How to really mess with a surveillance state

BREAKDOWN:

ANALYSIS AND REBIRTH

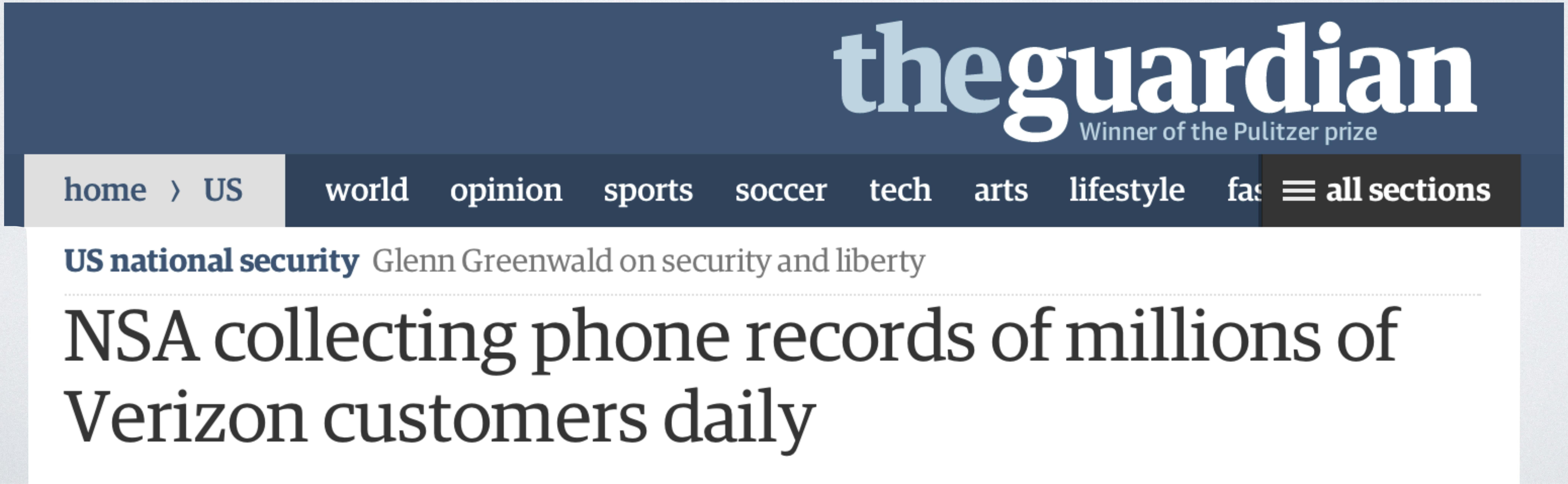
1. Politics: *how is shit hitting the fan?*
2. Psychology: *why does shit hit fans?*
3. Design: *how can we engineer shit and fans to not come in contact?*

THE HIGHLIGHT ZONE

June 6, 2013 *(526 days ago)*

June 6, 2013 *(526 days ago)*

Edward Snowden's leaked NSA docs lead to the first exposé of the mass surveillance it details. Many more to come.



theguardian
Winner of the Pulitzer prize

home › US world opinion sports soccer tech arts lifestyle fas ≡ all sections

US national security Glenn Greenwald on security and liberty

NSA collecting phone records of millions of Verizon customers daily

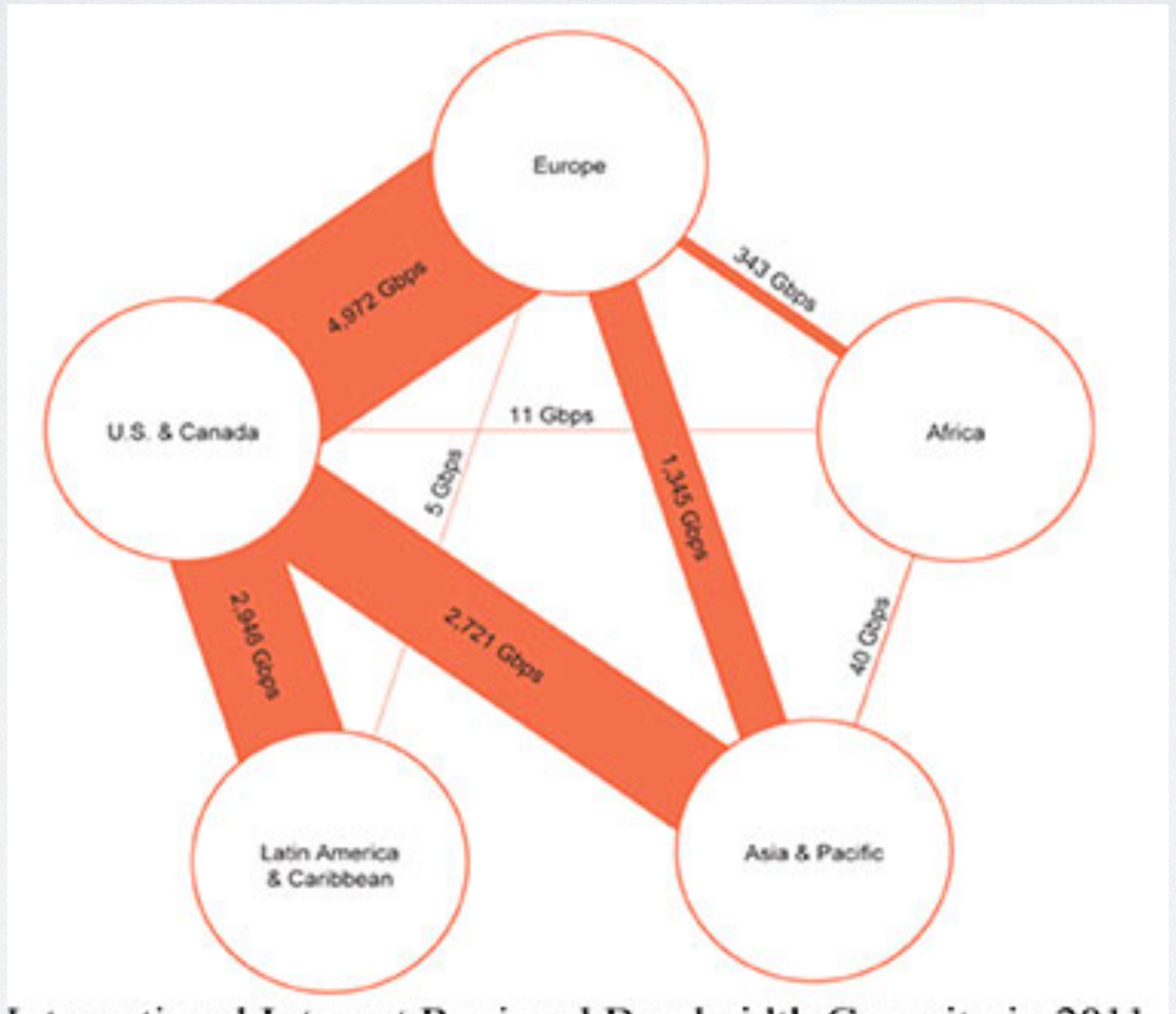
Today.

- ~2,500 pages released out of "hundreds of thousands"[1] of documents
- private analysis on the larger unreleased corpus reveals systemic inconsistencies with public government claims.

[1] http://www.3news.co.nz/tvshows/thenation/interview-glenn-greenwald-2014091311?ref=video

useful information

They have the advantage of being able to observe a large portion of international traffic through domestic fibers and partners.

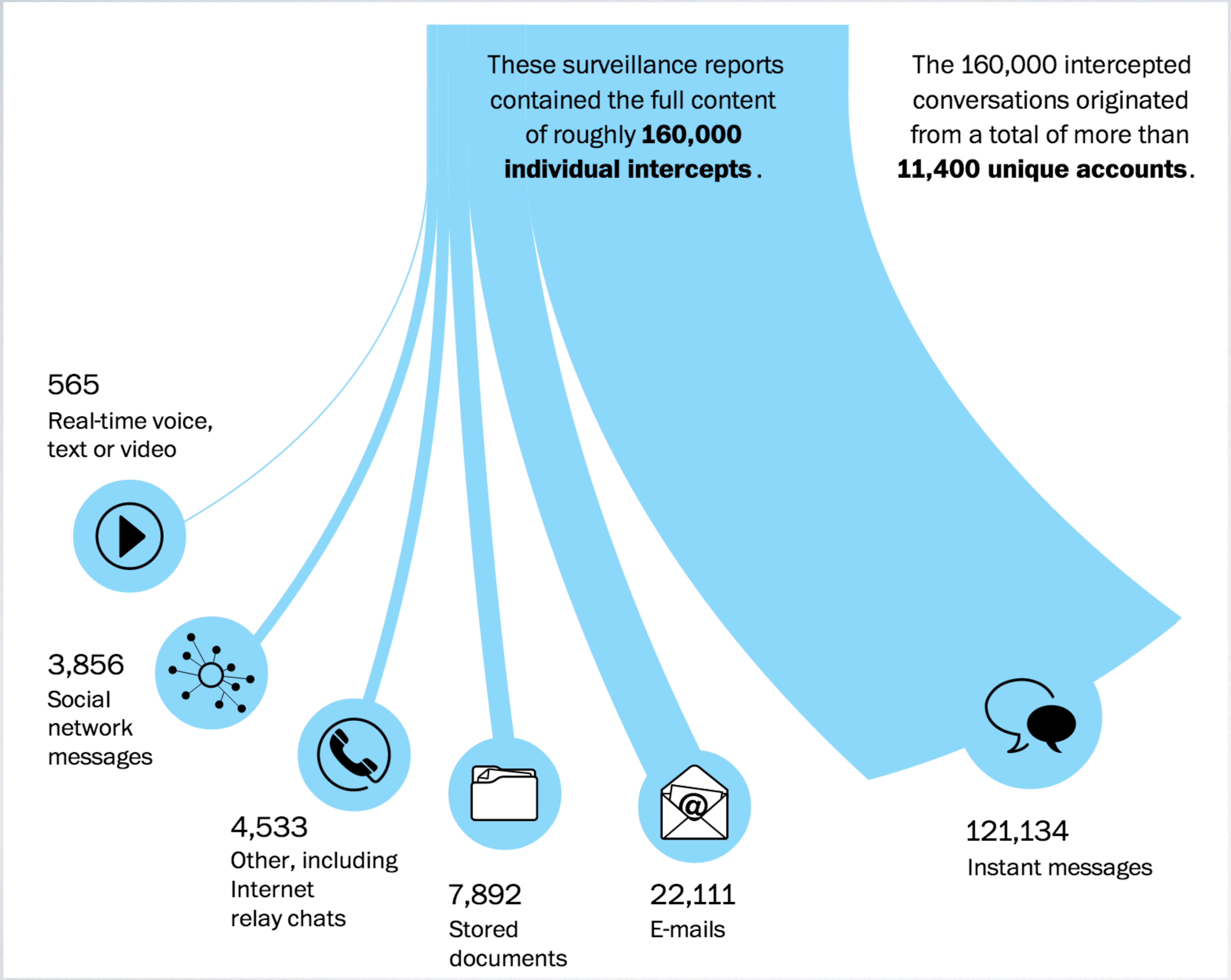


International Internet Regional Bandwidth Capacity in 2011

Source: Telegeography Research

TOP SECRET//SI//ORCON//NOFORN

of the traffic they collect, vast contents of intercepted traffic is IM



These surveillance reports contained the full content of roughly **160,000 individual intercepts**.

The 160,000 intercepted conversations originated from a total of more than **11,400 unique accounts**.

