Why Joanie Can Encrypt: Easy Email Encryption with Easy Key Management

John S. Koh
(koh@cs.columbia.edu)

Steven M. Bellovin
Jason Nieh

<EURO/SYS'19>

Columbia | Engineering
The Fu Foundation School of Engineering and Applied Science

Email is valuable to you and your enemies.

US & Canada

18 revelations from Wikileaks' hacked Clinton emails

27 October 2016

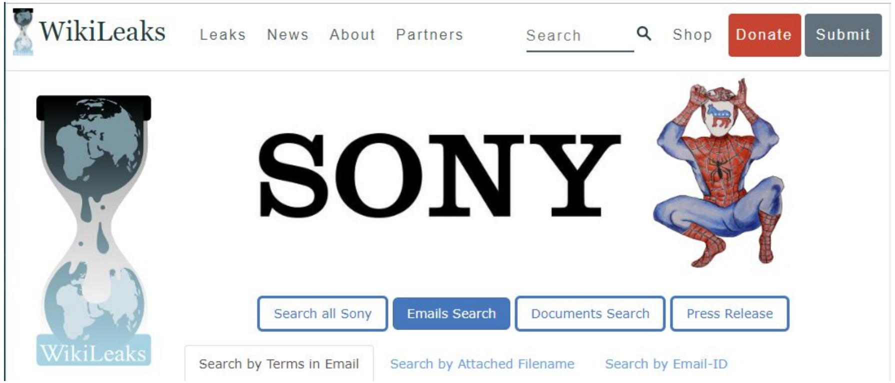
Share

US Election 2016

The New York Times

SUBSCRIBE | LOG IN

In Hacked D.N.C. Emails, a Glimpse of How Big Money Works



WIRED   Edward Snowden's Email Provider Shuts Down Amid Secret Court

Email storage: massive, free, convenient, and the perfect target.

- A **single successful attack** is enough to compromise **all** your email.
- The problem worsens over time: we are keeping more and more email.
- Mail services provide:
  - Massive amounts of free storage (e.g. Gmail gives 15 GB for free).
  - Reliability and backups --- it’s all hosted in the cloud.
  - Easy access from all of your devices.
- The cost of a single **account or server compromise** therefore is increasing.

Solutions exist but are rarely used.

End-to-end encrypted email.

Pretty Good Privacy
(PGP)

Secure/Multipurpose
Internet Mail Extensions
(S/MIME)

Difficult and confusing. Rarely used in practice.

End-to-end encryption: the reason why “Johnny” can’t encrypt.

- Why **Johnny** Can’t Encrypt [Whitten, 1999]
  - **Johnny** 2 [Garfinkel 2005]
  - Why **Johnny** Still Can’t Encrypt [Sheng 2006]
  - Why (Special Agent) **Johnny** (Still) Can’t Encrypt [Clark 2011]
  - Helping **Johnny** 2.0 to Encrypt His Facebook Conversations [Fahl 2012]
  - Confused **Johnny** [Ruoti 2013]
  - Why **Johnny** Still, Still Can’t Encrypt [Ruoti 2015]
  - Maybe Poor **Johnny** Really Can’t Encrypt [Benenson 2015]
  - Leading **Johnny** to Water [Atwater 2015]
  - Why Won’t **Johnny** Encrypt [Orman 2015]
  - Helping **Johnny** Understand and Avoid Mistakes [Ruoti 2015]
  - Can **Johnny** Finally Encrypt? [Herzberg 2016]
  - And plenty more.

Johnny hasn’t encrypted for 20 years!

End-to-end encryption is hard and complicated.

The steps for setting up end-to-end encrypted email for non-technical users:

① *"What's a keypair?"*

② *"What's signing?"*

③ *"What's a private key?"*

④ *"Which one is the public one?"*

⑤ *"Oops, I sent you my private key."*

⑥ *"How do I encrypt?"*

Concepts: Keypairs, PKI, signing, trust, CAs, key exchanges, encrypting, ...

We need to make some trade offs to gain usability.

- The state of the art is unusable but secure email that almost nobody uses.
- At the other end, we have usable email with no security that everyone uses.

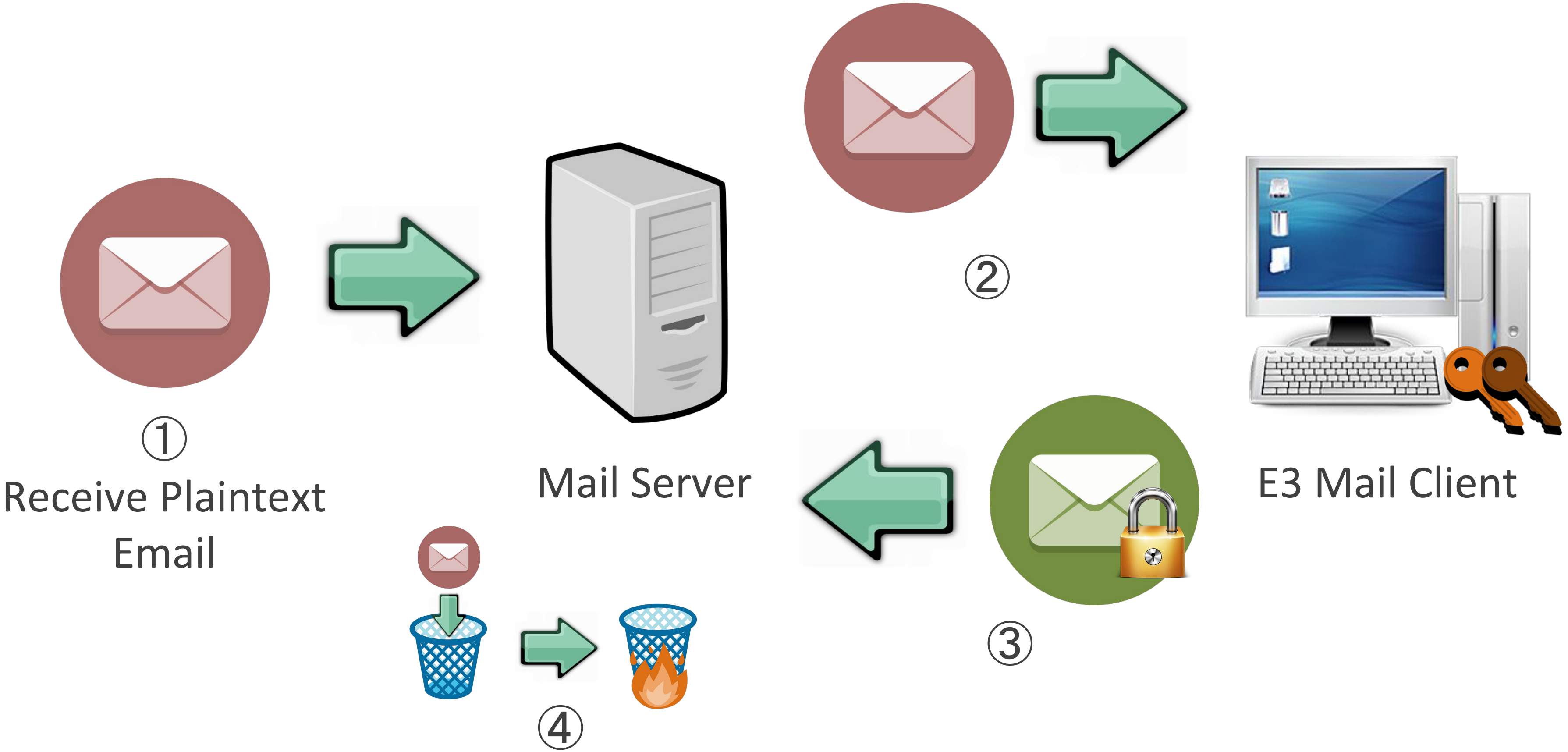
- We can trade some security for tremendous usability to fill the void.

