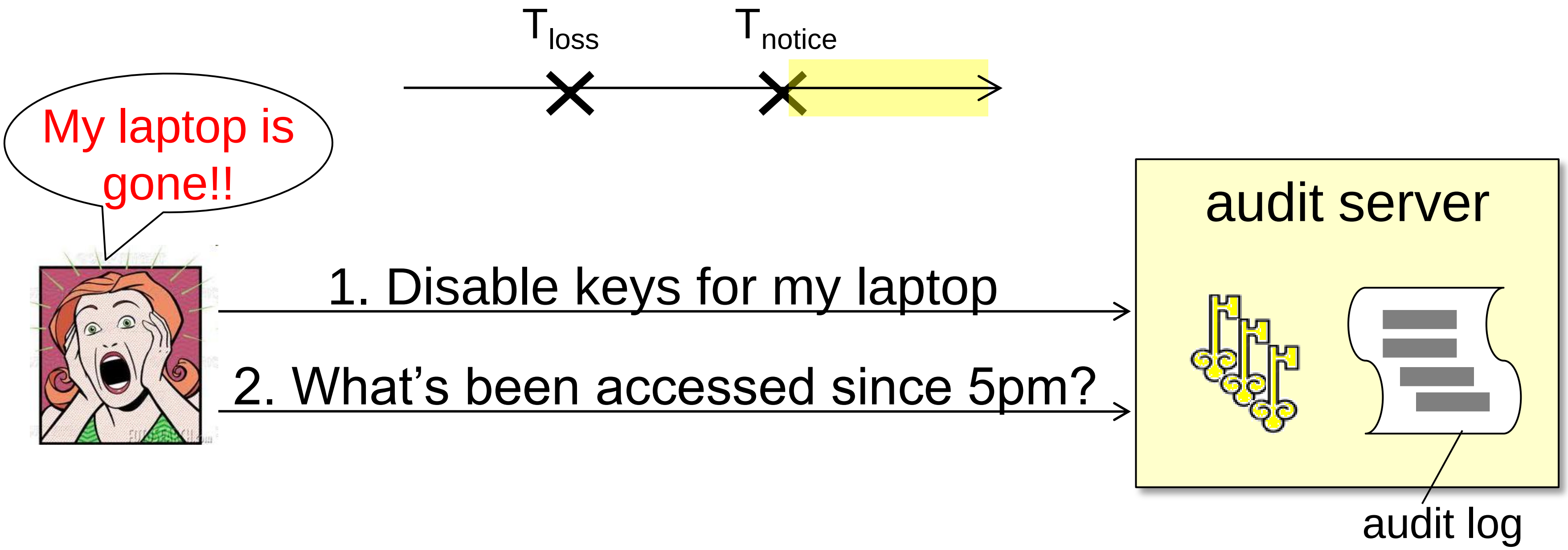
- Provides fine-grained remote access auditing and control
- Core idea: Force remote access auditing with encryption
  - Encrypt each file with its own random key
  - Store the keys on a remote server, which logs all accesses



Any compromise leaves a forensic trail on the server.

Keypad: An Auditing Encrypted File System

- Provides fine-grained remote access auditing and control
- Core idea: Force remote access auditing with encryption
  - Encrypt each file with its own random key
  - Store the keys on a remote server, which logs all accesses



T_{loss} T_{notice}

My laptop is gone!!