565
Real-time voice, text or video

3,856
Social network messages

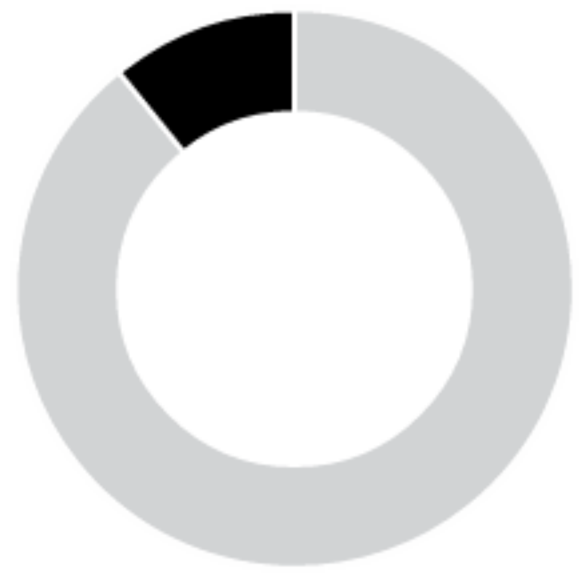
4,533
Other, including Internet relay chats

7,892
Stored documents

22,111
E-mails

121,134
Instant messages

Eleven percent of the accounts were NSA targets.



The remaining **89 percent** of the accounts were bystanders, or non-targets.*

* This figure excludes “mini mized” U.S. persons (See below)

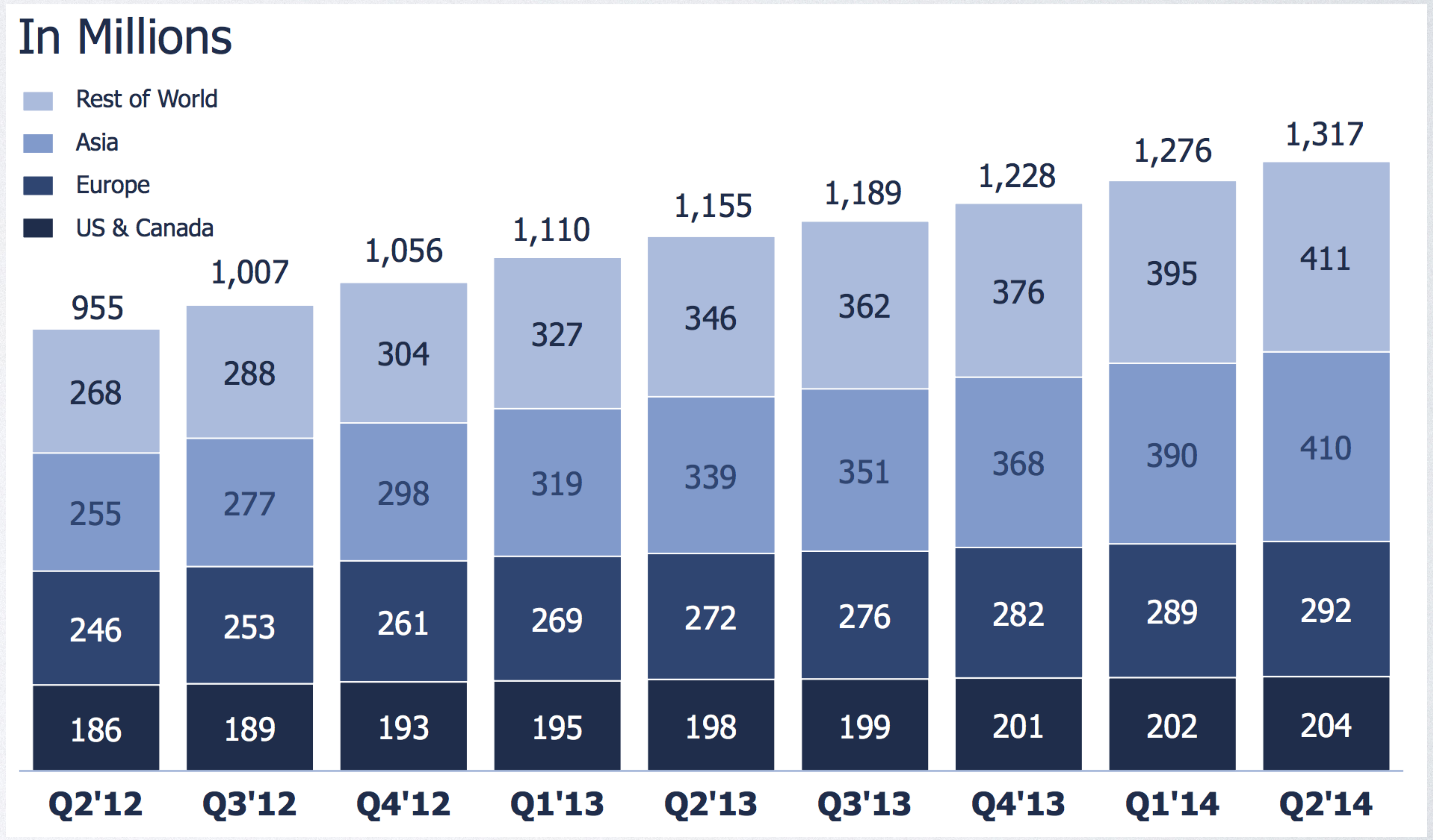
source: http://apps.washingtonpost.com/g/page/world/communication-breakdown/1153/

They ***do*** have connections and resources that give them access to the great modern mother lodes of the internet.

They ***do*** have connections and resources that give them access to the great modern mother lodes of the internet.

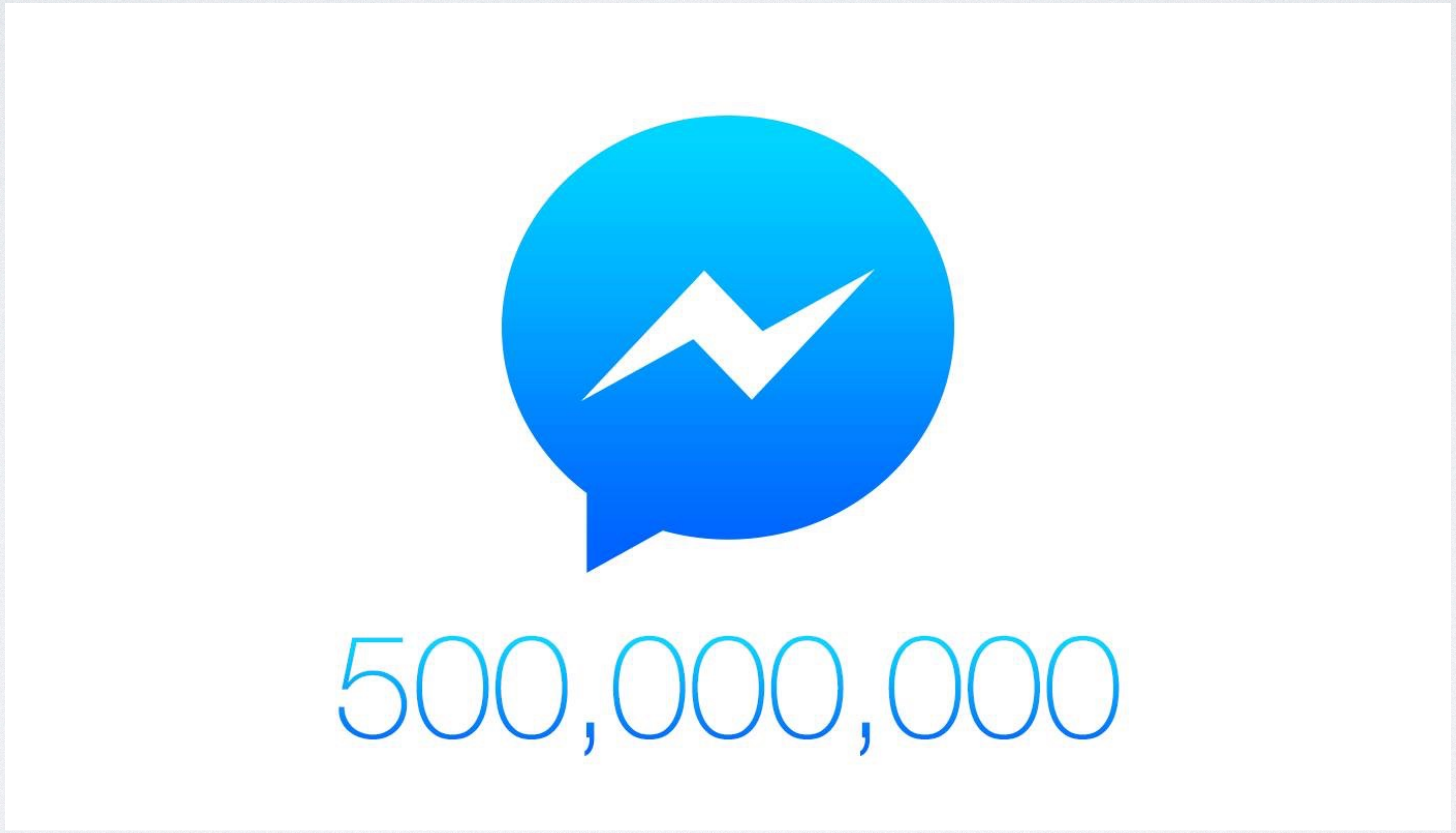
1.3 billion active users

~18% of world population
~45% of the internet population



	Q2'12	Q3'12	Q4'12	Q1'13	Q2'13	Q3'13	Q4'13	Q1'14	Q2'14
Rest of World	268	288	304	327	346	362	376	395	411
Asia	255	277	298	319	339	351	368	390	410
Europe	246	253	261	269	272	276	282	289	292
US & Canada	186	189	193	195	198	199	201	202	204
	955	1,007	1,056	1,110	1,155	1,189	1,228	1,276	1,317

They ***do*** have connections and resources that give them access to the great modern mother lodes of the internet.

