Building Trustable Cloud Systems

Cullen Jennings,
Oct 20, 2013

v3

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 23, 2014.

Copyright Notice

Summary

- Problem with cloud services: existing service providers keep a whole bunch of data, which governments/attackers can then collect from the service providers

- Strategy: limit the data exposed to service providers

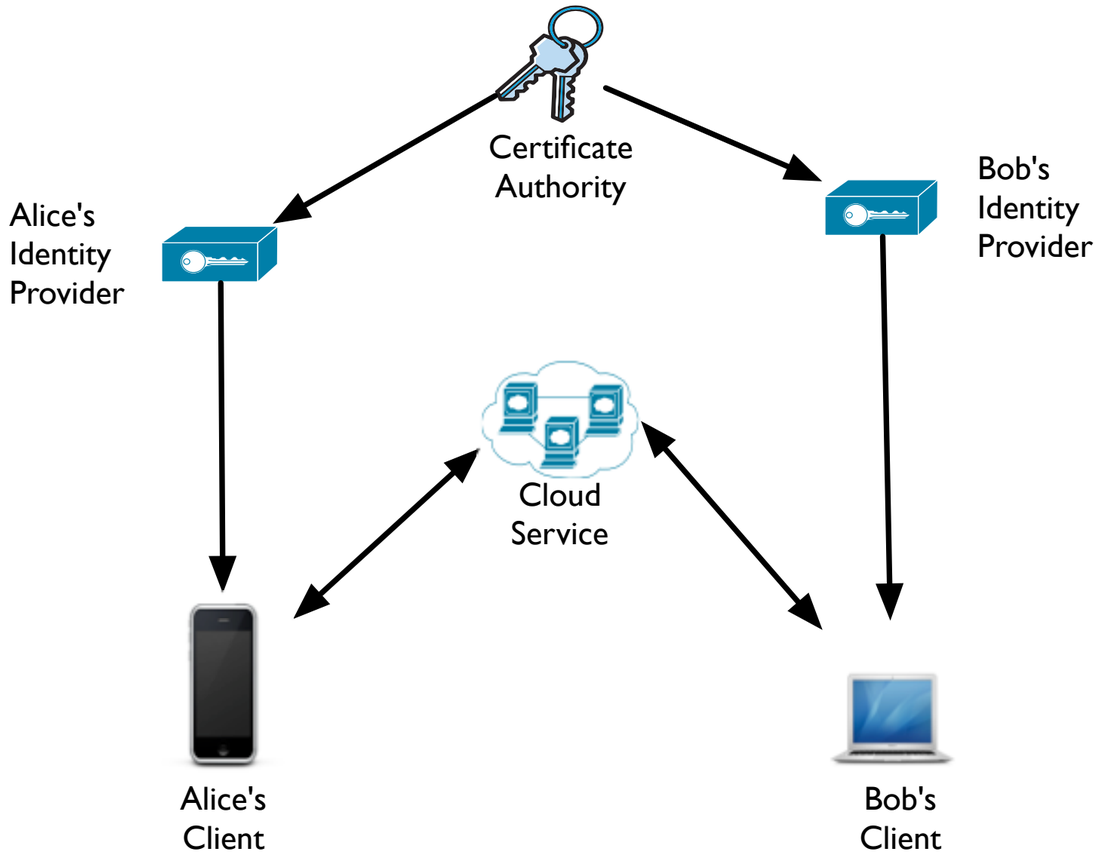
- Technical approach:
- Service provider has only the envelope with encrypted data
- A user trusted identity provider facilitates user key access and key management between devices

Goals

- Housley Criteria: Be able to detect if your communications have been compromised
- Support voice, video, instant message, stored messages, file sharing...

Strategy

- Cloud Service sees only encrypted data and envelope information
- All users have public/private key
- The user's Identity Server manages the users private keys and provides public key to others
- Identity providers authenticate to others using Certificate from the Certificate Authority
- Content is encrypted by clients and the information to decrypt it is encrypted with the public keys of all the authorized users