1. Disable keys for my laptop

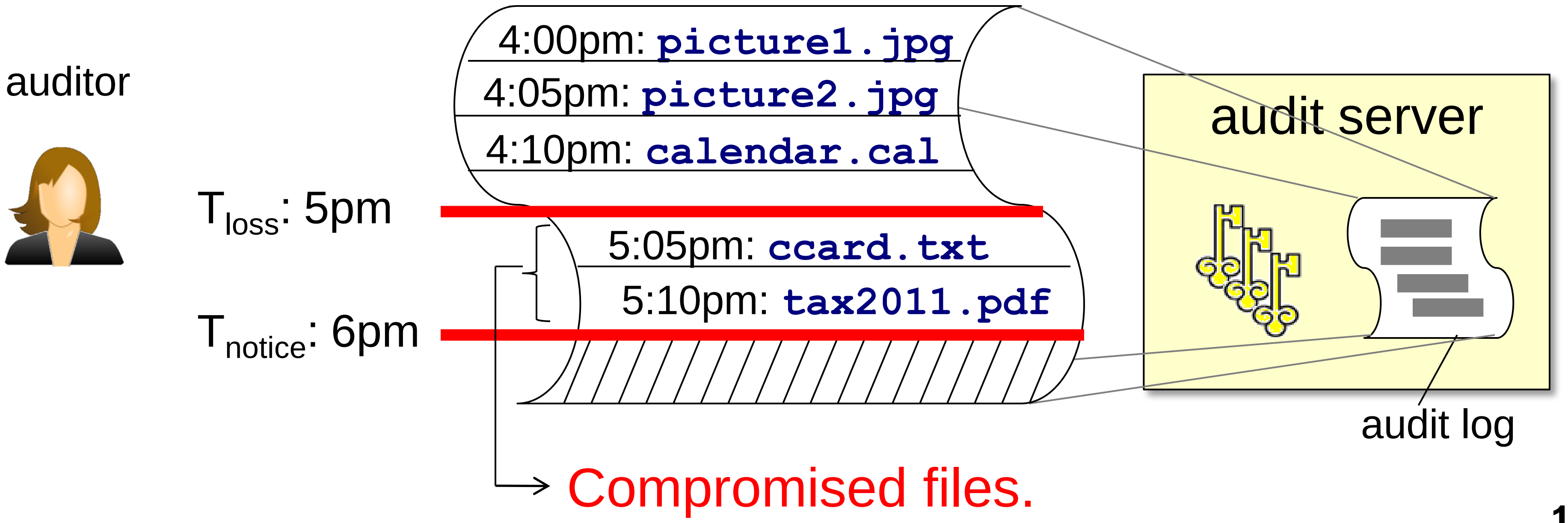
2. What’s been accessed since 5pm?