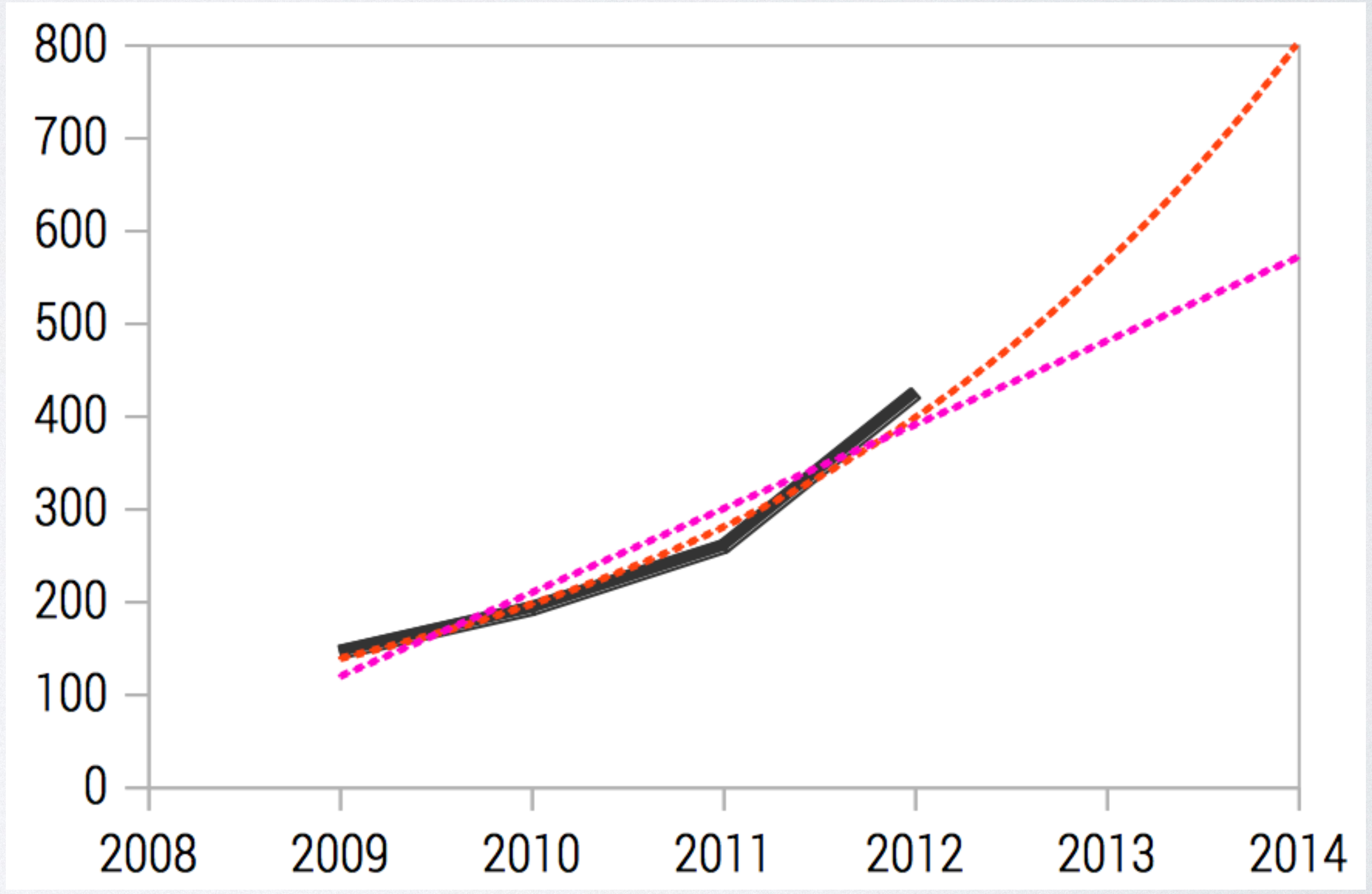


(WhatsApp is at 600 million, also Facebook-owned)

They ***do*** have connections and resources that give them access to the great modern mother lodes of the internet.

~550-800 million active users

~10% of world population
~30% of internet population



They ***do*** enjoy looking at naked pictures of you.

well, not as classified: http://blogs.wsj.com/washwire/2013/08/23/nsa-officers-sometimes-spy-on-love-interests/

They ***do*** enjoy looking at naked pictures of you.

well, not as classified: http://blogs.wsj.com/washwire/2013/08/23/nsa-officers-sometimes-spy-on-love-interests/

… so much more. but, we’ve been reminded of something:

intelligence agencies are not magical unicorns

they are humans in a big bureaucratic labyrinth

- Made Stuxnet and managed to collect data in amounts that surprised even the moderately paranoid person
- Still, pretty bad opsec. They still even use polygraph tests on their employees.
- “modernization plans that are constantly put off and an ever-increasing flood of information that the NSA is forever trying to get under control"

They ***don't*** seem to have broken the mathematical assumptions of modern cryptography.

- DLP primitives still seem hard, e.g. RSA, Curve25519
- Modern recommended standards still seem hard, e.g. AES, SHA-2.
- Evidence shows they tend toward lower-hanging fruit than this

They ***do*** seem to have attacked the standardization procedures in getting good crypto ubiquitously implemented

- Dual_EC_DRBG
- Opposition to constant-time implementations and higher key strength
- Poisoning the well with doubt of non-standardized crypto
- Slow the phase-out of broken crypto

THE NSA LOVES HUMAN ERROR

which leads us to…

PSYCHOLOGY

our minds are the NSA's favorite broken security system

THE LOW-HANGING FRUIT IS **US**

- Through our own free will, we've opted for convenience.
  (remember how nearly half of the internet will be active on Facebook this month)
- **Convenience** and **security** are rarely referred to in the same breath.

THE LOW-HANGING FRUIT IS **US**

Assertion:
Most people do care about privacy, but it's only one factor in a larger equation when deciding to participate in activities.

Social pressures typically prove to be stronger.

THE LOW-HANGING FRUIT IS **US**

Companies dump non-truths on customers in order to make them believe their privacy is preserved more than it is.

"we ***can't*** look at your messages"

vs.

"we ***won't*** look at your messages"

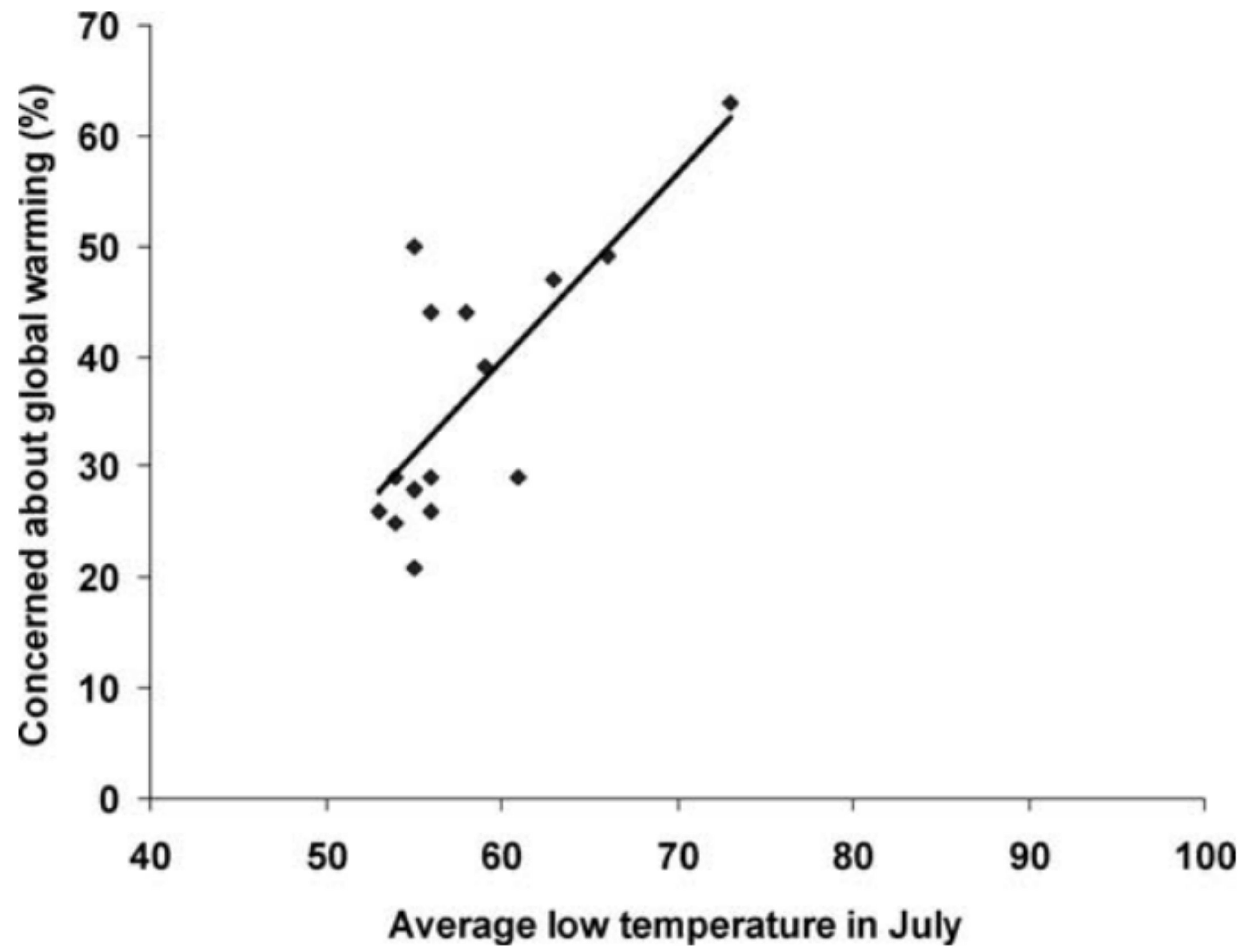
THE LOW-HANGING FRUIT IS **US**

Surveillance is very intentionally abstract and intangible.

Why? Because they know humans suck at long-term risk.

“Finite pool of worry”

GLOBAL WARMING: THE PSYCHOLOGY OF LONG TERM RISK



CHOICE BLINDNESS

We do say we care about privacy, and we even put massive effort into "crypto for the people" during the crypto wars.

However, the mass populous doesn't use them.

Somebody wanna scrape a PGP keyserver and find out the number of active keys?

WHY?

AKA: let's keep kicking PGP

- Training is required to use it without shooting yourself in the foot. Easy to do really stupid things.

- Socially difficult due to required lifestyle change of both you and the people you talk to

- Lots of "you're on your own" with key management to prevent inevitable compromise.

WHY?

AKA: let’s keep kicking PGP

- *friction*
- *friction*
- *friction*

These are all high cognitive load.

PGP

- Works in asynchronous environments
- Lacks forward/future secrecy
- Lacks deniability
- Complicated setup and usage