Easy Email Encryption (E3)

Encrypt on receipt: encrypt emails when they are received.

Per-device keys: A keypair for every device.

Encrypt on receipt



Easy Email Encryption (E3): encrypt on receipt.

E3's main security guarantee:

- ✓ E3 protects all emails received **prior** to a compromise.

If a compromise happens, all the attacker sees is encrypted emails.

Trade off: E3 does not protect new emails that arrive after a compromise.

But one compromise does not give up your thousands of old emails.

Easy Email Encryption (E3): encrypt on receipt.

Users do not *send* encrypted email.

- ✓ But in-transit emails are protected by TLS. After Snowden, TLS is increasingly used.

E3's slightly relaxed guarantees allow for great gains in usability:

- ✓ No requirement to know about keys, PKI, or trusted third parties.
- ✓ No requirement to know how to use PGP or S/MIME.
- ✓ Elimination of confusing key exchanges and coordination with others.

Easy Email Encryption (E3): encrypt on receipt.

New possibilities not available in other approaches.

- ✓ Requires only client-side changes. No server or protocol changes.
- ✓ Works with any mail service including Gmail, Yahoo, AOL, Yandex, etc.
- ✓ Compatible with ad-based business model for email services like Gmail.
- ✓ Does not interfere with spam filtering and anti-virus scanning.

Encrypt on receipt is platform independent.

Encrypt on receipt emails are encrypted in standard formats.

- Encrypted messages either use the standard PGP or S/MIME formats.
- E3 emails can be read on unmodified mail clients that support encryption.
    - Just need to run a separate E3 app to configure the client for the user.

Easy Email Encryption (E3)

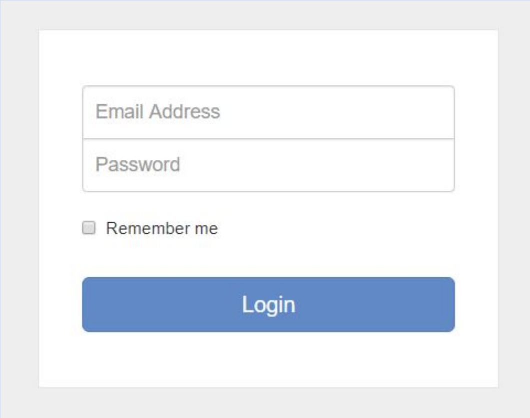
Encrypt on receipt: encrypt emails when they are received.

Per-device keys: A keypair for every device.

E3 is not immune to key management concerns.

E3 Mail Client

An E3 client differs a little from a regular client.



1. Ask for email credentials.

(Regular Mail Client)

2. Generate self-signed keypair.
3. Encrypt received emails using public key.
4. Decrypt received emails using private key.

(E3 Mail Client - Not seen by user)

Secure email has two critical problems: **multiple devices and key recovery.**

Existing key management approaches have problems.

- Even technical people don't know how to handle private keys.

superuser

How to manage GPG keys across multiple systems?

superuser

Is it ok to share private key file between multiple computers/services?

superuser

Moving PGP Keys

5

Easy: *

```
gpg --export my_key -o my_public_key.gpg
gpg --export-secret-key my_key -o my_secret_key.gpg
```

Then:

```
gpg --import my_public_key.gpg
gpg --allow-secret-key-import --import my_secret_key.gpg
```

* Easy?

Having multiple devices used to be a pain point.

- Existing key management approaches fall apart when using multiple devices.
  - People don’t know how to move private keys around.
  - People don’t know how to backup and recover their private keys.
- **But E3’s unique approach turns multiple devices into a strength.**
  - PDK streamlines key management and key backups/recovery.

Most users have multiple devices in some form.

- Most users **personally own multiple devices**.

- But some users do only have one personal device.
  - And a **trusted family member or friend** with at least one device. (This counts as multiple devices.)

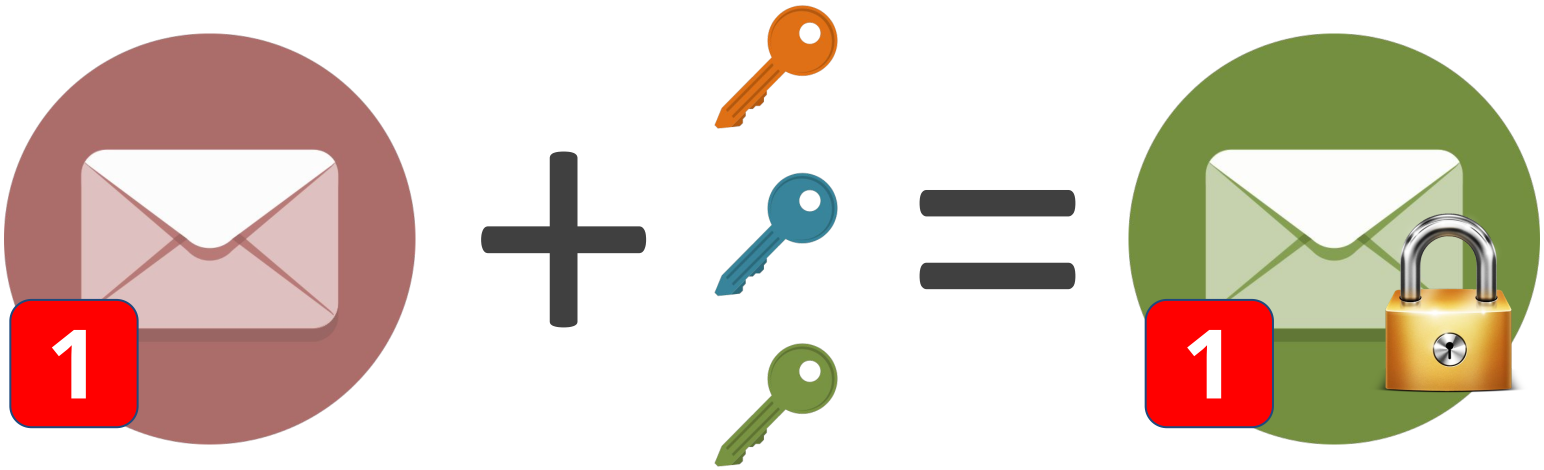
- If a user **really, really** only has a single device:
  - Backup encrypted copy of keypair in the cloud, print out recovery key.
    - Apple and Microsoft use similar approaches already.

  - **Or,** users who backup their devices can recover without a recovery key. (These aren't PDK.)

The inspiration for PDK: multiple recipients.

Emails can be encrypted to multiple recipients.

In other words, **a single email can be encrypted to multiple public keys.**



Per-Device Keys (PDK): the user's perspective.