audit server

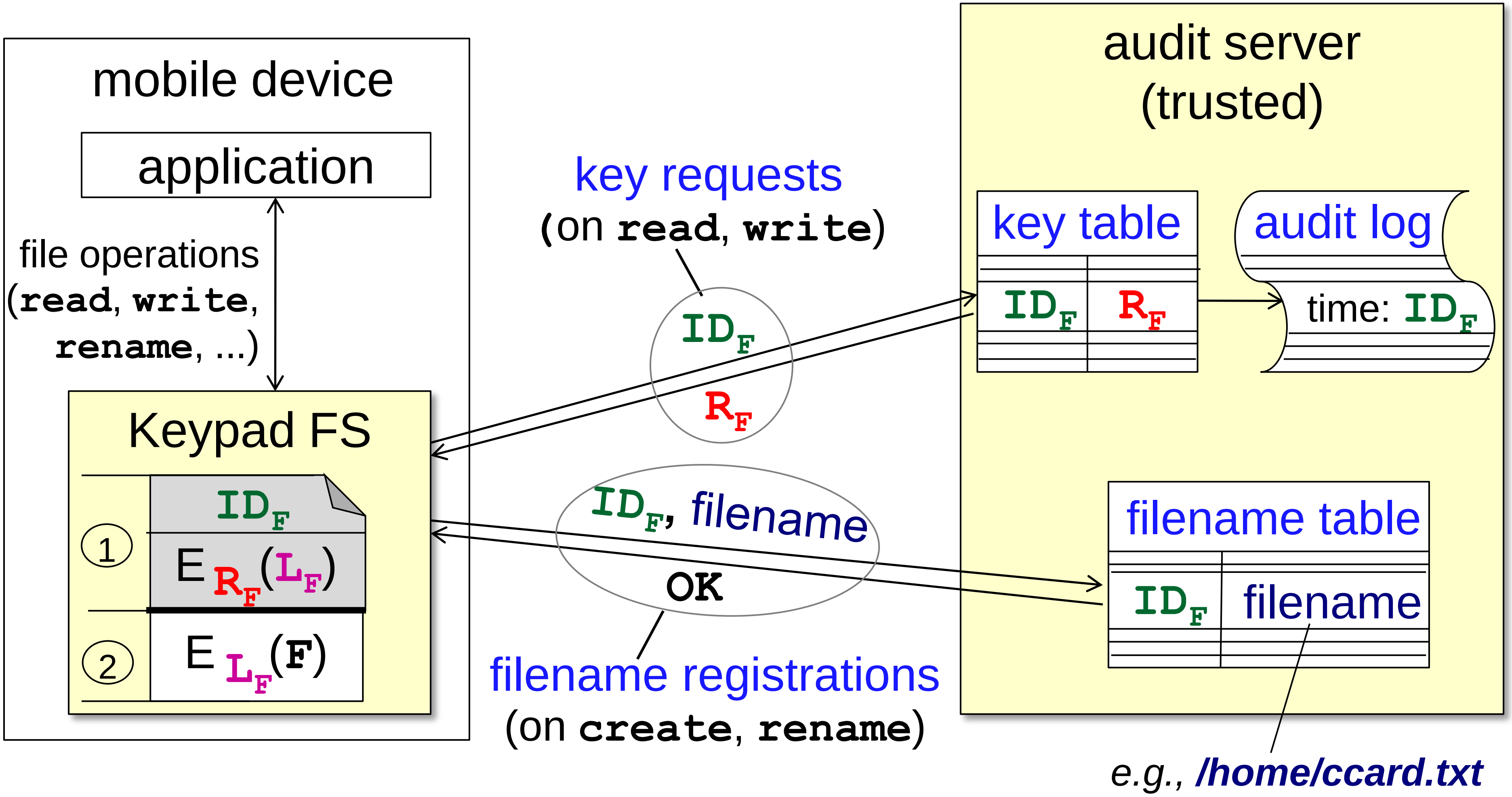
audit log

Keypad: An Auditing Encrypted File System

- Provides fine-grained remote access auditing and control
- Core idea: Force remote access auditing with encryption
  - □ Encrypt each file with its own random key
  - □ Store the keys on a remote server, which logs all accesses



Keypad's Architecture



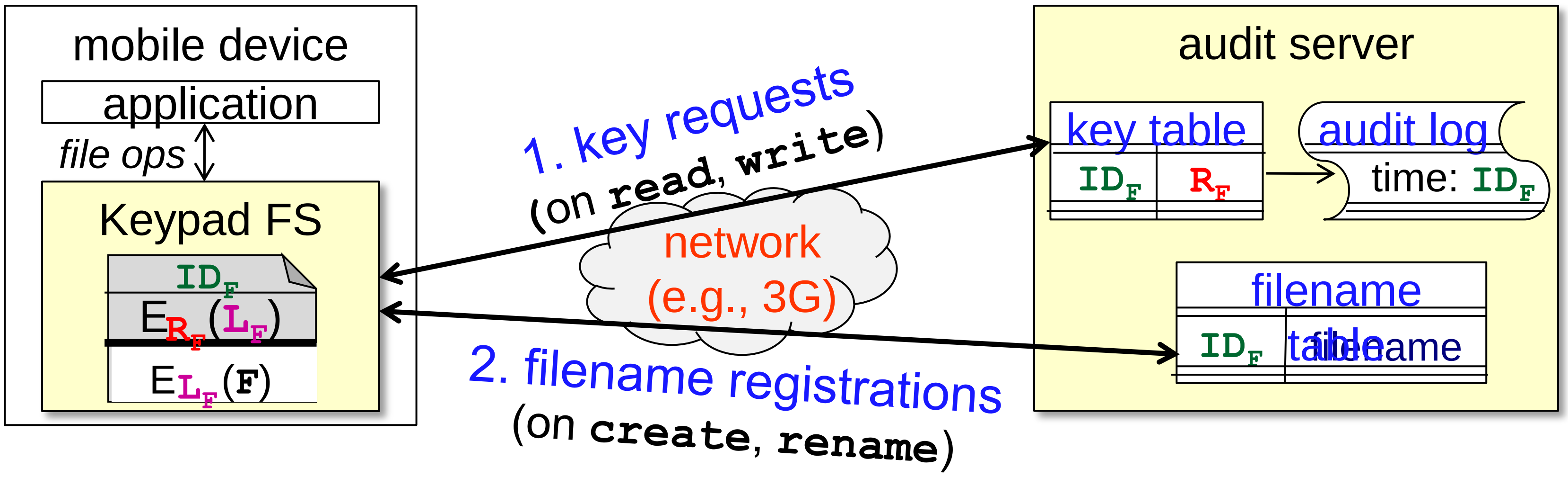
① *file F's internal header (ID_F is a long, random number)*

② *file F's contents, encrypted with symmetric key L_F*

Huge Practical Challenges

- Challenge 1: Performance over mobile networks
  - Mobile networks have huge RTTs (e.g., 300ms for 3G)
- Challenge 2: Disconnected data access
  - Disconnection is rare (WiFi, 3G, 4G), but it happens
- Keypad's design includes novel techniques to address challenges while preserving strong auditing semantics
  - Short-term key caching
  - Localized key prefetching
  - Key preallocation
  - Key derivation
  - Limited scope/granularity
  - IBE-based filename registrations
  - Device pairing
  - …