PGP

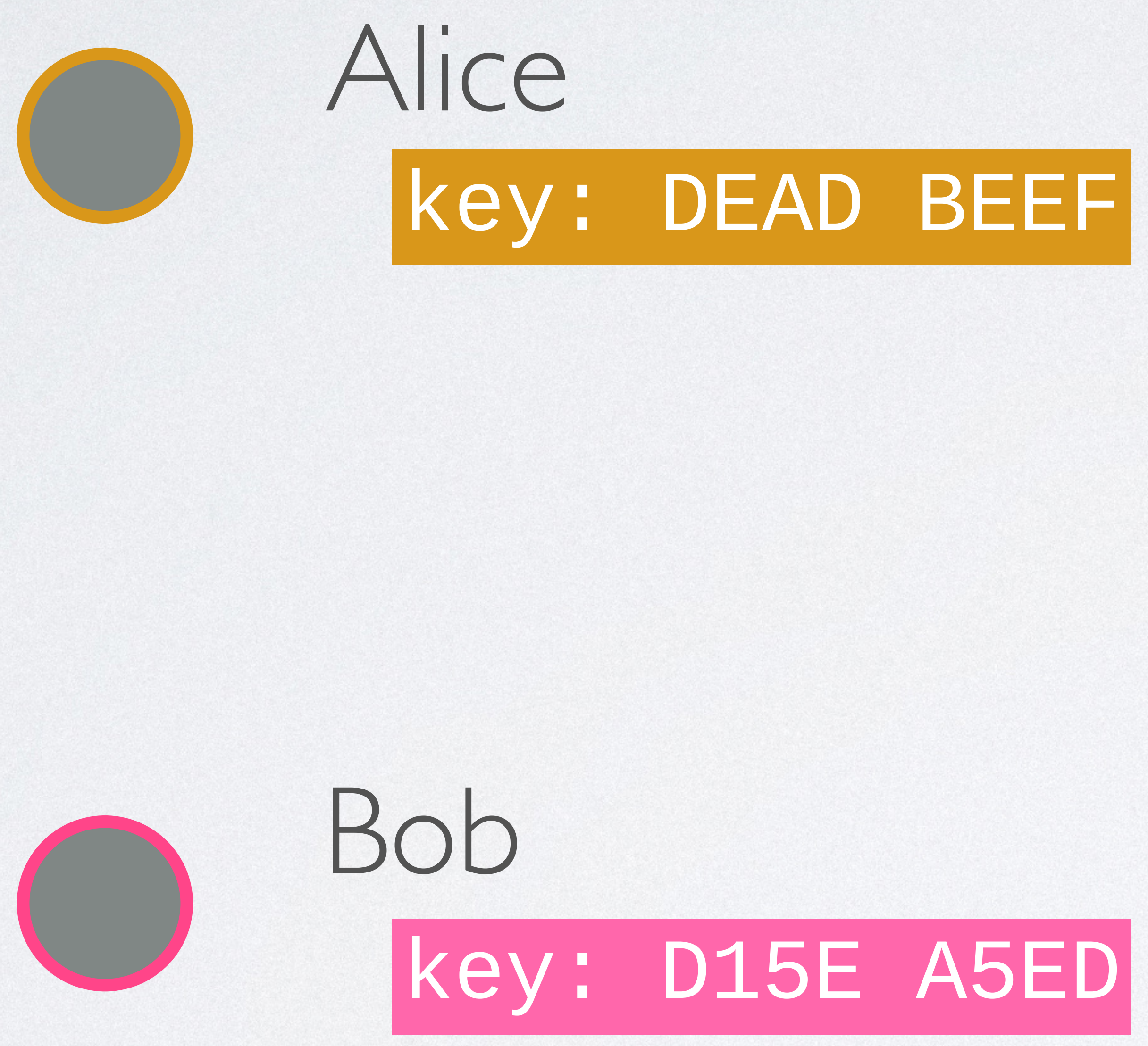
Alice

`key: DEAD BEEF`

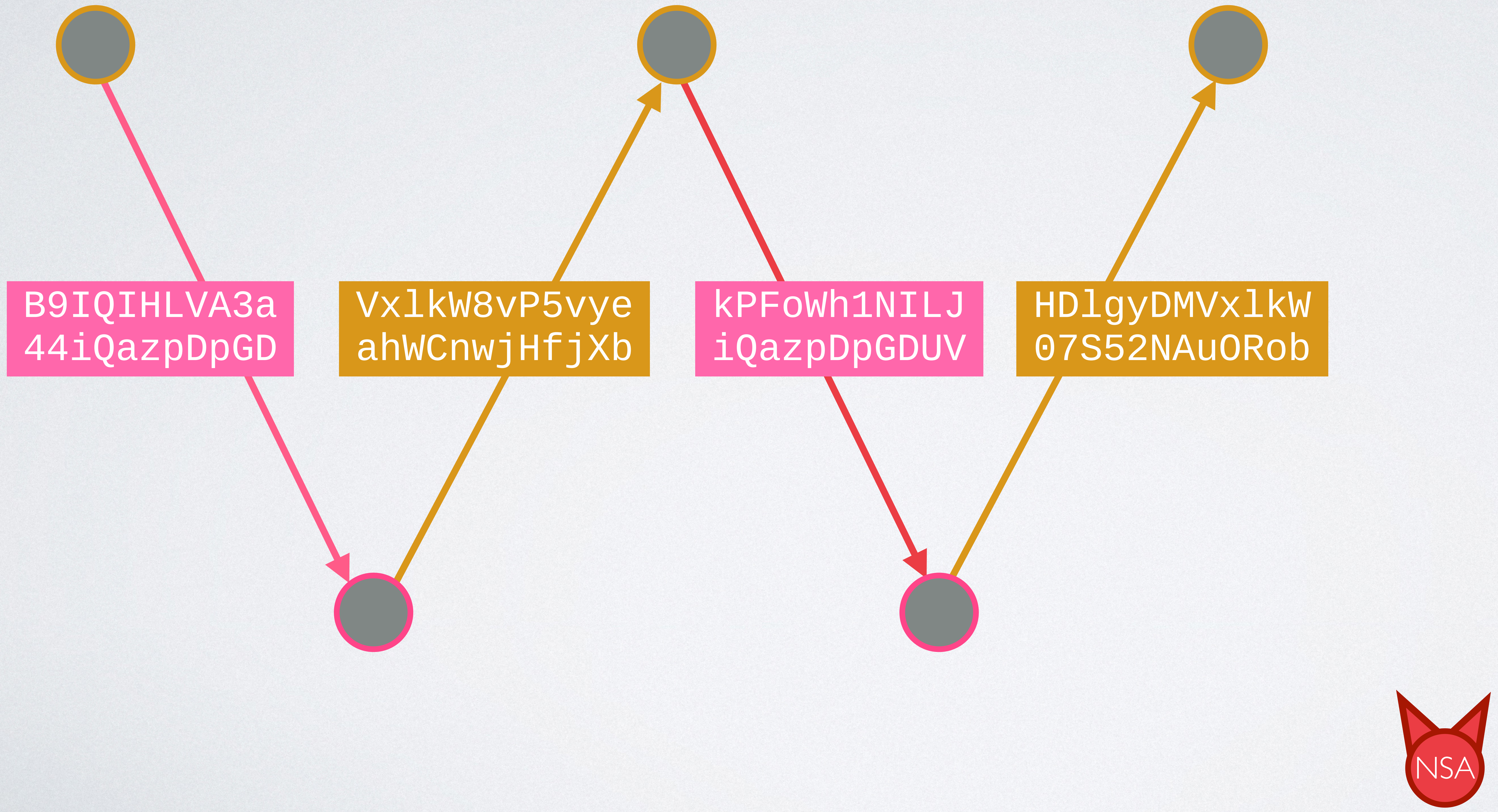
Bob

`key: D15E A5ED`

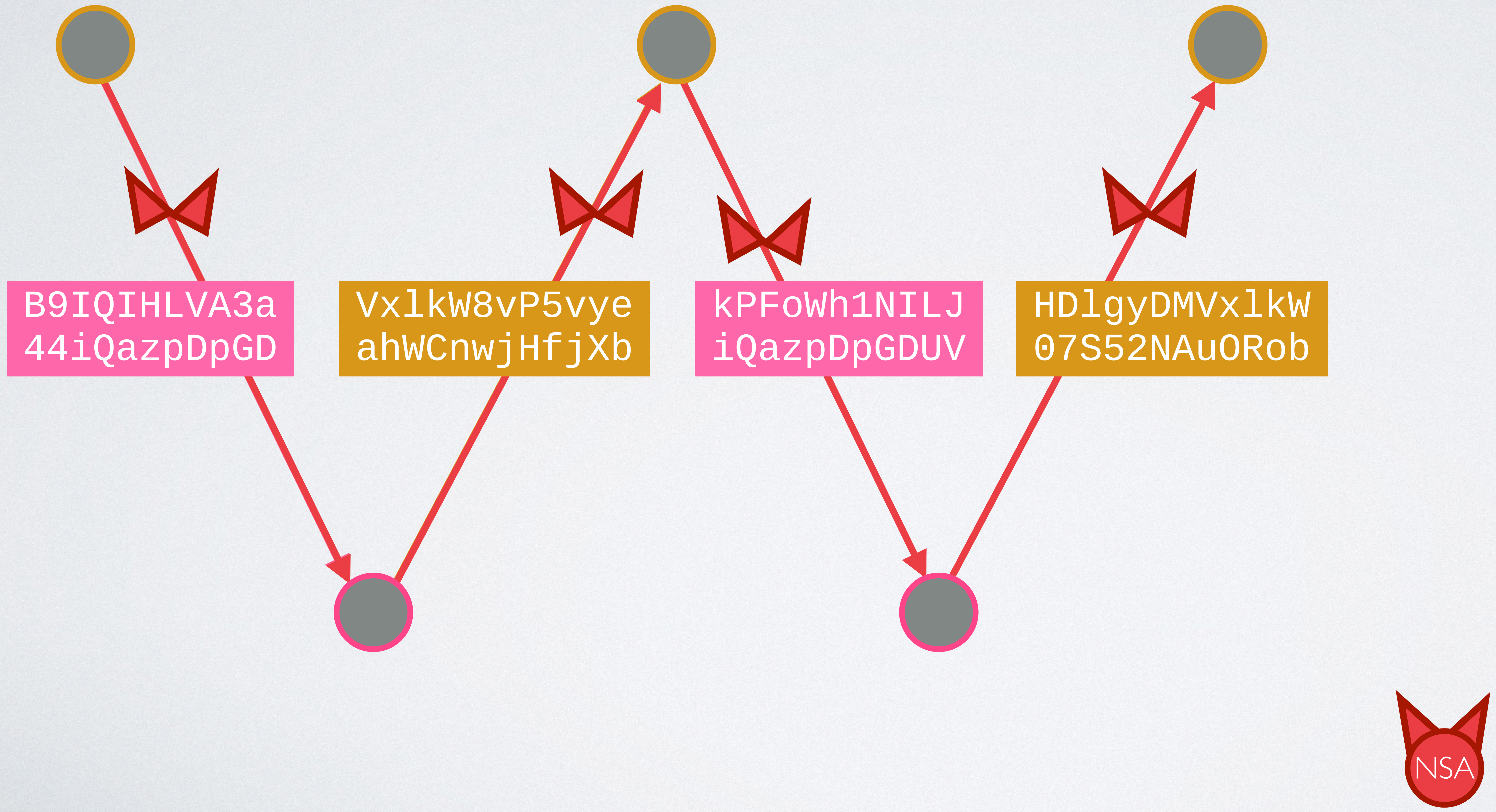
PGP



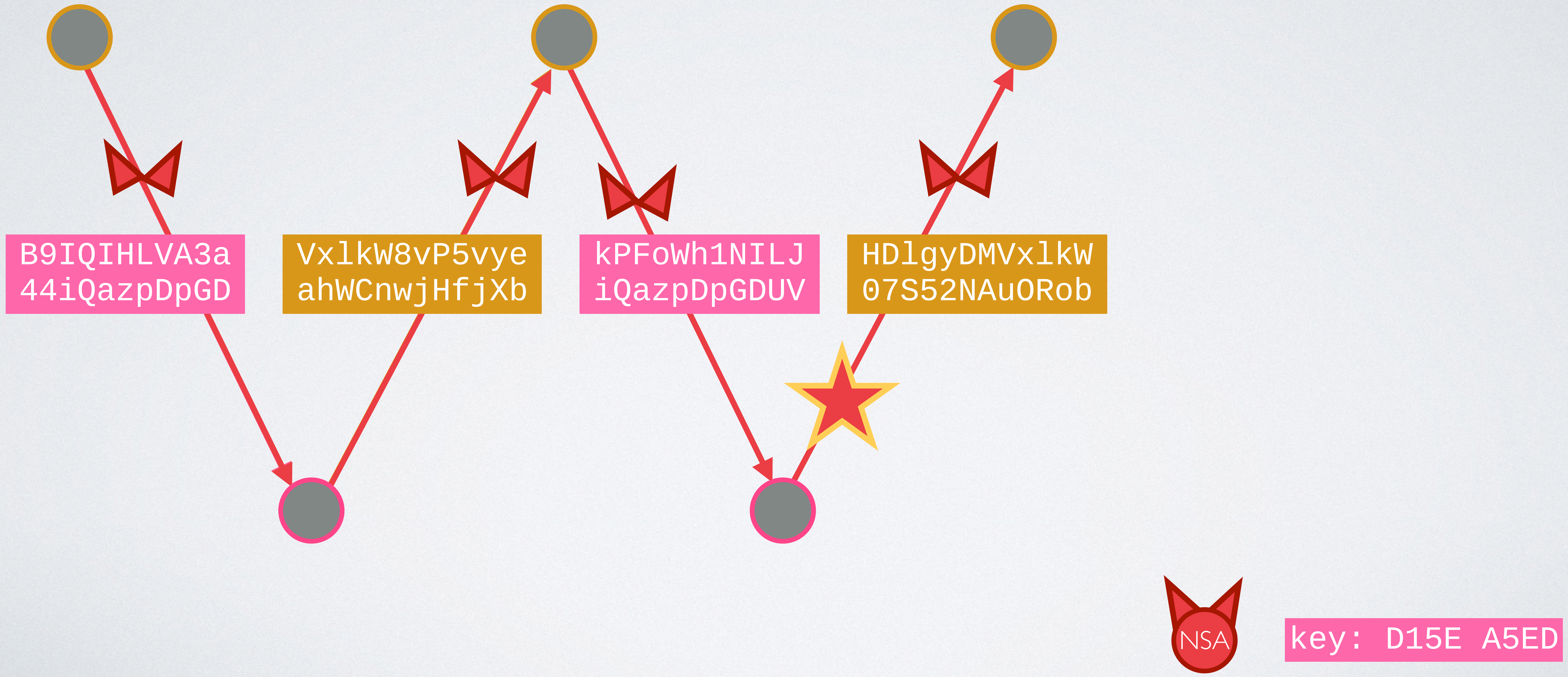
PGP


B9IQIHLVA3a
44iQazpDpGD
VxlkW8vP5vye
ahWCnwjHfjXb
kPFoWh1NILJ
iQazpDpGDUV
HDlgyDMVxlkW
07S52NAuORob
NSA