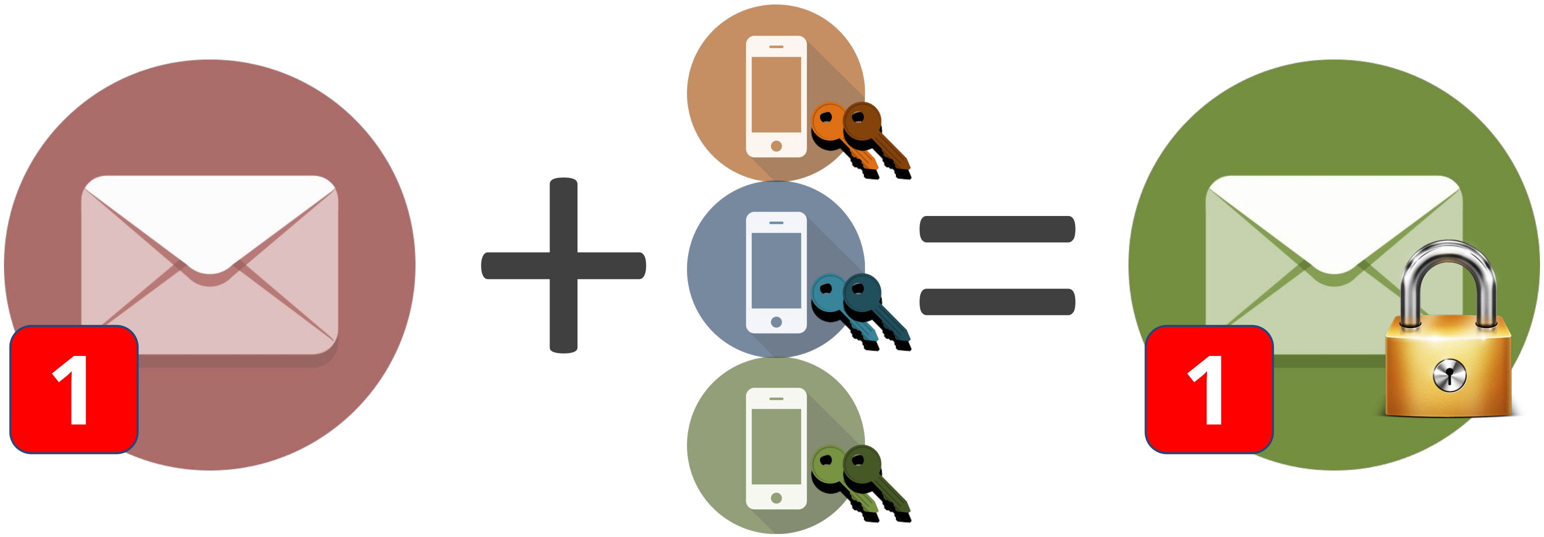
Users don't know "public keys."
They know computers, laptops, smartphones, tablets, etc.
These devices can be **their own** or **those of someone they trust.**

So **this is what we show to users**:

Per-Device Keys (PDK): the simple technical perspective.

Users don’t need to know about the keys. But encrypted email uses keys.

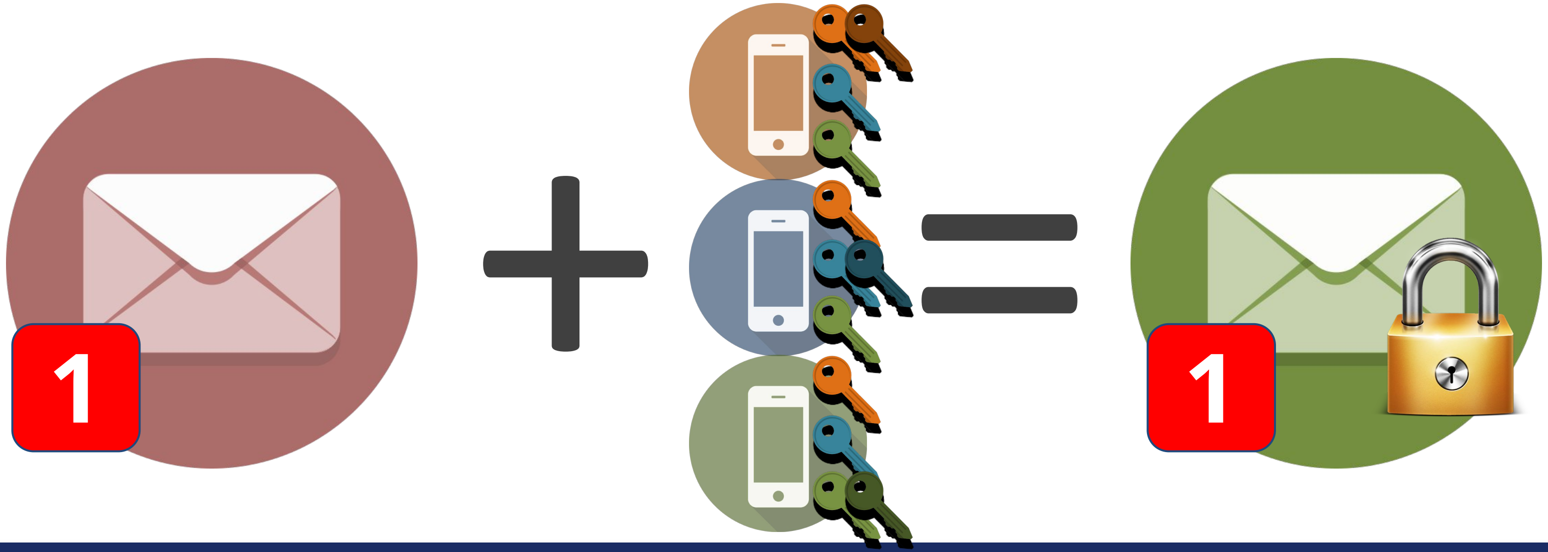
So what it looks like is a unique public ( ) and private ( ) key per device:



Per-Device Keys (PDK): the technical perspective.

Users need to read email on every device.

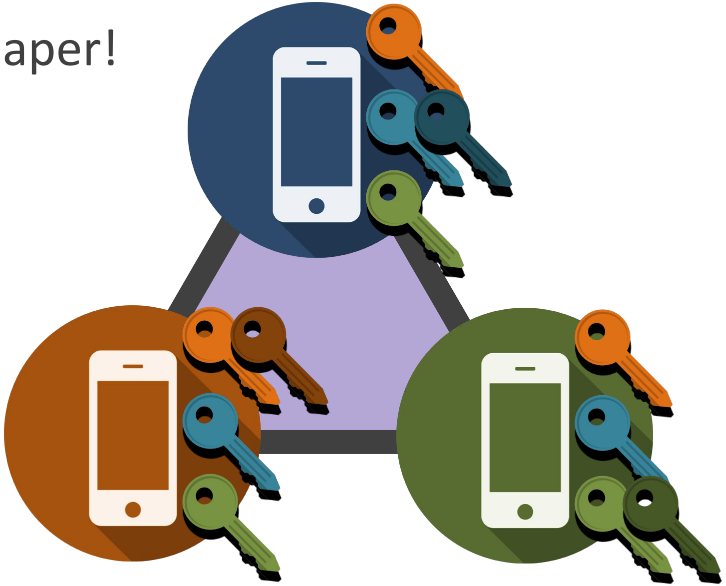
Every device knows about all devices’ **public** keys, distributed transparently.
The only private key each device knows is its own.



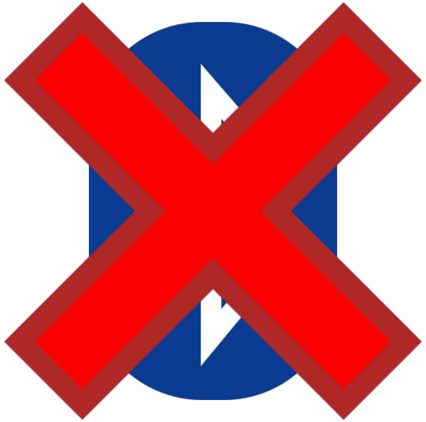
Per-Device Keys (PDK): public key distribution.

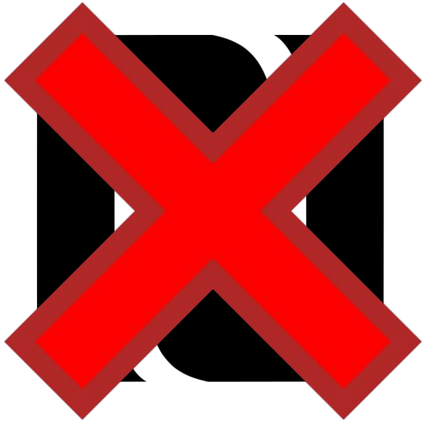
All the user sees is a **device pairing abstraction** via a **two-way verification**.
Users use an existing device to validate a new device.
It looks a little bit like Bluetooth pairing --- but it’s not.