Challenge 1: Performance Over Mobile Networks

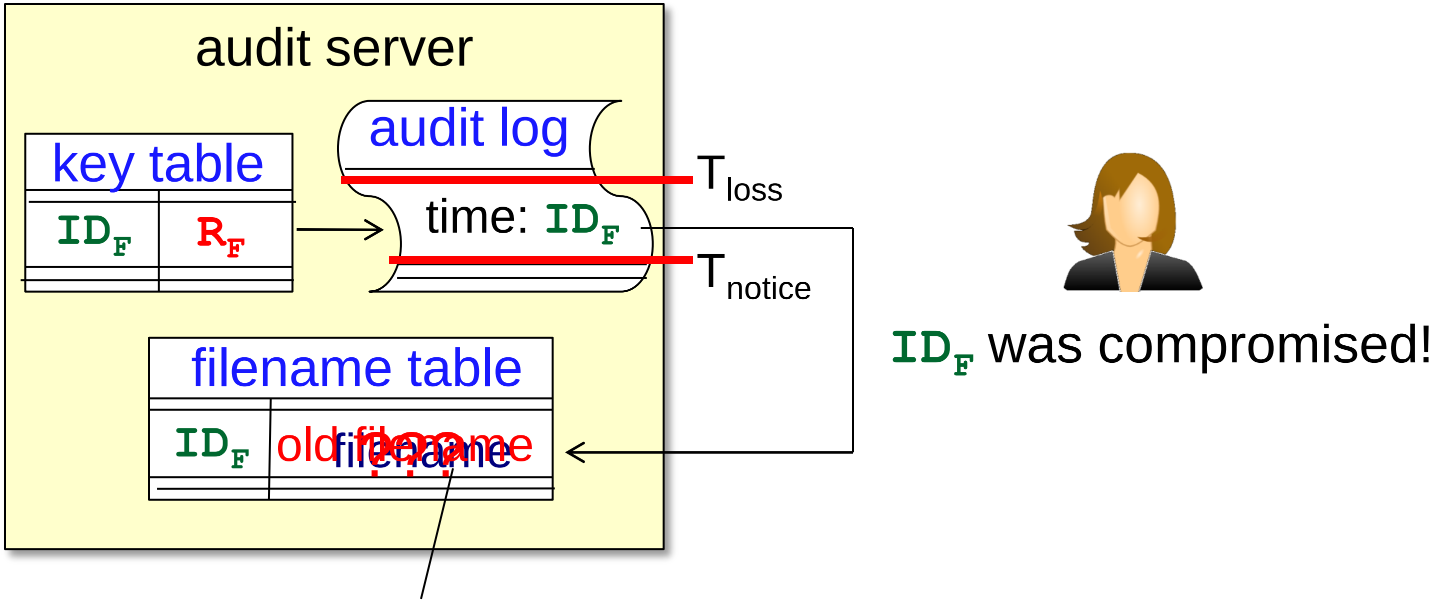


1. Optimizing key requests:
   - □ Standard techniques: key caching, prefetching, preallocation, …
   - □ 2 order of magnitude improvement (compilation now takes 8 min)

2. Optimizing filename registrations:
   - □ After key optimizations, 56% of the time goes to registrations!
   - □ Next: optimizing filename registrations with strong semantics

Name Registrations: Semantics/Performance Tradeoff

- Strong semantics requires up-to-date filenames on the server for any compromised file **ID**



audit server

key table

$\mathtt{ID_F}$ $\mathtt{R_F}$

audit log

time: $\mathtt{ID_F}$

T_{loss}

T_{notice}

filename table

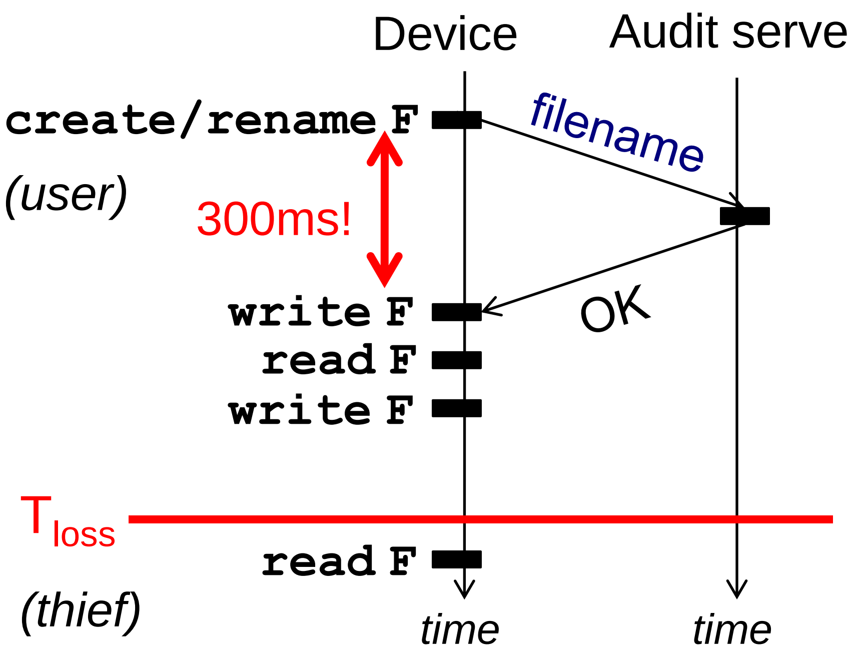
$\mathtt{ID_F}$ old filename filename ???

