Secure Messaging

CS 161: Computer Security

Prof. Raluca Ada Popa

Nov 29, 2016

Announcements

- Homework 3 due Dec 2
- Final Dec 15, 11:30-2:30

End-to-end encryption

- Encryption decryptable only by the ends
- Intermediary don’t receive decryption keys, do not see plaintext, and hence cannot read or modify the data
- SSL is an example

Some history: Lavabit email encryption

(not end-to-end encryption)

Shutdown to protect user privacy:

- “My company, Lavabit, provided email services to 410,000 people, according to news reports – and thrived by offering features specifically designed to protect the privacy and security of its customers. I had no choice but to consent to the installation of their device, which would hand the US government access to all of the messages – to and from all of my customers – as they travelled between their email accounts other providers on the Internet.”
- “But that wasn't enough. The federal agents then claimed that their court order required me to surrender my company's private encryption keys, and I balked. What they said they needed were customer passwords – which were sent securely – so that they could access the plain-text versions of messages from customers using my company's encrypted storage feature.” (Lavabit founder)

KIM ZETTER SECURITY 03.17.16 5:30 PM

A GOVERNMENT ERROR JUST REVEALED SNOWDEN WAS THE TARGET IN THE LAVABIT CASE

FBI–Apple encryption dispute

From Wikipedia, the free encyclopedia

The **FBI–Apple encryption dispute** concerns whether and to what extent courts in the United States can compel manufacturers to assist in unlocking cell phones whose contents are cryptographically protected.[1] There is much debate over public access to strong encryption.[2]

In 2015 and 2016, Apple Inc. has received and objected to or challenged at least 11 orders issued by United States district courts under the All Writs Act of 1789. Most of these seek to compel Apple "to use its existing capabilities to extract data like contacts, photos and calls from locked iPhones running on operating systems iOS 7 and older" in order to assist in criminal investigations and prosecutions. A few requests, however, involve phones with more extensive security protections, which Apple has no current ability to break. These orders would compel Apple to write new software that would let the government bypass these devices' security and unlock the phones.[3]

End-to-end encryption for messaging


future tense THE CITIZEN'S GUIDE TO THE FUTURE | APRIL 5 2016 5:33 PM

WhatsApp Adds End-to-End Encryption for Its 1 Billion Users

By *Lily Hay Newman*

257 84 6

Sensitive information receiving the protection it deserves.

WhatsApp

CADE METZ BUSINESS 04.05.16 11:00 AM

FORGET APPLE VS. THE FBI: WHATSAPP JUST SWITCHED ON ENCRYPTION FOR A BILLION PEOPLE

ANDY GREENBERG SECURITY 07.08.16 8:30 AM

'SECRET CONVERSATIONS:' END-TO-END ENCRYPTION COMES TO FACEBOOK MESSENGER

TextSecure

- The protocol at the basis of Whatsapp encryption and Facebook messenger
- Created by Moxie Marlinspike

former head of the security team at Twitter and founder of Open Whisper Systems; also sailor, captain, shipwright

The Pink Stool

A prank war in a low moment.

Hypothermia

I made a series of mistakes that culminated in the worst sailing accident of my life, and almost took me to the bottom of the ocean.

Challenge-A-Day

There is a Pittsburgh punk tradition called "Fun-A-Day." For the entire

Let's recreate TextSecure

- Together! It will be an interactive lecture!

- Real security protocols can be quite **complex! So pay attention**
- **I simplified/adapted it** for this lecture, retaining some security components but not others.

Why not just SSL for chat?

- Users don’t have public keys, certificates
- Chat conversations last for a long time, even when parties are not online any more
- Other extensions: group chat