More details in the paper!



The two-way verification process is platform independent.





- Because not all devices have Bluetooth, NFC, or short-range wireless tech.
- It uses the user’s mailbox to securely communicate.
  - It therefore works on any device that has a screen and Internet.

E3’s experience is just like a regular mail client.

Sending and receiving email on E3 is the same as a regular mail client.

Encryption and decryption happen transparently.

The difference between E3 and regular mail is in the initial PDK configuration.

Users only see the two-way verification process which is easy to use.

Encrypt on receipt + PDK = E3

What E3 does:

Encrypts emails on receipt

PDK {
- Encrypts emails using the public keys distributed among a user's devices.
- Abstracts away key management into device management.

The question: how do real users feel about setting up E3?

User study design.

- **<u>Conservative</u> study design focusing on initial mail client setup**
  - Users configured K-9 Mail on multiple Android devices.
  - Small-scale pilot study -- 8 users --- with promising results.

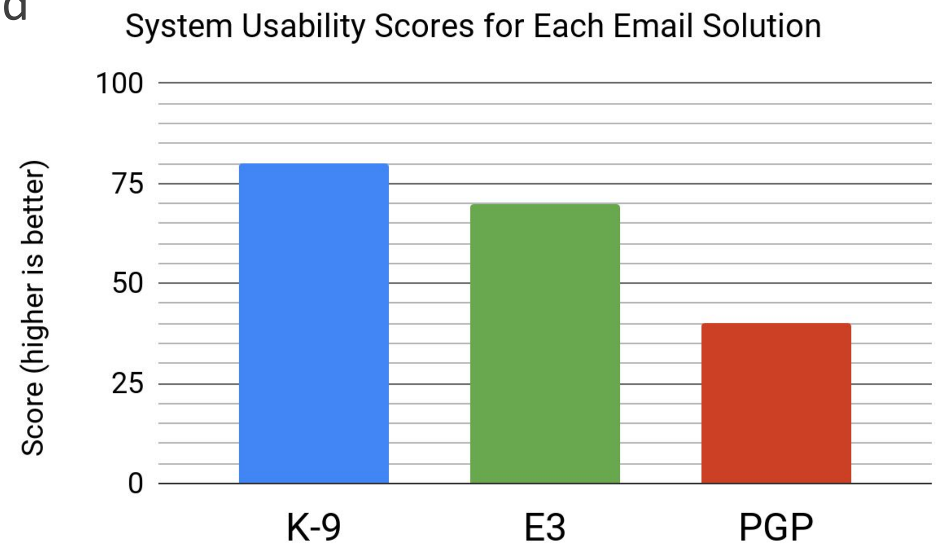
Samsung Galaxy S7

Google Nexus 7 (2013)

Huawei Honor 5X

Real people agree that E3 is easy to setup and use.

- Finish setting up each mail client, and send/read email on all 3 devices.
- **Users agree:** E3 is easy to use and much easier than PGP.
- The System Usability Score
  - Industry standard.
  - [Ruoti, 2013, 2015, 2016].



Less than 50 is unacceptable.
Greater than 70 is good.
[Bangor, 2009]

- Why **Johnny** Can’t Encrypt [Whitten, 1999]
- **Johnny** 2 [Garfinkel 2005]
- Why **Johnny** Still Can’t Encrypt [Sheng 2006]
- Why (Special Agent) **Johnny** (Still) Can’t Encrypt [Clark 2011]
- Helping **Johnny** 2.0 to Encrypt His Facebook Conversations [Fahl 2012]
- Confused **Johnny** [Ruoti 2013]
- Why **Johnny** Still, Still Can’t Encrypt [Ruoti 2015]
- Maybe Poor **Johnny** Really Can’t Encrypt [Benenson 2015]
- Leading **Johnny** to Water [Atwater 2015]
- Why Won’t **Johnny** Encrypt [Orman 2015]
- Helping **Johnny** Understand and Avoid Mistakes [Ruoti 2015]
- Can **Johnny** Finally Encrypt? [Herzberg 2016]

Johnny, 1999

Joanie, 2019

Thanks!

John S. Koh

(koh@cs.columbia.edu)

Steven M. Bellovin

Jason Nieh

What attackers actually do in the wild.

Your adversary is **not:**

- ✗ Tapping the Internet backbone.
- ✗ Eavesdropping on your (TLS-encrypted) network connections.
- ✗ Stealing your device.

Your adversary **is:**

- ✓ **Trying to steal your email account password.**
- ✓ **Issuing a subpoena for your email provider's servers.**
- ✓ **Hacking your email provider's servers.**

Can we trust mail services to be honest?

- In short: probably.

- Case Study: Google's Retention Policy [1] states that they do completely delete data when deletion is requested.

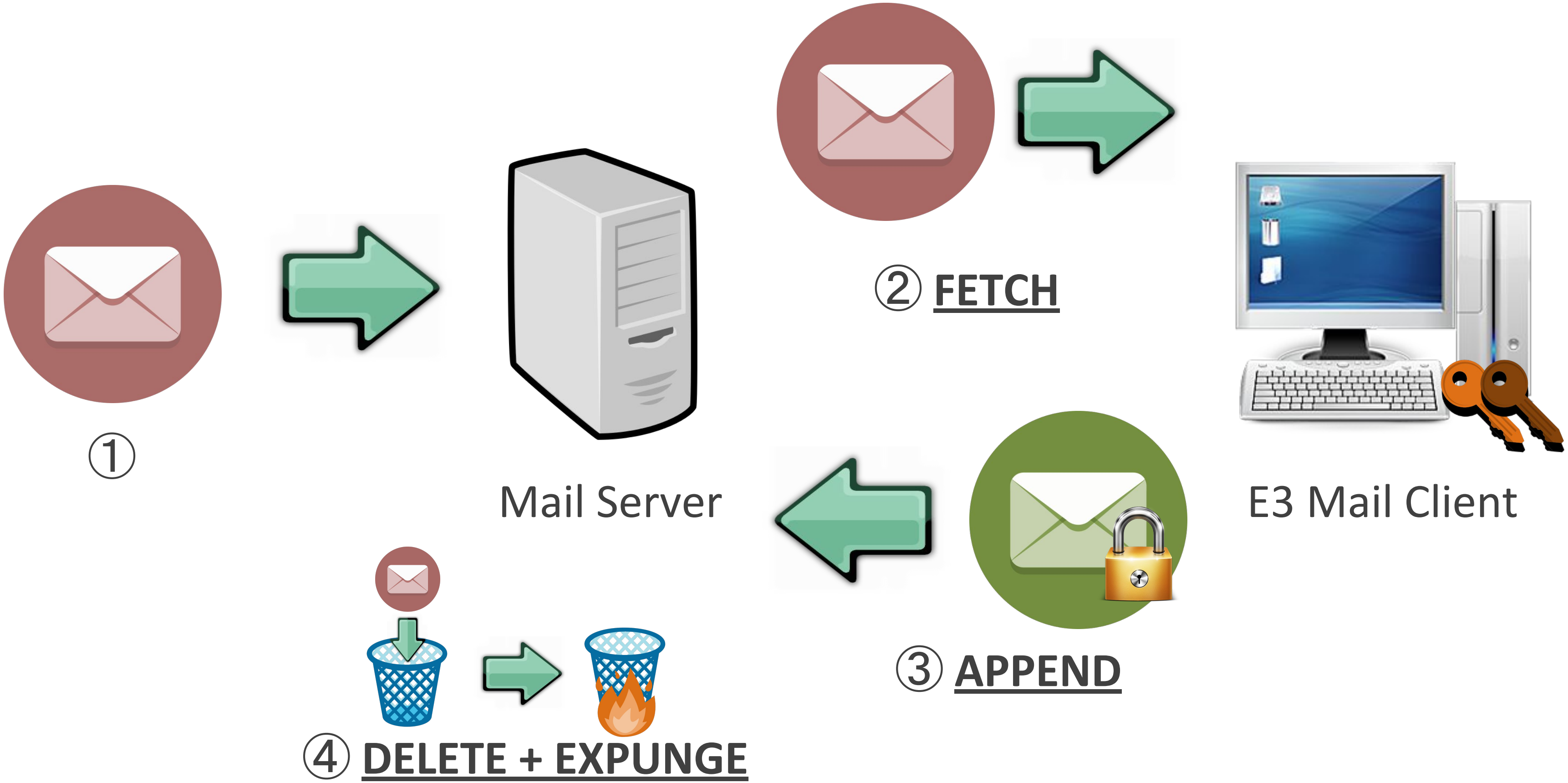
- If they were lying, they would be subject to legal action.
  - The Federal Trade Commission Act, §5 outlaws deceptive practices [2].
  - Similar laws exist in most countries.