$\mathtt{ID_F}$ was compromised!

e.g.: /tmp/IRS_form.pdf instead of
/home/my_taxes.pdf

Two Options for Filename Registrations

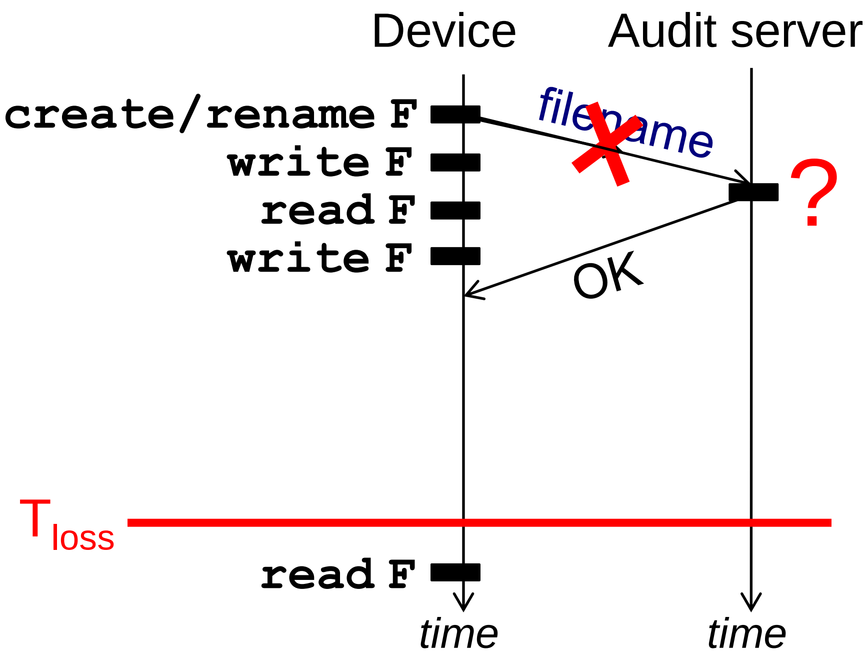
Blocking registrations



Good semantics

Poor performance

Non-blocking registrations



Poor semantics

Good performance

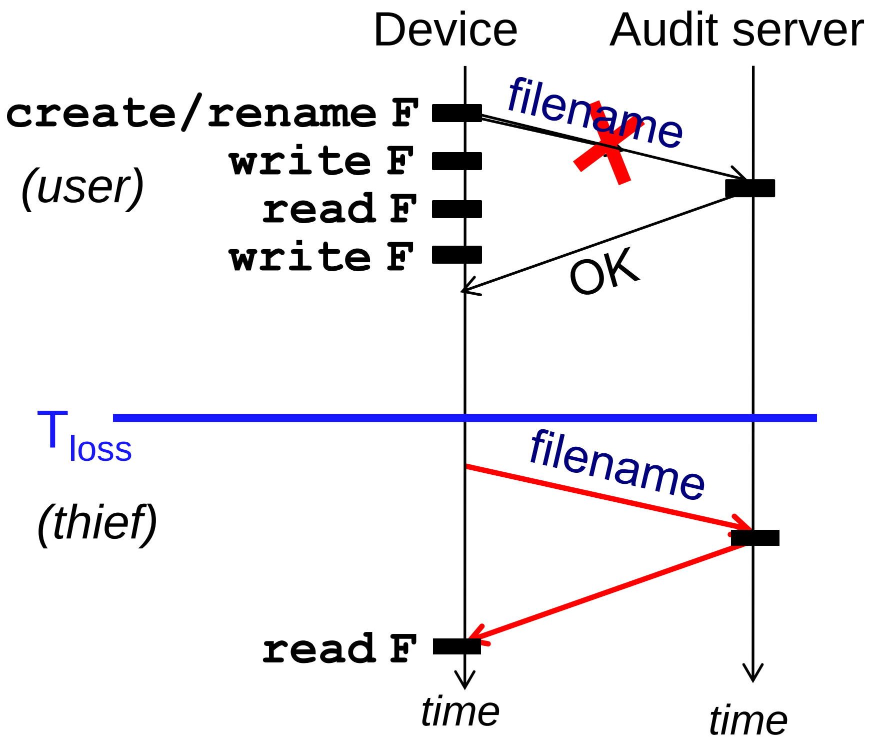
How to Have Your Cake and Eat It Too

Our Idea:

- Do non-blocking registration
- But if it fails, force the thief to reveal the filename in order to access the file!

The Challenge:

- How do we force the thief to tell us the filename?
  - Thief might lie to mislead user
  - E.g., declare /tmp/download instead of /home/ccard.txt



Good semantics
Good performance

One Solution: Identity-based Encryption (IBE)

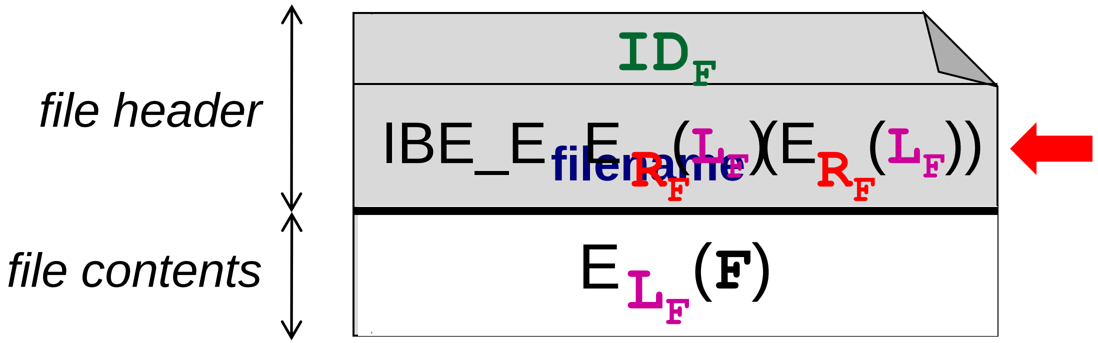
- We develop a protocol for both efficient and secure filename registrations that relies on IBE

- IBE background [Boneh, Franklin '01]:
  - A client can encrypt data using any string as the public key
  - A designated server can produce a private key for any public key
  - To decrypt, client must provide public key to get private key

- Our protocol uses the filename as the public key

IBE-Based Filename Registrations (Intuition)

- Wrap encrypted L_F with IBE using filename as the public key*
  - □ Only the audit server can compute the private IBE key



- Thief must provide the true filename to server to obtain L_F!
  - □ Lying about the filename prevents file access
- For performance, we cache L_F in memory for one second
  - □ Normally, user workloads will not block waiting for private key

* *A nonce is also included in the IBE public key for security.*

Summary of Filename Registration Protocol

- Our protocol enables both efficient (non-blocking) filename registrations and strong semantics
- Idea: Force the thief to reveal the true name of a file in order to access it
- We use IBE in a unique way:
  - It is typically used for confidentiality
  - We use it for auditing