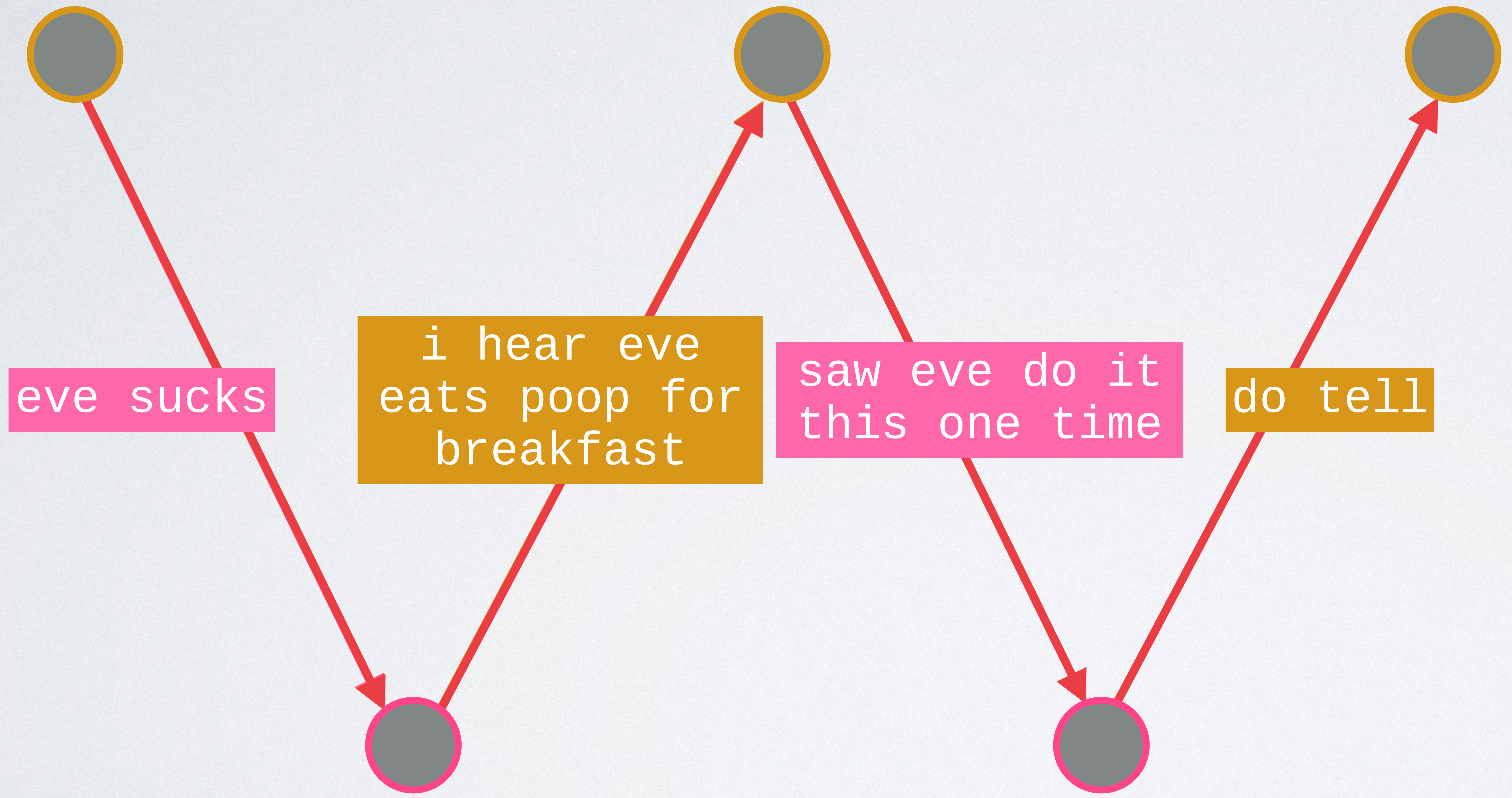
PGP


B9IQIHLVA3a
44iQazpDpGD
VxlkW8vP5vye
ahWCnwjHfjXb
kPFoWh1NILJ
iQazpDpGDUV
HDlgyDMVxlkW
07S52NAuORob
NSA

PGP


B9IQIHLVA3a
44iQazpDpGD
VxlkW8vP5vye
ahWCnwjHfjXb
kPFoWh1NILJ
iQazpDpGDUV
HDlgyDMVxlkW
07S52NAuORob
NSA
key: D15E A5ED

PGP


eve sucks
i hear eve eats poop for breakfast
saw eve do it this one time
do tell

key: D15E A5ED

OTR

- Forward secrecy via a ratcheting ephemeral key exchange
- Fewer ways to shoot yourself in the foot
- Synchronous
- Still unsupported on most used chat clients
- Requires security tribal knowledge (keys, fingerprints, …)

WHAT DO WE *NEED*?

- Limited damage from key compromise
- Sane defaults
- Opportunistic, transparent encryption
- Resilience even in the most hairy network environment: *mobile*.

WHAT DO WE *NEED*?

We need the lock icon to become a thing of history that we describe nostalgically to our grandchildren.

"One click encryption is one click too many."

–Bruce Schneier like 2 minutes ago

AXOLOTL

AXOLOTL

- ✓ Async-tolerance (missing/lost/out of order messages OK)
  unlike OTR
- ✓ Both forward and future secrecy
  unlike SCIMP, PGP, and stronger guarantee than OTR
- ✓ Deniability
  unlike SCIMP, PGP

How, you ask?

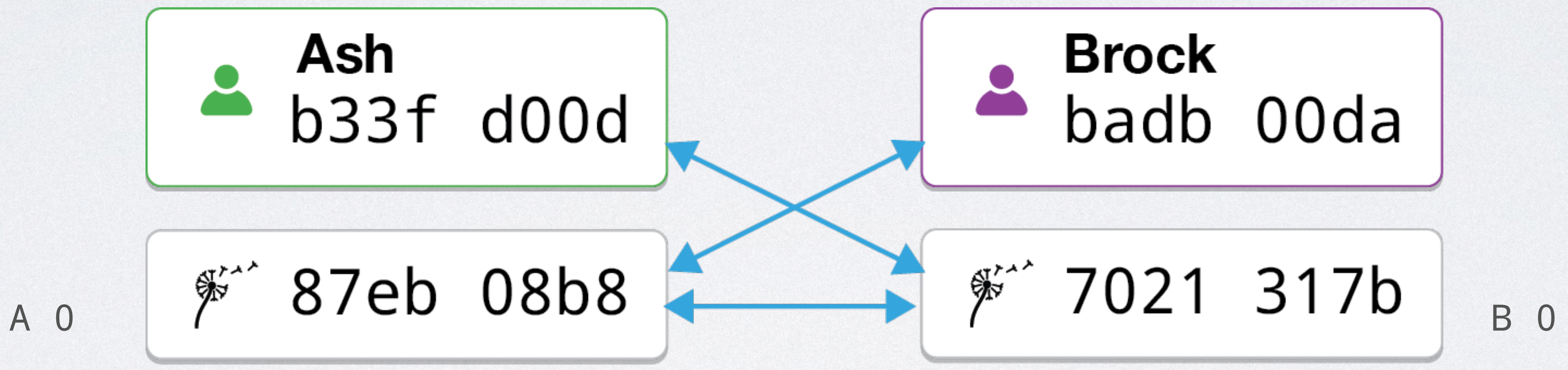
AXOLOTL: THE AXPLANATION

Familiar session-based cryptographic protocol setup:

1. Establish a shared master “root” secret via Triple DH
2. Generate associated chain and message keys
3. Get schemin’

AXOLOTL: THE AXPLANATION

session establishment with 3DH

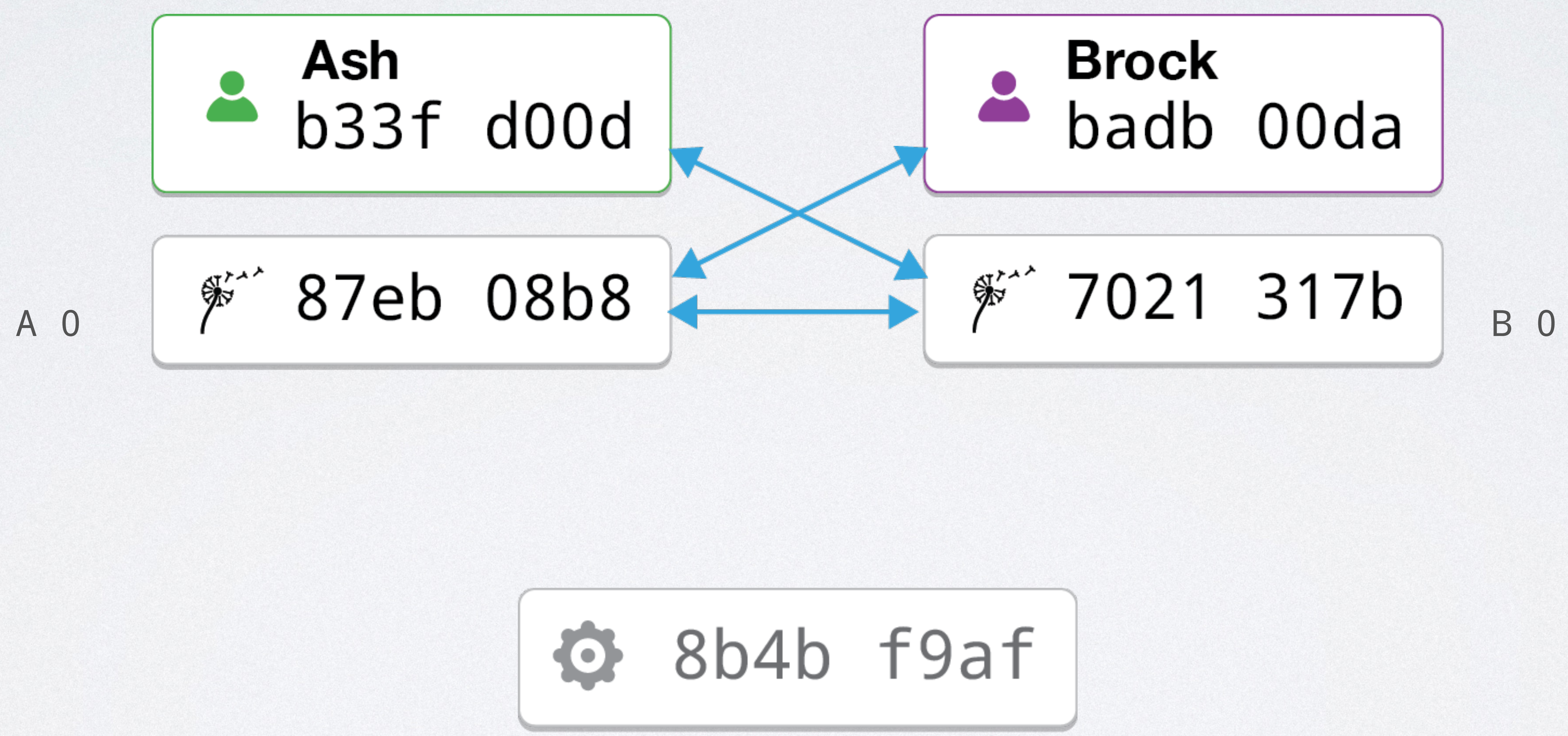


h a s h (D H (A, B 0) || D H (B , A 0) || D H (A 0 , B 0))