Threats

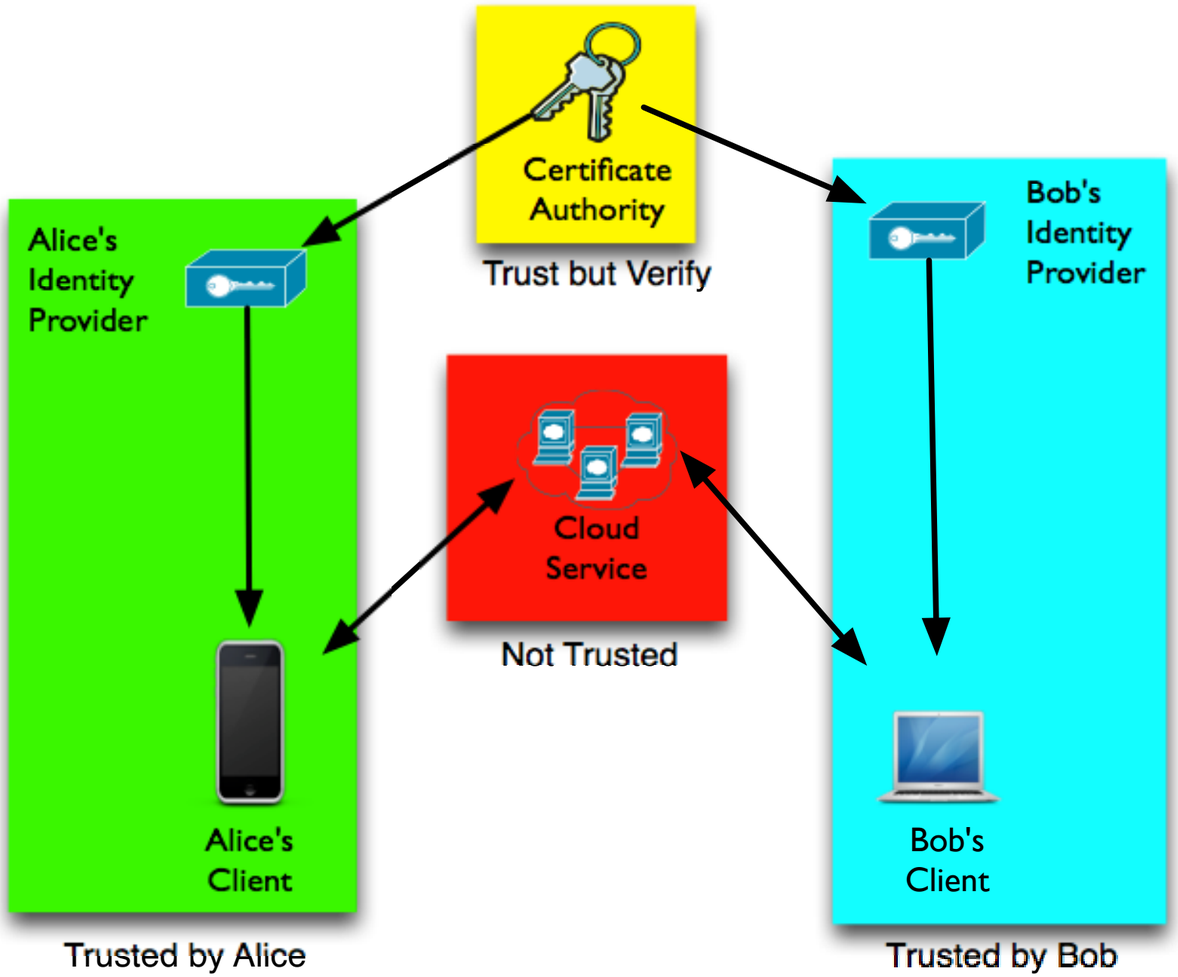
- 1. Governments obtaining user data from service providers
  - Mitigation: The architecture in slides for limiting SP access
- 2. TLS and VPN access *somehow* (crypto break or key leaks)
  - Not really a protocol issue
  - If crypto, CFRG
  - If key leaks, can we do anything?
- 3. Large-scale passive traffic collection
  - Encrypt + anonymize what's not encrypted
  - Encryption: TLS Everywhere (no work needed, just deploy?)
  - Anonymization: Overlay routing (e.g., TOR, P2P with RELOAD)
- 4. Possible badness in PRNGs
  - See rusting randomness slide

Threats Not Addressed

- High const surveillance of a small set of user
- Protecting the social graph information of who is talking to who
- Protecting all the meta and "envelope" data
  - Getting the envelope data is surveillance and a serious privacy concern

Trust

- Alice trust her Identity provider to with private key
- The CA is "honest" is that you can tell if it issues your certificate to someone else but there is no way to stop it from doing that
- The cloud service never gets keys to see encrypted data