[1] https://policies.google.com/technologies/retention?hl=en
[2] https://www.federalreserve.gov/boarddocs/supmanual/cch/ftca.pdf

E3's encrypt on receipt uses standard IMAP client commands.

- **FETCH**
  - Download messages from IMAP server.
- **APPEND**
  - Upload messages to IMAP server.
- **STORE `\Deleted`** (my shorthand: "**DELETE**")
  - Mark messages as deleted --- not actually deleted yet.
- **EXPUNGE**
  - Delete messages with the `\Deleted` flag. (i.e. "Emptying the trash.")

Encrypt on receipt: IMAP commands issued by the client.



Backups in the cloud are commonplace.

- Usage of the cloud for automatic backups is growing tremendously.

- iCloud for Apple devices:
  - **190 million** active users in October 2012 [1].
  - **782 million** active users in February 2016 [2].
    - 78% of total Apple devices --- many users own multiple devices [2].

- Android's default backup service uses Google Drive:
  - **120 million** active users in November 2013 [3].
  - **800 million** active users in March 2017 [4].

[1] https://appleinsider.com/articles/13/03/21/apples-icloud-is-most-used-cloud-service-in-the-us-beating-dropbox-amazon
[2] https://appleinsider.com/articles/16/02/12/apple-music-passes-11m-subscribers-as-icloud-hits-782m-users
[3] https://techcrunch.com/2013/11/12/gmail-users-no-longer-need-to-download-attachments-as-google-drive-gets-baked-into-the-inbox/
[4] https://techcrunch.com/2017/03/09/google-drive-now-has-800m-users-and-gets-a-big-update-for-the-enterprise/

The IMAP protocol doesn't have atomicity guarantees.

- Race conditions
  - Multiple E3 clients may race to encrypt and upload a message.
  - Normal mail clients can encounter a similar situation.
  - This can sometimes result in duplicate messages.
  - Solution:
    - Tell users to only do automatic modifications on one client, or
    - Use IMAP CONDSTORE with IMAP flags.

Avoiding race conditions with standard IMAP.

- CONDSTORE
  - Together with IMAP flags can be used as a primitive locking mechanism.
  - Server maintains a last-modified sequence number (mod-sequence) for a message.
  - Client observes the current mod-sequence and tells the server:
    - "Add the `\E3Encrypting` flag only if the mod-sequence number is unchanged."
    - If it succeeds, the client knows that it has a lock on the message.
    - If it fails, that means someone else already got the lock.

Search

- Local search
  - ✓ Most modern mail clients index mail contents locally.
  - ✓ Fully compatible with E3.

- Remote server search
  - ✗ Not possible with standard IMAP servers and encrypted email.
  - ✗ Often slow due to network latency.
  - ✗ Most IMAP servers use naive string matching per the IMAP standard.

The advantages of local search.

- E3 avoids server-side modifications.
  - We want incremental deployment and mail service independence.

  - But if mail services want, they can implement encrypted search schemes (e.g. [Aviv et al, 2007]).
    - These search schemes don’t require the private key.
    - But they do require somewhat complicated public key management.

- Mail clients already prefer local search.

Re-encrypting emails after a successful verification.

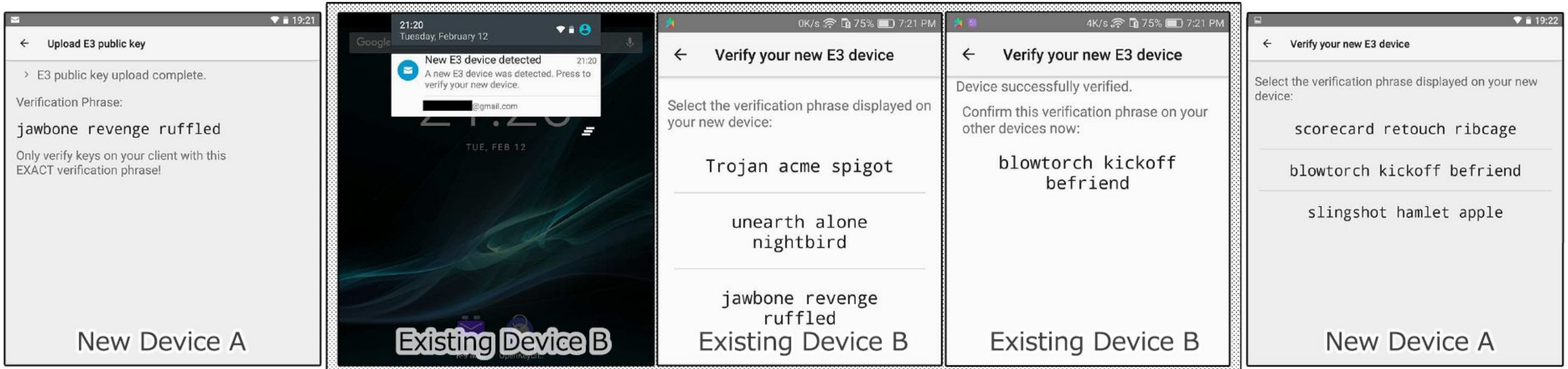
- Once a new public key is added to the E3 ecosystem, all of a user’s emails need to be re-encrypted to include that public key.

- Adding a new mail client or device is uncommon.

E3 Implementations

- K-9 Mail (Android)
  - E3 using the PGP encrypted format together with OpenKeychain.
  - E3 using the S/MIME EnvelopedData format.
- Python daemon (Windows/Linux/macOS/…)
  - Performs encrypt on receipt and basic PDK functionality.
  - Command line client proof of concept --- no user interface.
- Google Chrome extension
  - Proof of concept to show that it is possible to read E3 emails on web browser clients.

Two-way verification as it appears on Android.



① New device A completes E3's setup and displays a verification phrase.

② User's existing E3 device B detects new device A.

③ User matches the verification phrase on his existing device B.

④ Existing device B now displays a verification phrase.

⑤ The second phrase is verified on new device A to complete the process.

Columbia | Engineering
The Fu Foundation School of Engineering and Applied Science

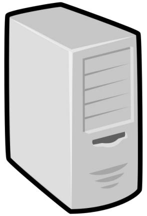
Device pairing via two-way verification. (1 / 6)

It’s like Bluetooth pairing except in both directions and using the mailbox.
Consider a user with E3 already on Device A, but he wants to add Device B.

Device A

<u>Goal</u>:
Get Device A and B to trust each other and exchange public keys.