🟦 = Diffie-Hellman

AXOLOTL: THE AXPLANATION

session establishment with 3DH



🟦 = Diffie-Hellman

AXOLOTL: THE AXPLANATION

getting to the first message: chain and message keys

From the most recent root key

`8b4b f9af`

AXOLOTL: THE AXPLANATION

getting to the first message: chain and message keys

From the most recent root key

⚙ 8b4b f9af

Ash generates a new ephemeral keypair

A1: 1e22 5e10

AXOLOTL: THE AXPLANATION

getting to the first message: chain and message keys

From the most recent root key ⚙ 8b4b f9af

Ash generates a new ephemeral keypair A1: 1e22 5e10

Ash calculates a new root key HASH(last_root || DH(A1, B0))

AXOLOTL: THE AXPLANATION

getting to the first message: chain and message keys

From the most recent root key ⚙ 8b4b f9af

Ash generates a new ephemeral keypair A1: 1e22 5e10

Ash calculates a new root key 2dff 90d5

AXOLOTL: THE AXPLANATION

getting to the first message: chain and message keys

From this new root key

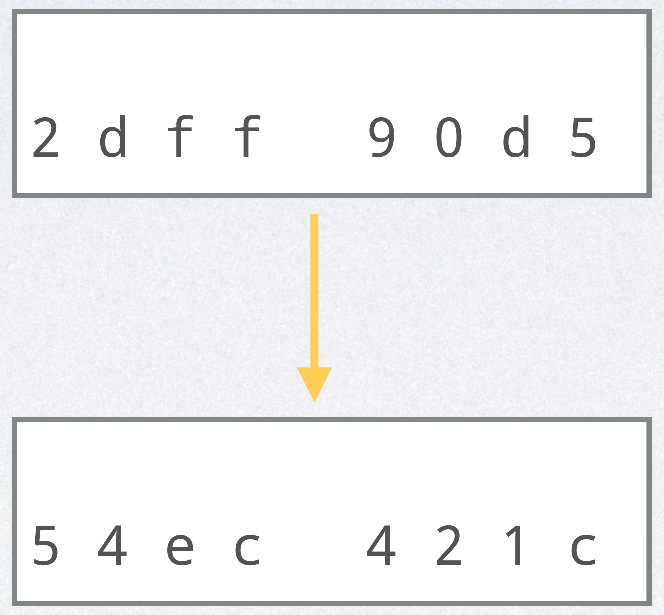
```
2dff 90d5
```

AXOLOTL: THE AXPLANATION

getting to the first message: chain and message keys

From this new root key

derive a chain key via KDF



🟨 = One-way hash

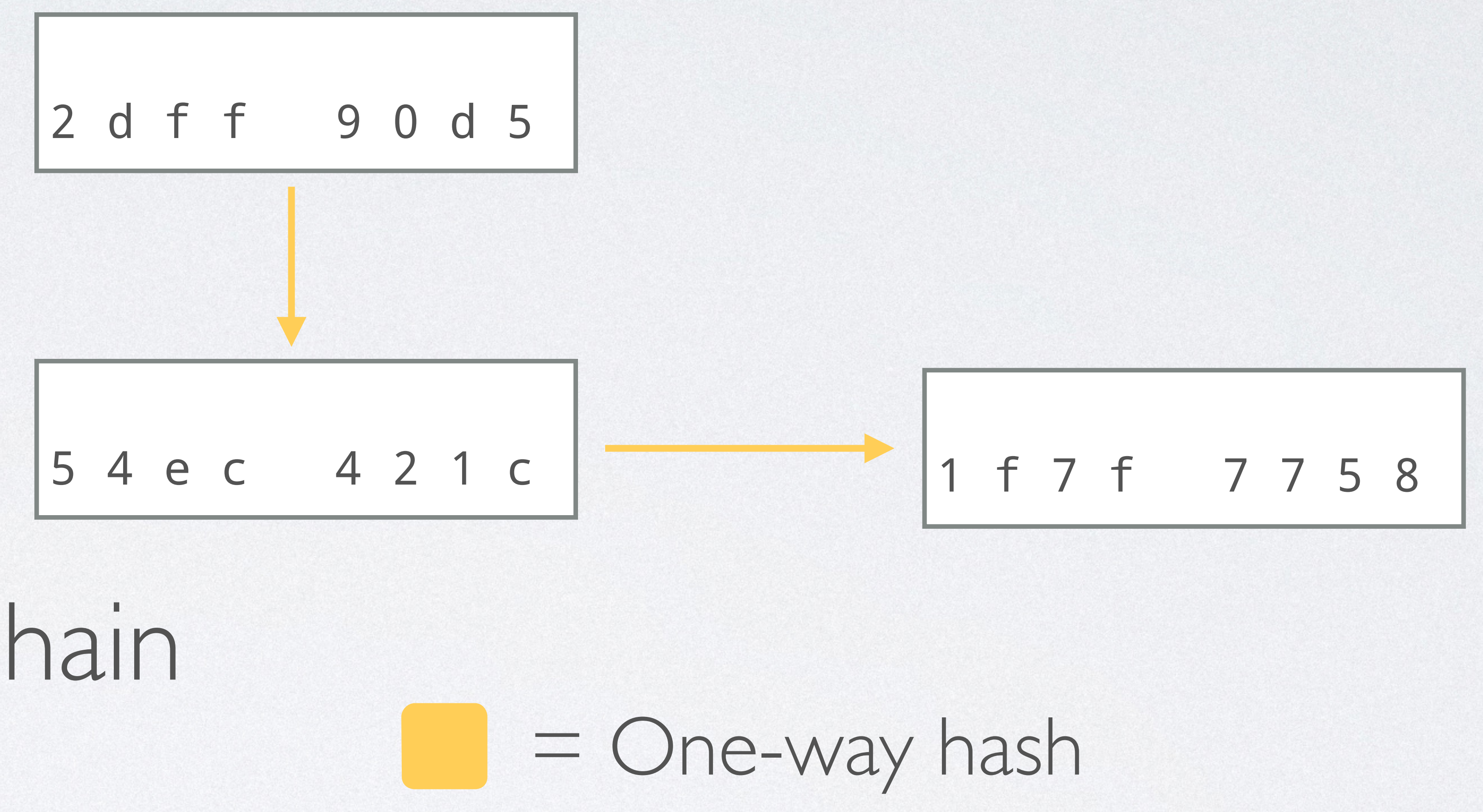
AXOLOTL: THE AXPLANATION

getting to the first message: chain and message keys

From this new root key

derive a chain key via KDF

then derive a message key from chain



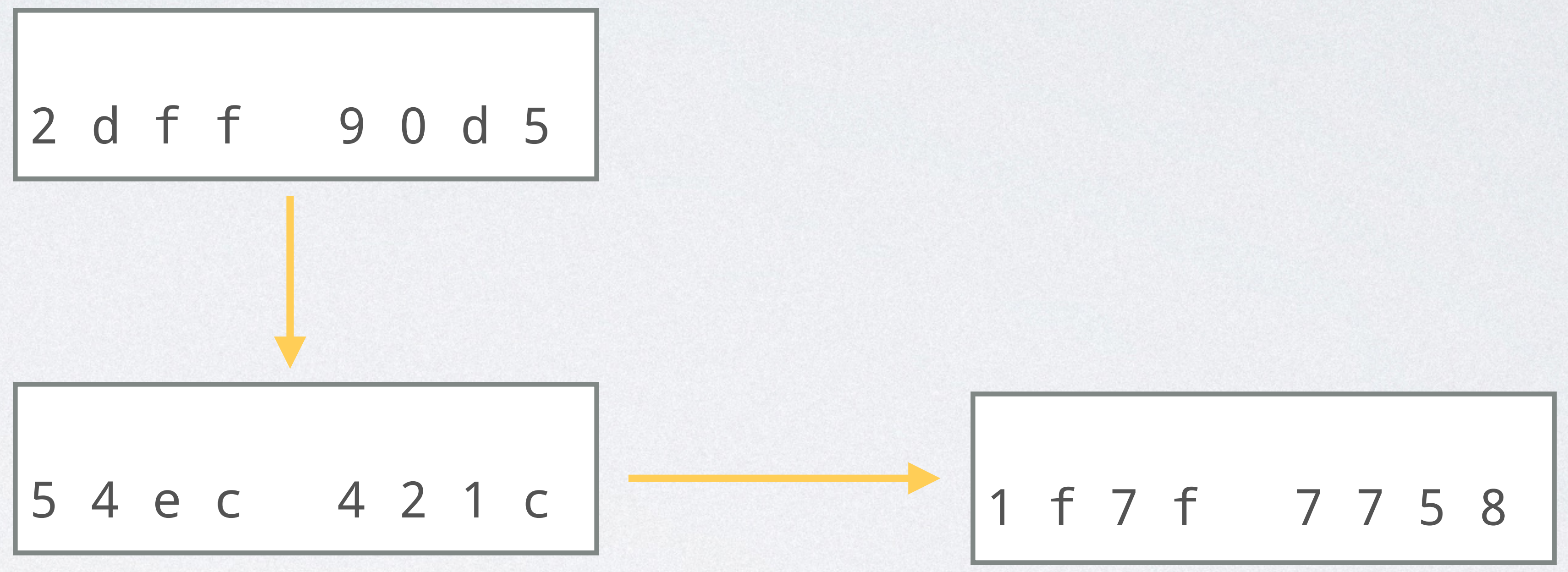
AXOLOTL: THE AXPLANATION