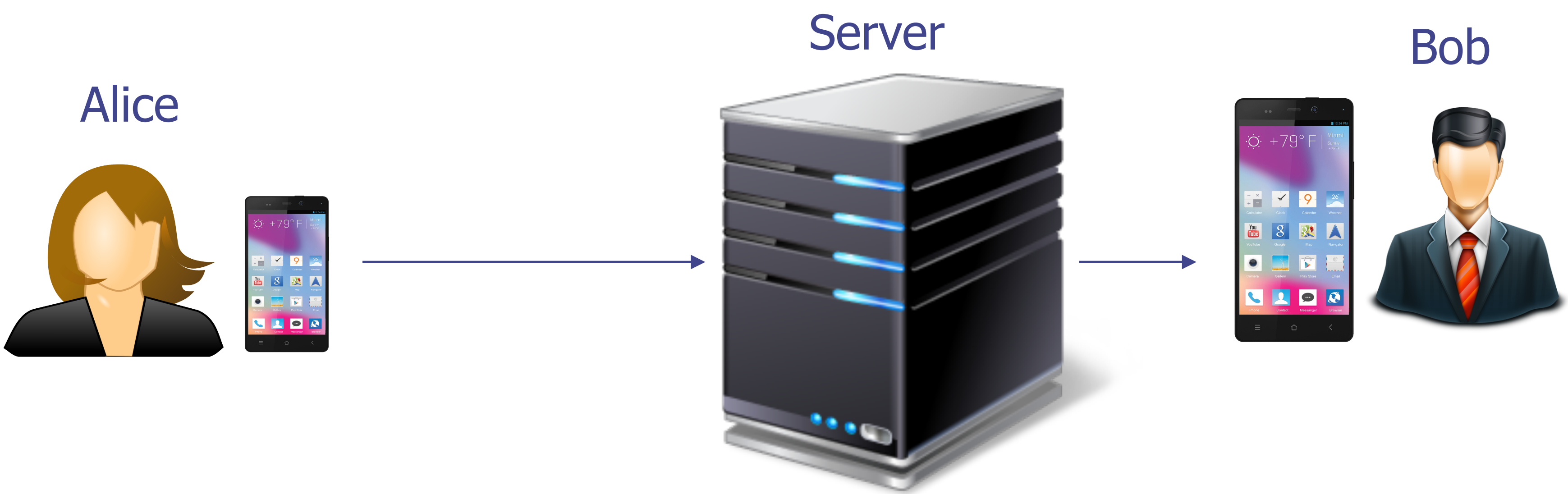
TextSecure

Phases:

1. Registration
2. Setup conversation
3. Converse

Setup

Consider the context of Whatsapp, where users have phone numbers



Goal: only Alice and Bob should see these private messages. The server or other intermediary should not be able to see them.
Server threat model: could be malicious attacker (man-in-the-middle) with the exception of a few times during setup when assumed just passive on-path

Phase 1: Registration

What property would the server/client like to ensure during registration?
What attack could a user perform?

Registration process

- Authenticate server to client
- Authenticate client to server (to prevent impersonation of a user by another):
  - Server sends a token to user's phone and expects the user to send that token back – checks that user indeed owns that phone
- Provide some public keys to the server

On projector

Step 2: conversation setup in TextSecure*

simplified and adapted to the class

Conversation setup: agree on key between Alice & Bob

Server

Alice

Bob

a, PK_A =

b, PK_B =

hi Bob, PK_A

Bob | PK_B

$Enc_A[K_E, K_{MAC}]$

K_E, K_{MAC}

Opportunistic encryption:

K_E, K_{MAC}

K_E, K_{MAC}

Server can be MITM attacker.

Assumes server is passive (not MITM) attacker during key-setup, but can be MITM attacker during conversation.

TOFU (Trust on first use) - conflicts with changing keys

Upon first key exchange, assume no MITM attacker

Each user keeps track of PK of another user,

if PK of Bob changes, Alice's client flags warning

Alice calls Bob, to check that $K_1 = K_2$

Assume: attacker cannot dynamically change voice
but it can listen to it

A: $H(K_1) = h_1$ B: $H(K_2) = h_2$

$\xleftrightarrow{h_1, h_2}$

TextSecure (simplified) Short Authentication Strings

Alice | S | Bob

$a, g^a = PK_A$ | MITM | $b, g^b = PK_B$

Diffie-Hellman key exchange

$\xrightarrow{g^a}$

$\xleftarrow{g^b}$

$K_{shared} = g^{ab} \bmod p$ | $K_{shared} = g^{ab} \bmod p$

Alice $\xrightarrow{g^a}$ e $\xrightarrow{g^e}$ Bob

Alice $\xleftarrow{g^f}$ f $\xleftarrow{g^b}$ Bob

g^{af} $\quad g^{eb}$

$K_1 = g^{af}$ $\qquad g^{eb} = K_2$

$f = eba^{-1}$

To prevent MITM, Alice & Bob can compare shared keys

$K_1 \overset{?}{=} K_2$ $\quad \boxed{\text{yes} \Rightarrow \text{no MITM}}$ $\quad g^{af} = g^{eb}$

else attacker solved discrete log

no $\Rightarrow$ MITM

Suppose $\boxed{(g^{af})^{f^{-1}} = (g^{eb})^{f^{-1}}}$

$g^a = g^{ebf^{-1}}$

$\boxed{f, e} \Rightarrow f^{-1}$

$(g^{af})^{f^{-1}} = g^a$

$\log_{g^b} g^a$

$(g^a)^x = g^b$ $\quad x =$

3 types of keys:

long-lived keys (a, g^a) , (b, g^b)

pre-keys (medium-lived T) $(x_{a,0}, g^{x_{a,0}})$, $(x_{a,1}, g^{x_{a,1}})$ ...

ephemeral keys (session keys t): $(\overline{x_{a,0}}, g^{\overline{x_{a,0}}})$ ...