Device B

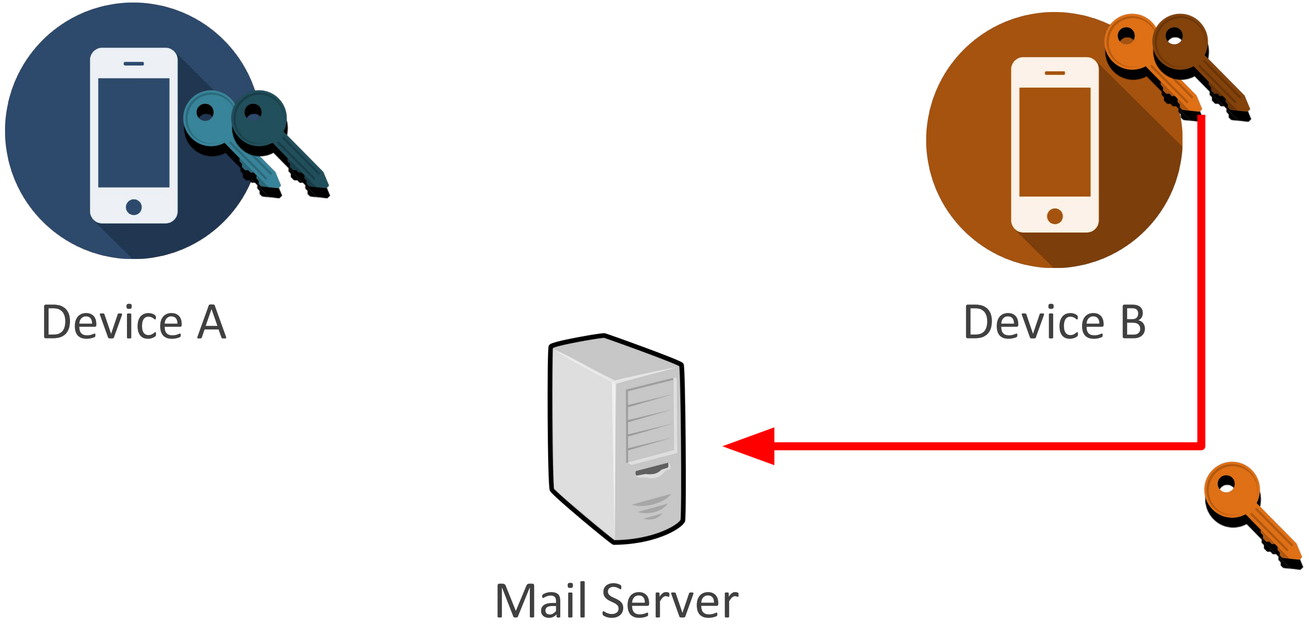


Mail Server

<u>Note</u>:
Devices communicate securely using the mailbox as the channel.

Device pairing via two-way verification. (2 / 6)

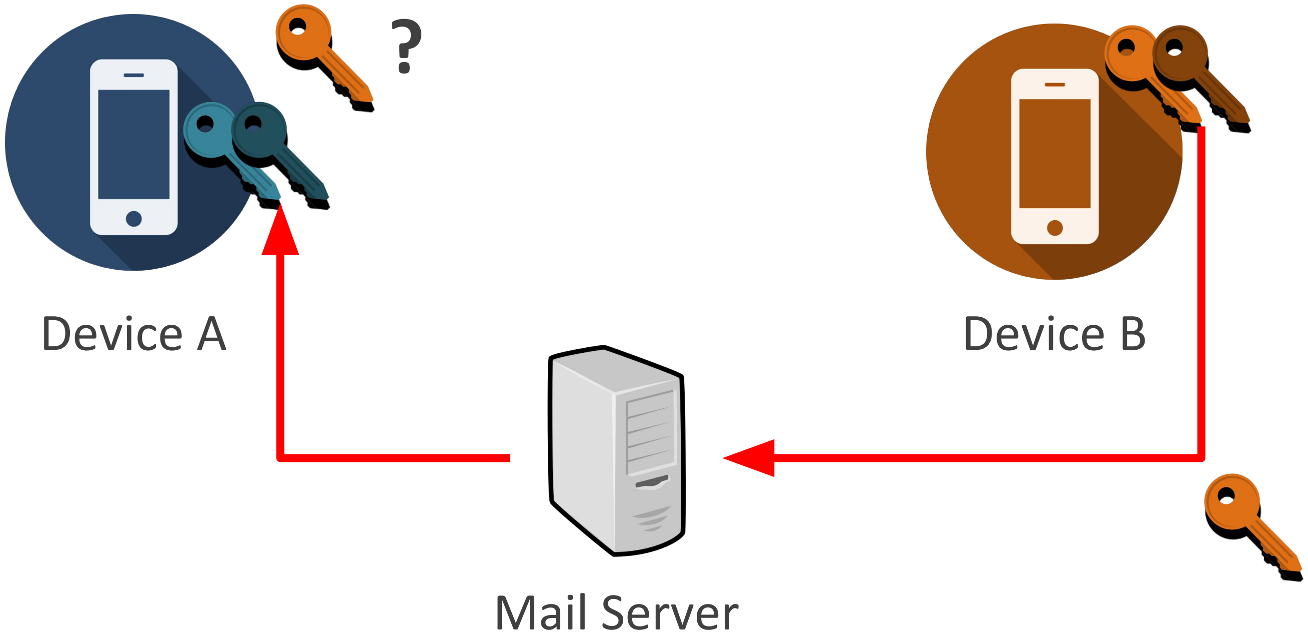
The user configures Device B with an E3 client which generates a keypair.
It **automatically** uploads its public key to the Mail Server.



Device pairing via two-way verification. (3 / 6)

Device A detects the public key that Device B uploaded.

Device A **automatically** downloads the public key **but does not accept it yet.**



Device pairing via two-way verification. (4 / 6)

Device B now displays a **three word verification phrase** to the user.

Device A then displays the same phrase plus two incorrect phrases.

Device A

Trojan	**jawbone**	**unearth**
acme	**revenge**	**alone**
spigot	**ruffled**	**nightbird**

Device B

jawbone
revenge
ruffled

Device pairing via two-way verification. (5 / 6)

Device A accepts B’s public key **iff** the user selects the correct phrase.

Device A

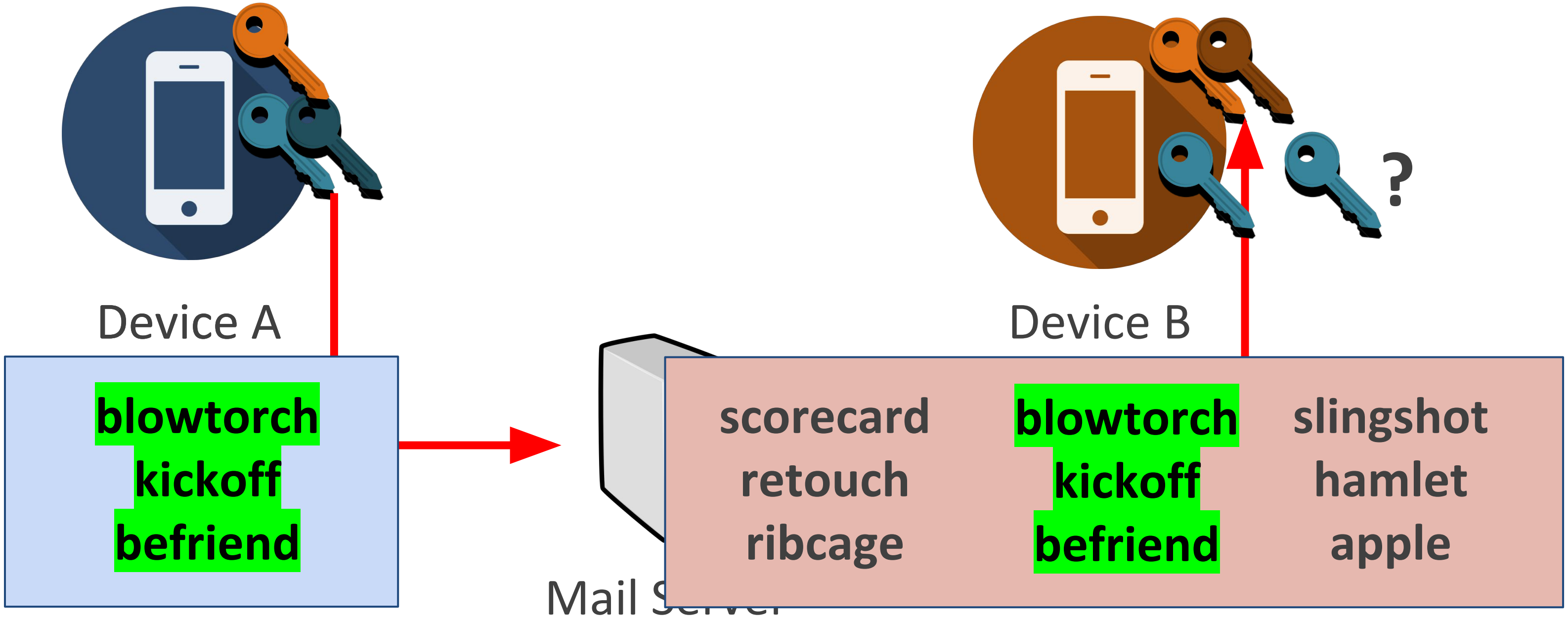
Trojan	**jawbone**	**unearth**
acme	**revenge**	**alone**
spigot	**ruffled**	**nightbird**