getting to the first message: chain and message keys

From this new root key

derive a chain key via KDF

then derive a message key from chain



🟨 = One-way hash

When Ash sends his message

```
message
```

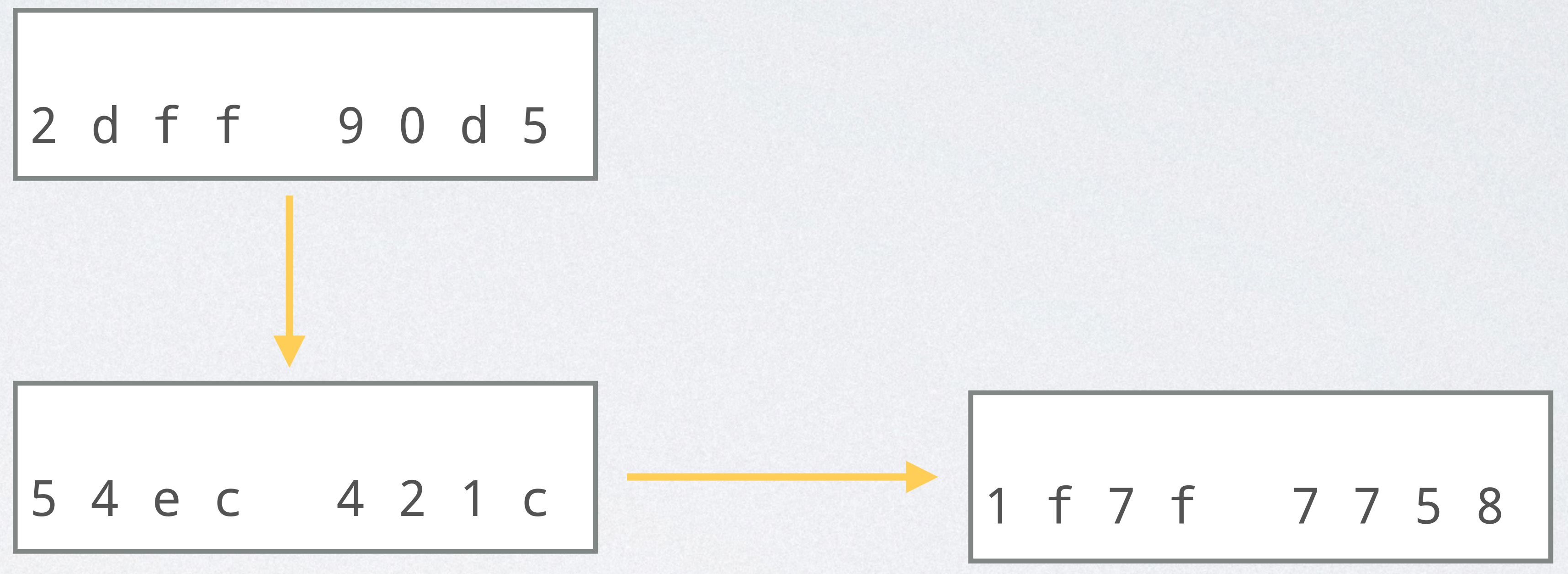
AXOLOTL: THE AXPLANATION

getting to the first message: chain and message keys

From this new root key

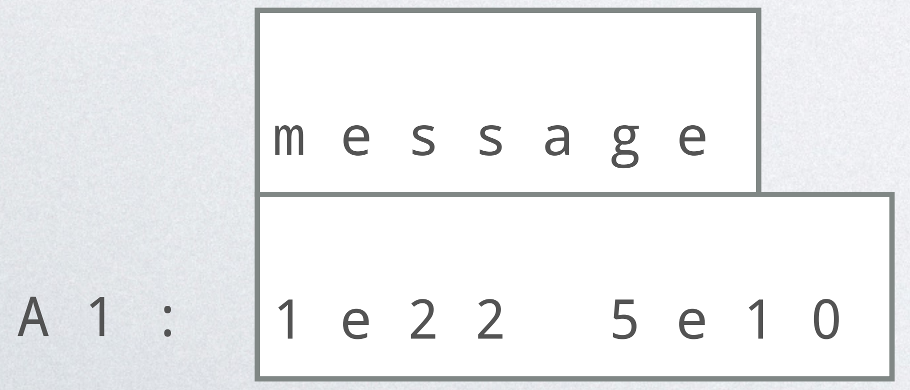
derive a chain key via KDF

then derive a message key from chain



🟨 = One-way hash

When Ash sends his message, he also sends his new ephemeral pubkey



AXOLOTL: THE AXPLANATION

getting to the first message: chain and message keys

When Brock receives Ash’s message including his new ephemeral key, he can calculate the new root key and generate a matching message key.

message
A1:
1e22 5e10

AXOLOTL: THE AXAMPLE

Ash and Brock are concerned that Misty is actually a shill for Team Rocket.

They need a secure channel to discuss, and choose axolotl since axolotls look a bit like pokemon.

AXOLOTL: THE AXAMPLE

session establishment with 3DHE

Ash
`b33f d00d`

Brock
`badb 00da`

AXOLOTL: THE AXAMPLE

session establishment with 3DHE

Ash
`b33f d00d`

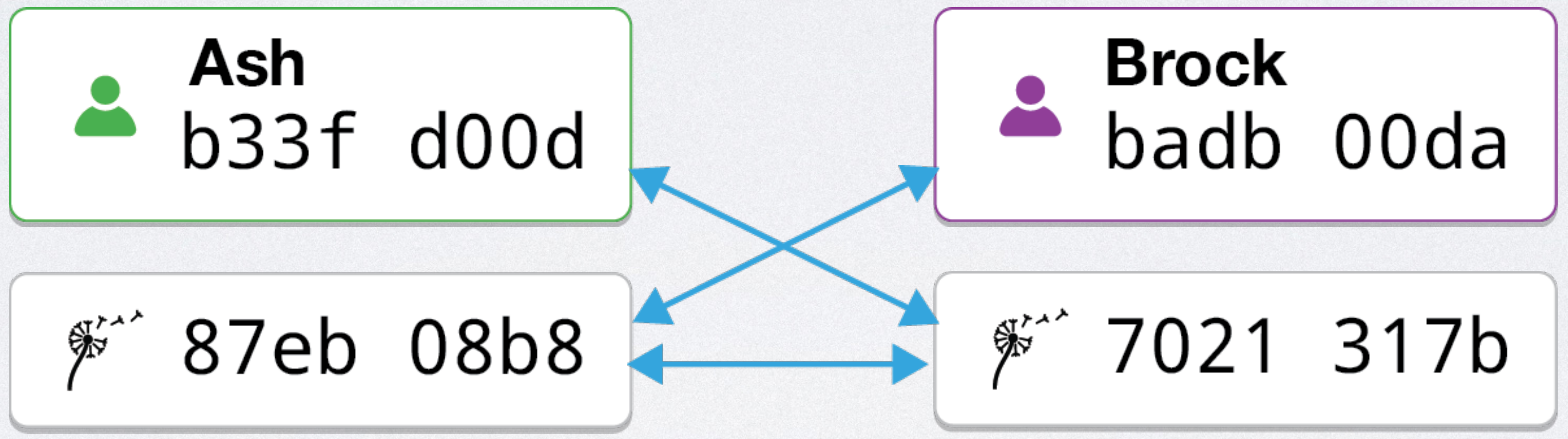
`87eb 08b8`

Brock
`badb 00da`

`7021 317b`

AXOLOTL: THE AXAMPLE

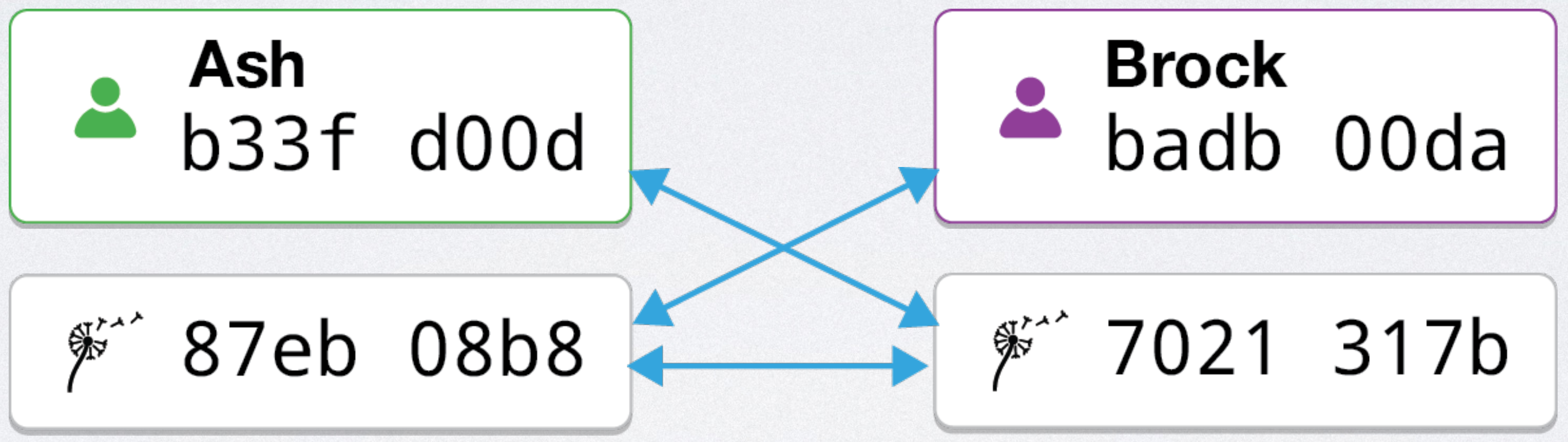
session establishment with 3DHE



🟦 = Diffie-Hellman

AXOLOTL: THE AXAMPLE

session establishment with 3DHE

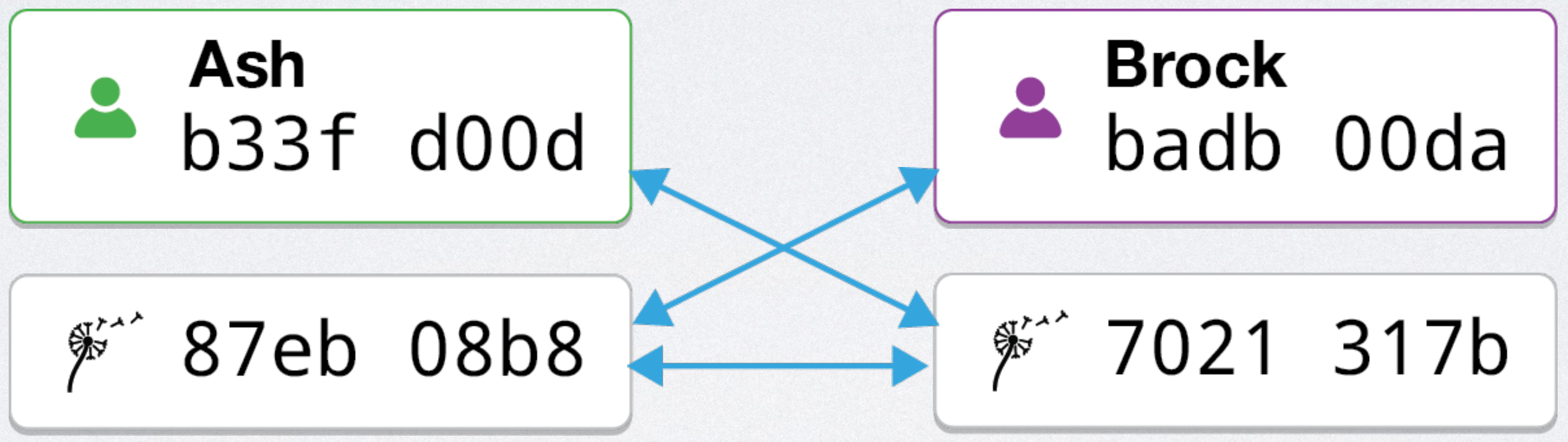


DH(A👤, B) || DH(B👤, A) || DH(A, B)