Encrypted Data Content

- Each piece of content belongs to a group. Each group has one content owner
- Data touched by the cloud is encrypted
- The content encryptions keys are encrypted using the public keys of all users authorized to read this content
- If others user can modify this data, the signature key for this content is encrypted with the public key of all users authorized to write this content
- The content is encrypted and signed and bundled with all the relevant meta data
- List of authorized user to read/write a piece of given content is managed by identity server for the content owner

Identity Provider (IdP)

- You have to trust your identity provider. Looking at IdPs enterprises can run for employes and IdPs users can run on for themselves
- Each user's device authenticates to IdP to get users private key
- IdP provides public keys to others
- IdP authenticates by having certificate for domain it serves
- IdP for a user is discovered using domain name of the user identity
- Each device talks to IdP to find out list of public keys for any groups that users owns. IdP provides API to manage group membership.

Certificate Authority

- Provides TLS style certificates
- Provides an audit log such that anyone can check which certificates it has created
- If the CA creates bad certificates, which it can, the security of the whole system can be compromised but the goal is to be able to detect this

Key Revocation

- Relies on the Identity Providers and Cloud Service cooperating to get rid of the old key
- If a private key for a user is compromised, it is replaced with a new by the Identity Provider and the Cloud Service is informed to deprecate old key
- For any content that the old compromised private key could access, the Cloud Service ask the Identity Provider that owns that content to provide new meta data for that content with the new private key

Key Continuity

- Goal is some belt and suspender security taking advantage of key continuity. Given user will like just click through big warnings, system also notifies administrators
- Any times a client detects a key has changed for a user, it can inform the user, identity provider, and cloud service to try and detect compromises
- Any time the Certificate changes for an Identity Provider or Cloud Service the Client can inform the user and Identity provider

Peer to Peer Services

- Often the media can be transferred directly between the clients without going though the cloud.
- Voice and Video and the Data Channel in WebRTC all support this

- Example:
  - http://sharefest.me/ uses WebRTC to directly transfer files between web browsers

Search

- Any time that the Cloud Service gets new content, it provides that content to all the Identity providers for users that can read that content along with an URI to be associated with the content
- Each Identity Provider can index that content for future search as it has the private keys to decode it
- Clients can perform a search using their Identity Provider, the a search terms, and a context controlled by a match expression on the URI to get the set of Cloud Service URI that match the search

Switched Audio / Video Conferencing

- Modern audio / video conferencing systems mostly don't decode, mix, encode the video. Instead they take the "relevant" media streams forward them on to the all the participants and let them mix them
- "Relevant" for audio often means 3 to 5 loudest speakers
- "Relevant" for video often means high resolution video of the most recent 1 to 3 speakers, the presentation video, and low resolution of a selected set of 7 to 25 participants.
- RTP has ways to cary how loud each speaker is in a way that not encrypted so that a switched conference can do all the above without the keys to decrypt the media content
  - (Yah, aware of arguments you can reconstruct content based on loudness - we could consider encrypting this with separate key from media)

Trusting Software

- Most modern applications have an auto update, or close to auto, with software that is signed by the developer
- Could a court order force the developer to insert code on the next update to compromise the clients private keys? How to protect against this?

Push Notifications

- Push notification have become an important part of mobile applications yet they create a large source of unencrypted information flowing thought just a few providers such as Apple

- Should provide a way the mobile phone having an pubic/private key pair where the public key is given to mobile applications on the phone and can be used to encrypt messages that can be sent over push notifications and decrypted and displayed by the mobile OS without needing the applications on the mobile to be running

Address Hiding

- This is hard. Lets do the easy stuff first

Trusting Crypto

- Hard to decide what is trustworthy - perhaps CFRG can help
- What is clear is crypto agility is important
- Perhaps the world needs a Suite Z for folks that don't like Suite B

Trusting Randomness

- Constant source of problems in implementations
- Even harder in "IoT" type devices

Trusting OS/Hardware/Device Drivers

- If you can't trust the OS, all the software with root, and all the device drivers, not to mention the thing plugged into you lighting connecting, very hard to protect against that

Trusting DNS

- Sorry, can't trust this yet

Standardization Needed

- Verification of all CA certs issued
- IdP Discovery: POSH?
- IdP Authentication of Client
- IdP API for management of IdP
- IdP API for pub/priv keys
- IdP API for search
- KeyRevocation API
- Key Continuity API
- Formats for encrypted objects and metadata
- Crypto Recommendations

Acknowledgments

- Design motivated by SiRiUS (Goh, Shacham, Modadugu & Boneh)
- Thanks to EKR, Richard Barnes