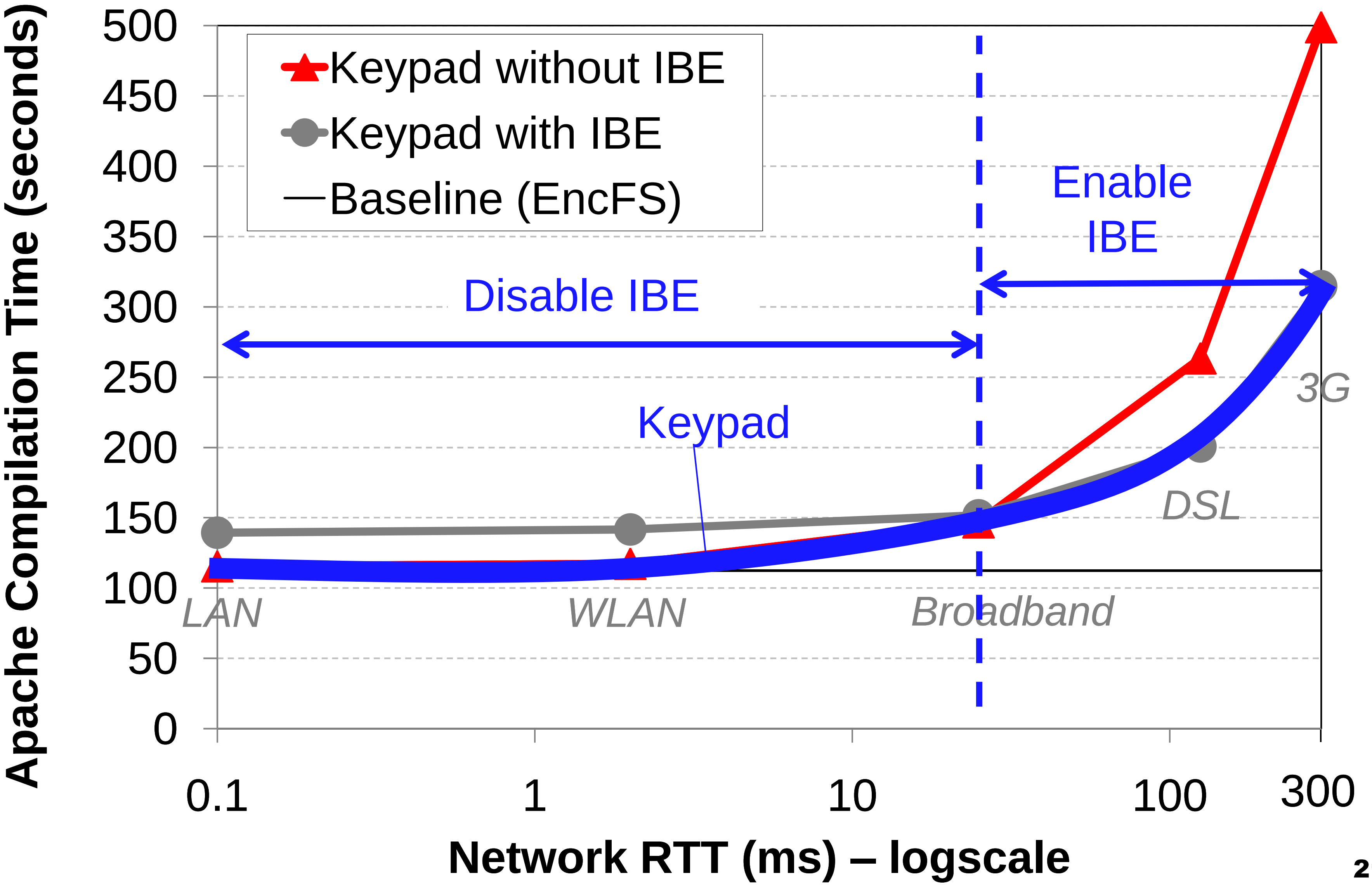
Keypad Implementation

- We built the Keypad file system on Linux
  - We augment EncFS with auditing and remote control
  - The audit server runs on Google's AppEngine

- I used Keypad for several weeks with 3G emulated latencies
  - Overall experience was positive – Keypad absorbs most latency

- We measured Keypad with many workloads and metrics
  - Microbenchmarks, Andrew benchmark, popular applications