So far: users perform Diffie-Hellman key exchange without MITM

② Setup of conversation

A

$S = \text{Adv.}$

B

talk to Bob $\longrightarrow$

pre-keys g^a, g^b

$g^{x_{a,0}}, g^{x_{a,1}} \ldots (T)$

$g^{x_{b,0}}, g^{x_{b,1}} \ldots$

$x_{b,z}$

b

$\longleftarrow g^{x_{b,z}}, g^b$

choose $\overline{x_{a,0}}$ \$

$g^{\overline{x_{a,0}}}, g^a \longrightarrow$

Compute:

$g^{x_{b,z} \cdot a}$

$g^{b \cdot \overline{x_{a,0}}}$

$g^{x_{b,z} \cdot \overline{x_{a,0}}}$

concatenated $\longrightarrow$ K_{ENC}, K_{MAC}

KDF

key derivation

K_{ENC}, K_{MAC}

$\longleftarrow$ exchange messages using $\longrightarrow$

K_{ENC}, K_{MAC}

forward secrecy = if $\mathcal{A}$dv compromises the long-lived Key of one user, attacker should not be able to decrypt messages in the past

If $\mathcal{A}$dv has a, but not b, $\mathcal{A}$dv cannot compute K_{ENC}, K_{MAC} because it cannot compute $g^{b \cdot \overline{X_{a,0}}}$

If $\mathcal{A}$dv has $X_{b,z}$ but not a $\Rightarrow$ $\underline{g^{X_{b,z} \cdot a}}$

cannot compute $g^{b \cdot X_{a,0}}$ because it does not have either b or $\overline{X_{a,0}}$

3) Conversation

(pay me 50, 1)

~~(pay me 20, 1)~~

Alice

k_{Enc}, k_{MAC}

Bob

k_{Enc}, k_{MAC}

m

ctr_B

$ctr_A = 0$

$C = Enc_{k_{Enc}}(m; ctr_A, \text{"from Alice"}), \; MAC_{k_{MAC}}(C)$

$ctr_B, \ldots$

Attacks	Defense
replay	Counter

- won't accept message with same counter

Short Authentication Strings

a

g^{ab}

b

g^{ab}

hash(g^{ab}) = 8fa2438432eba2…

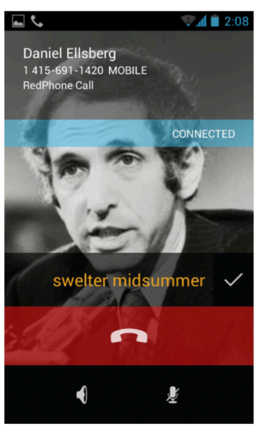
hash(g^{ab}) =
8fa2438432eba2…

What is a more usable way of checking they agreed on the same key?

What is a more usable way of checking they agreed on the same key?

hash(g^{ab}) = 8fa2438432eba2…

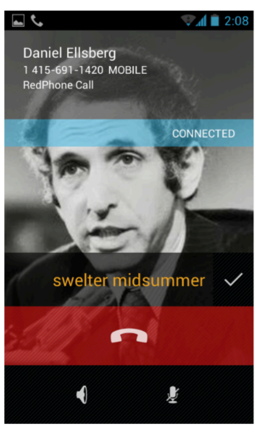
hash(g^{ab}) = 8fa2438432eba2…



Inattentive user

hash(g^{ab}) = 8fa2438432eba2…

hash(g^{ab}) = 8fa2438432eba2…



yes

Is your message Sweden Summer?