Device B

jawbone
revenge
ruffled

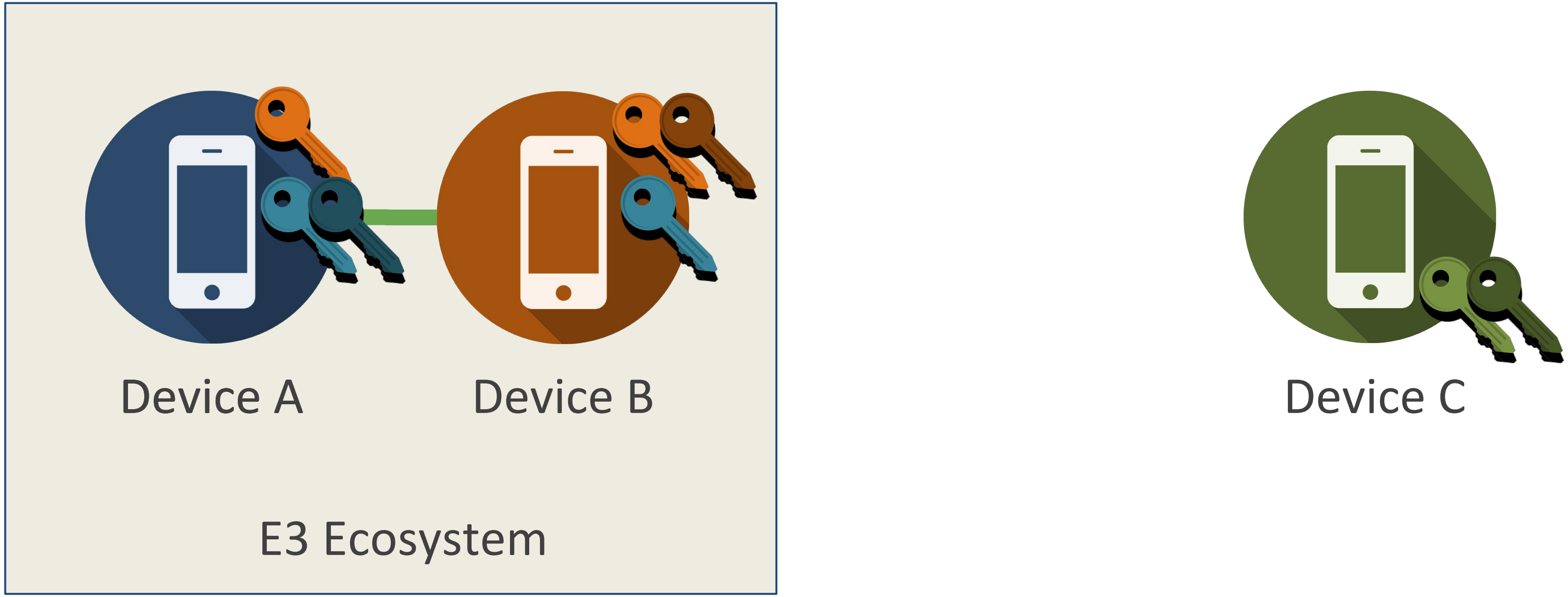
Device pairing via two-way verification. (6 / 6)

Upon successful verification, Device A then uploads its own public key.
This is the **two-way** portion --- repeat the same process the other way.



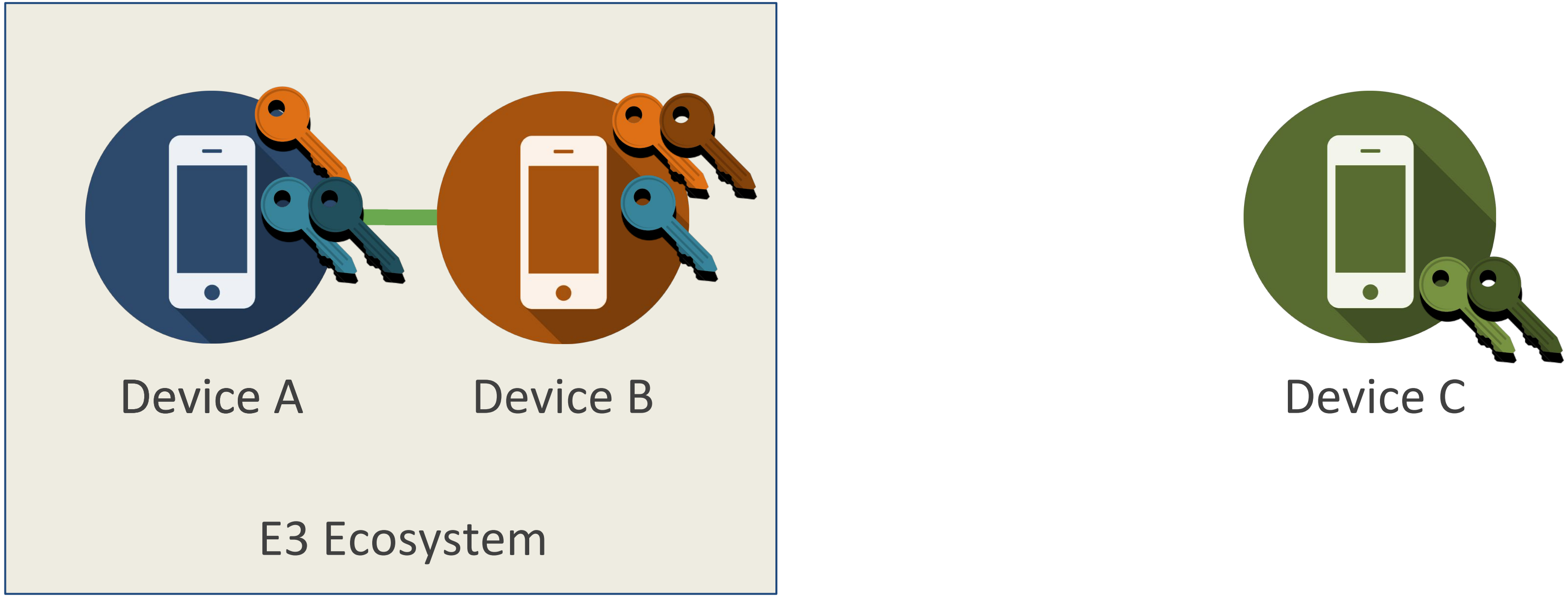
Adding an *N*th device via transitive trust. (1 / 3)

You now have an “E3 ecosystem” of devices that trust each other.
Yet, adding a new device still only takes a **single** two-way verification.