IBE's Performance Impact

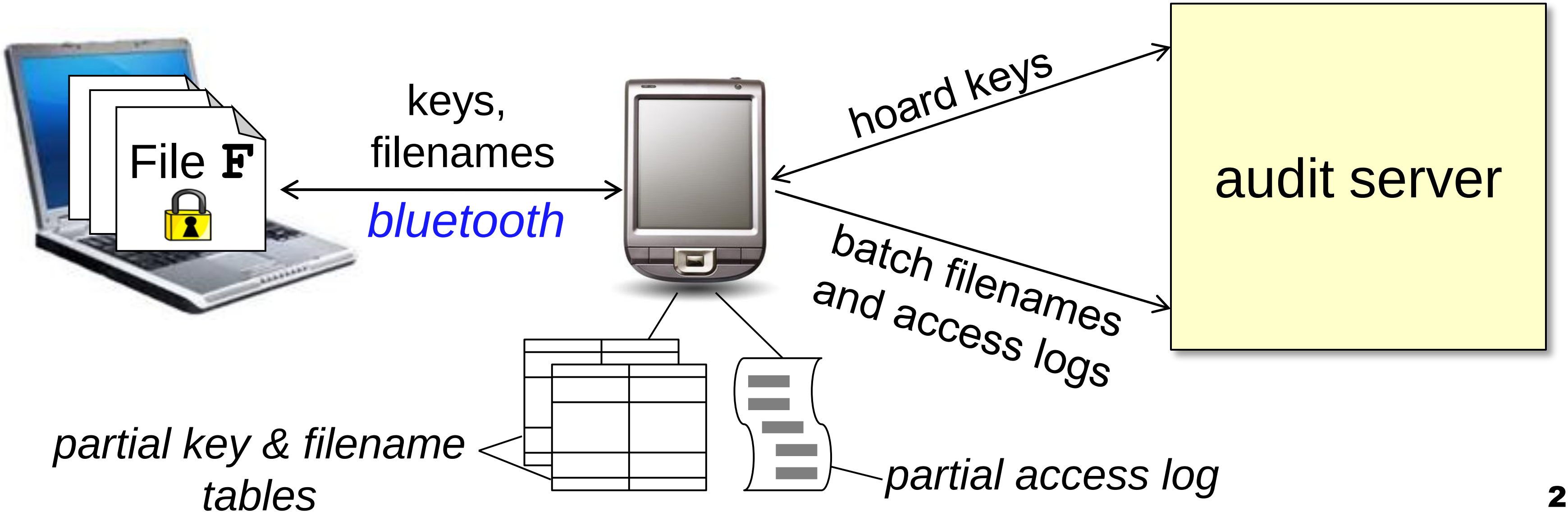


So, Is Keypad Practical?

Application	Task	Time (seconds)		
		Baseline (EncFS)	Keypad	
			WiFi	3G
OpenOffice Word Processor	Launch	0.5	0.6	4.6
	Save as	1.4	1.4	2.0
	Open	1.7	1.8	2.1
Firefox	Launch	3.7	3.8	8.8
	Save a page	0.7	0.7	1.3
	Open tab	0.2	0.2	0.2
Thunderbird	Launch	1.3	1.3	3.1
	Read email	0.3	0.4	1.9
	Quit	0.2	0.2	0.2
Evince PDF Viewer	Launch	0.1	0.1	0.1
	Open document	0.1	0.1	0.4
	Quit	0.0	0.0	0.0

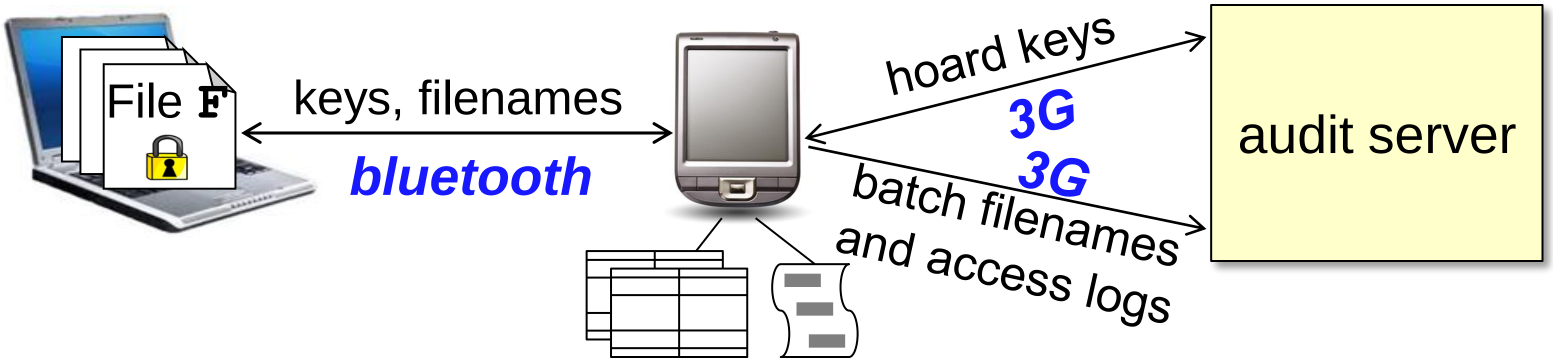
Challenge 2: Audited Disconnected Access

- Keypad's design relies on network connectivity for auditing!
- Our observation: today's users carry multiple devices
  - E.g.: laptop, phone, iPad, Kindle
- Paired-device Keypad extension uses one device to enable audited disconnected access on another device



Paired-Device Implementation

- We modified Keypad to support device pairing
  - Simple Python daemon runs on an Android Nexus One phone
- Bonus: device pairing can improve 3G/4G performance ☺
  - Bluetooth is one order of magnitude faster than 3G
  - We designed strong-semantics performance improvements
  - 44% improvement on 3G over the results we have seen before



Summary

- Traditional encryption systems fail silently
- Keypad enhances encrypted file systems with:
  - Fine-grained file access auditing after theft
  - Remote access control even in the absence of network
- Our use of cryptography is unique
  - Auditing instead of confidentiality