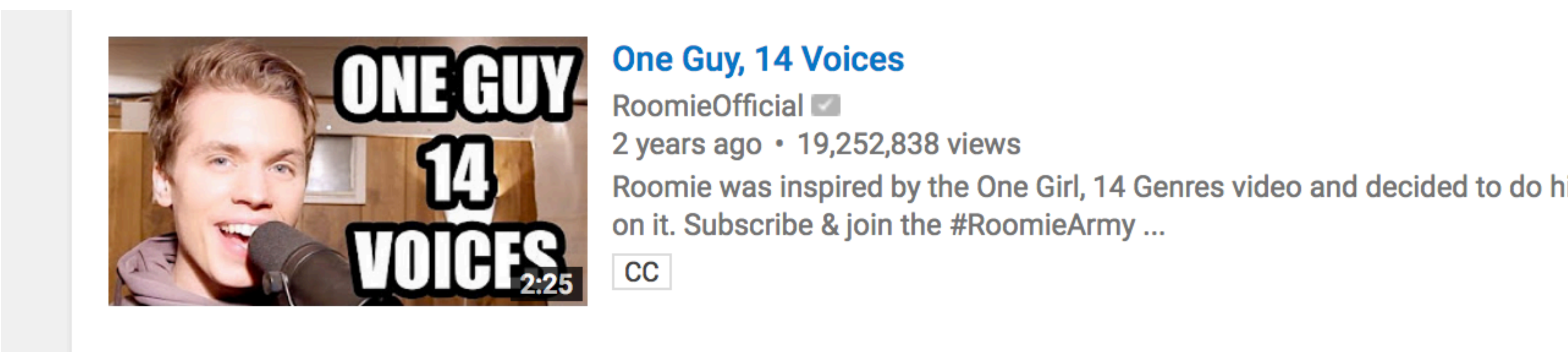
How can we fix the problem of an inattentive user?

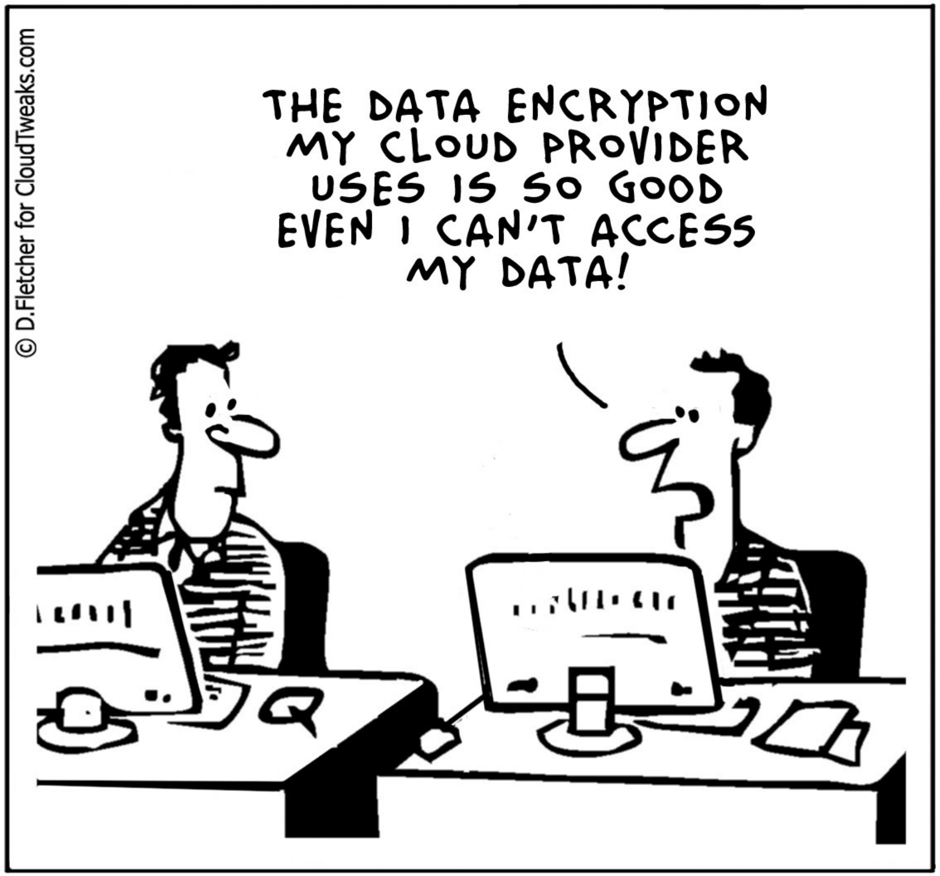
- Ask users to type in what the other is saying and have the client check it

Any other ways the attacker can attack this?

It can actually fake phone calls from recordings..

- Shirvanian and Saxena`14 show that using a small number of samples of a user’s voice, audio can be synthesized that is indistinguishable from the genuine user’s voice





© D.Fletcher for CloudTweaks.com
THE DATA ENCRYPTION MY CLOUD PROVIDER USES IS SO GOOD EVEN I CAN'T ACCESS MY DATA!

Questions?