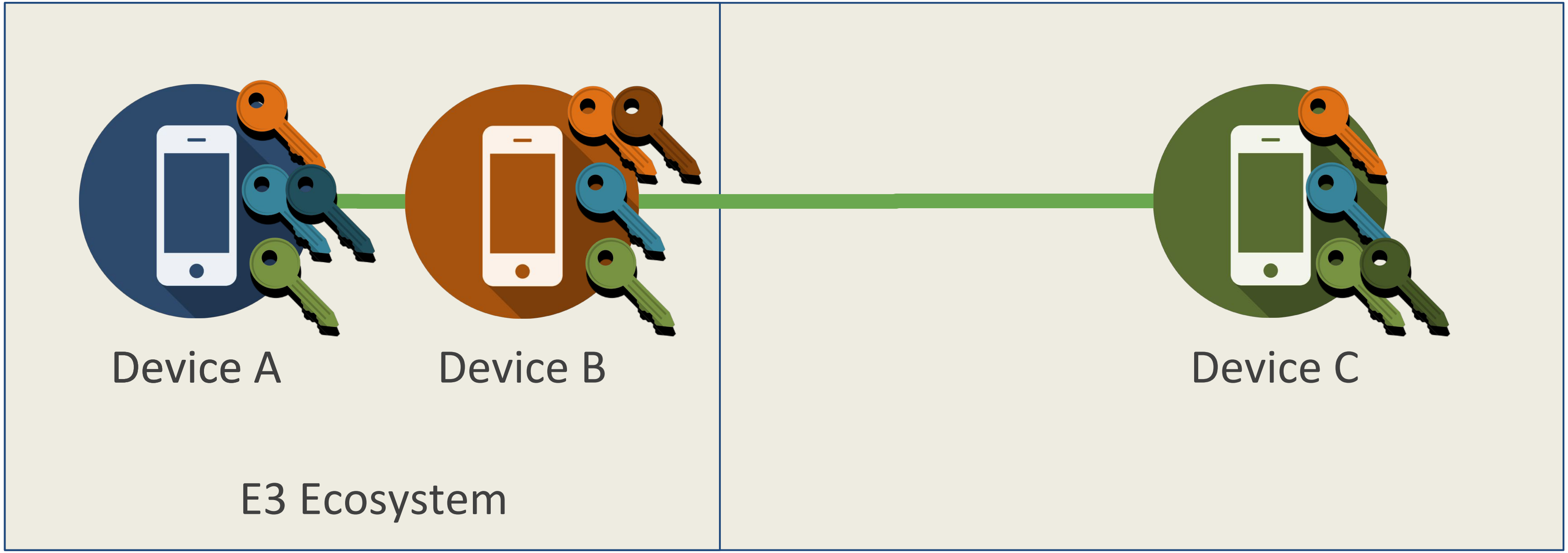
Adding an *N*th device via transitive trust. (2 / 3)

Device A and Device B already trust each other because you verified them.
So Device A will trust anything Device B trusts, and vice versa.



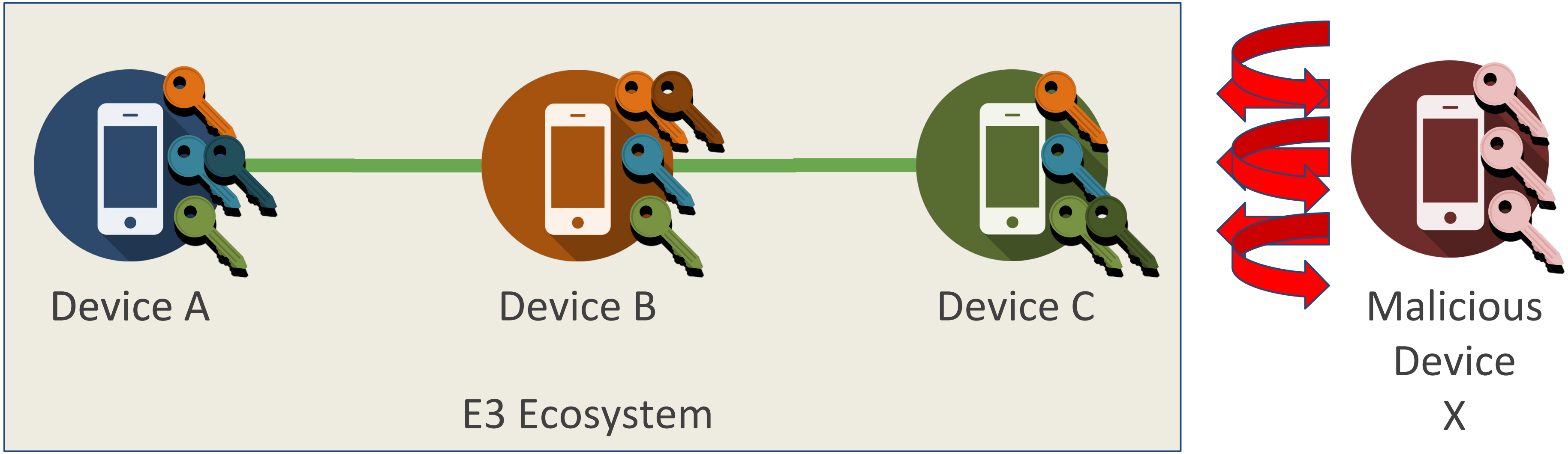
Adding an *N*th device via transitive trust. (3 / 3)

So if you do the two-way verification with Device B and Device C…
… then Device A will automatically trust Device C via the transitive property.



Devices do not accept messages signed by untrusted clients.

All messages must be **signed** so they are verified to originate from a device.
Devices A, B, and C only listen to messages signed by devices they trust.



Public keys from Device X are ignored because the user never verified it.

Ensuring that attackers can't compromise PDK.

- **Signatures**
  - A given device signs all its uploaded messages with its private key.
- **Temporal proximity**
  - All messages contain a secure and verifiable timestamp (see Google's Roughtime protocol).
  - A two-way verification request is only valid for a limited time window.
- **Rate limiting, denial of service detection, and other heuristics**
  - Two-way verification requests are rate limited.
  - Devices detect if their uploaded message is tampered with or deleted.

Some people do have only one device.

- But most people do have a family member or friend that they can trust.

- Make your family member join your E3 ecosystem (aka a key backup):
  1. Configure E3 on your family member’s device.
  2. Synchronize it with your existing devices.
  3. Optional: Remove your email account credentials from the device.
  4. Now your family member’s device has a key that can be used to access your email if your lose your device.

Two-way verification compared to other approaches.

- Other key management solutions have shortcomings.
  - They rely on specific technology unavailable on all devices.
    - Bluetooth, NFC, QR codes (requires camera), … [Schurmann, 2017].
  - Or they rely on a third-party service which we want to avoid.
    - Third-party identity-based encryption server [Ruoti 2013, 2016].
    - Social media sites and a third-party service [Lerner, 2017].

Users attempted to achieve a simple goal.

- **Goal:**
  - Finish setting up each mail client, and send/read email on all 3 devices.
    - 20 minute time limit for each of:
      - **K-9 Mail** (unmodified, regular email)
      - **E3**
      - **PGP**
  - E3 and PGP used:
    - K-9 Mail
    - OpenKeychain [Schurmann et al, 2017]

K-9 Mail **OpenKeychain**

How we gathered results.

- Time spent with each email solution.
- Whether user succeeded in completing all assigned tasks or not.
- An industry standard survey (System Usability Score).
- Our own customized survey to compare the three solutions.
- Any remarks by users during or after the study.