🟦 = Diffie-Hellman

AXOLOTL: THE AXAMPLE

session establishment with 3DHE

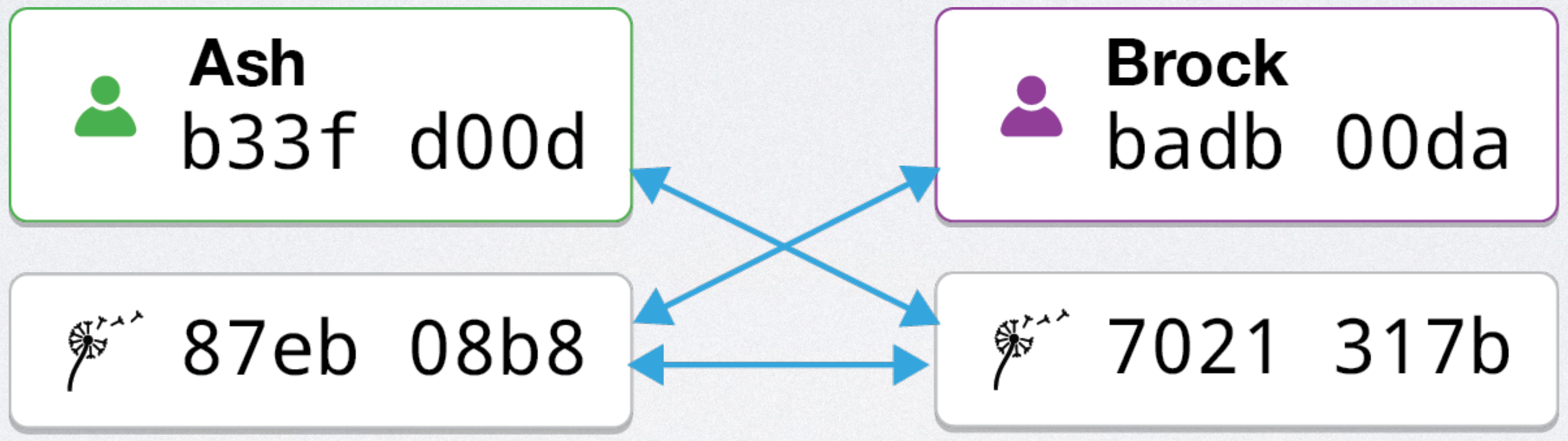


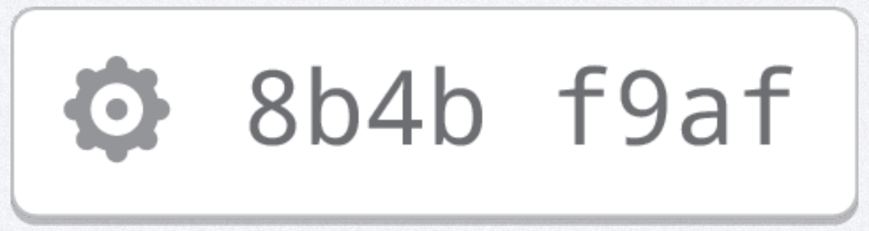
hash (DH (A, B) || DH (B, A) || DH (A, B))

🟦 = Diffie-Hellman

AXOLOTL: THE AXAMPLE

session establishment with 3DHE

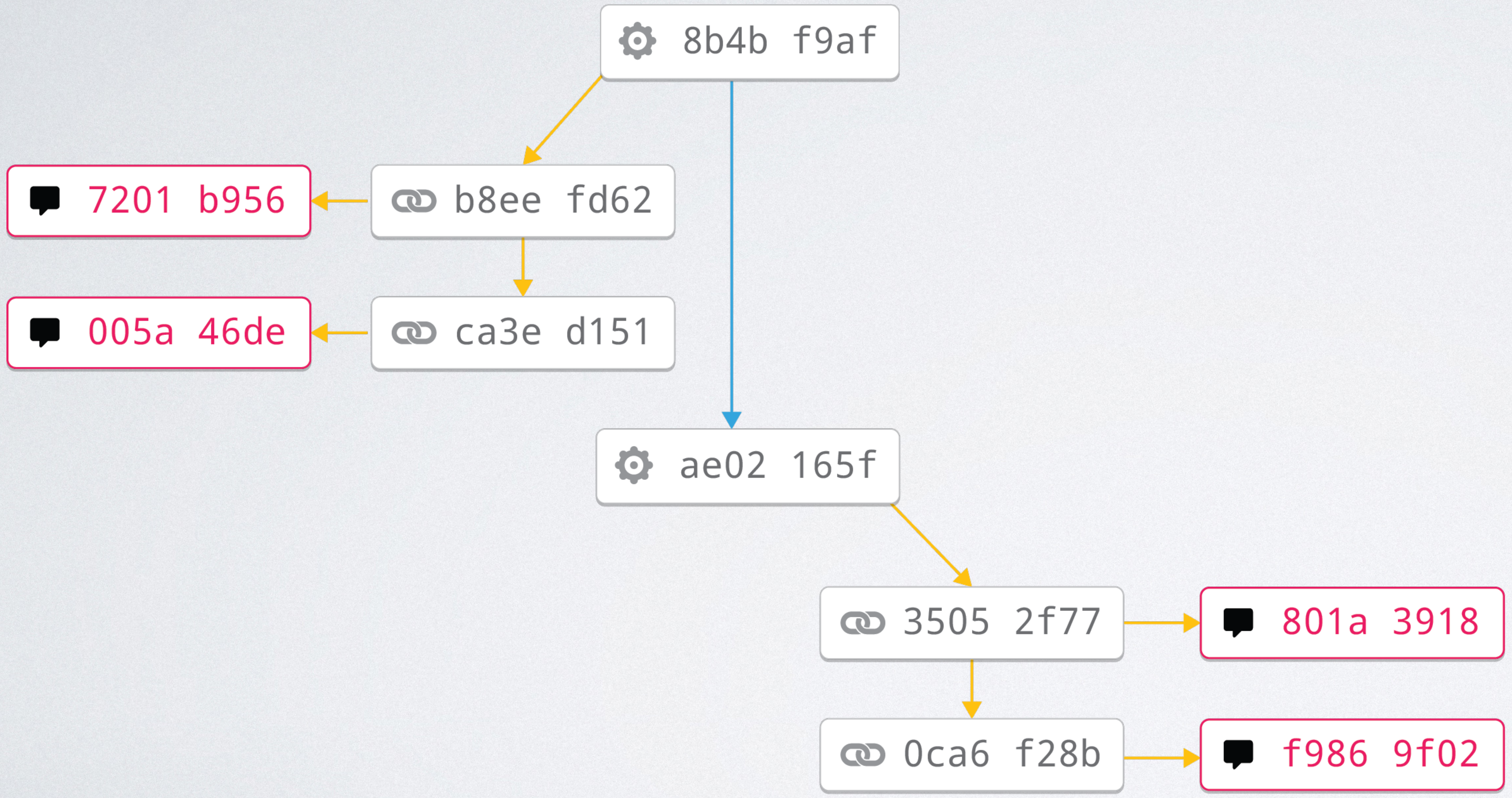




🟦 = Diffie-Hellman

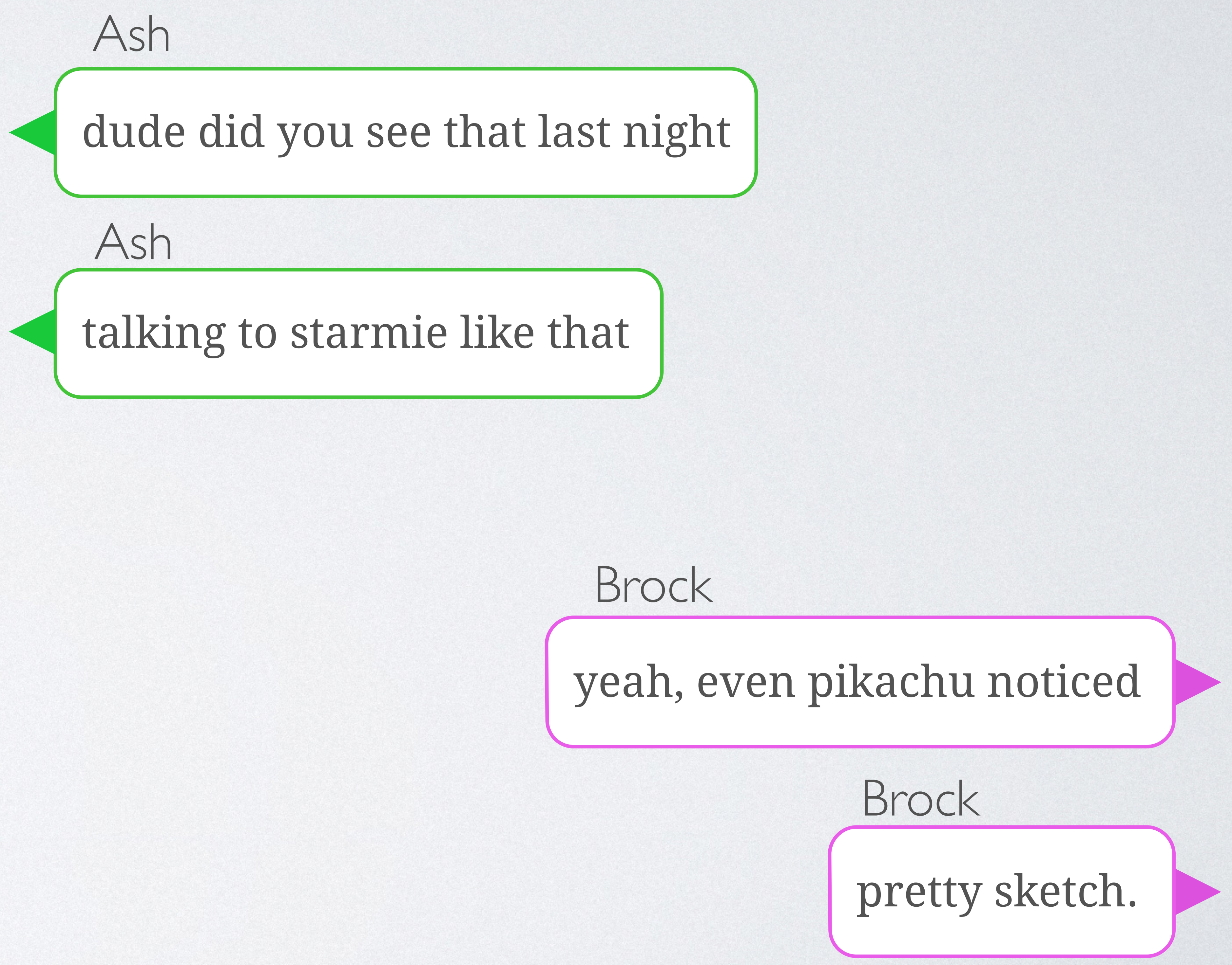
AXOLOTL: THE AXAMPLE

ratcheting



🟨 = Hash Function

🟦 = Diffie-Hellman Exchange


Ash
dude did you see that last night

Ash
talking to starmie like that

Brock
yeah, even pikachu noticed

Brock
pretty sketch.

AXOLOTL: THE AXAMPLE

summary: wins

- Ideal Diffie-Hellman ratcheting, ephemeral keys change as fast as possible.

- In between ratchets, the chain key means message keys are never reused, giving even more protection for forward and future secrecy.

- Not covered here in detail, but message loss is OK since you can cache message keys generated while ratcheting the chain forward.

axolotl over an optimized network protocol

REGISTRATION

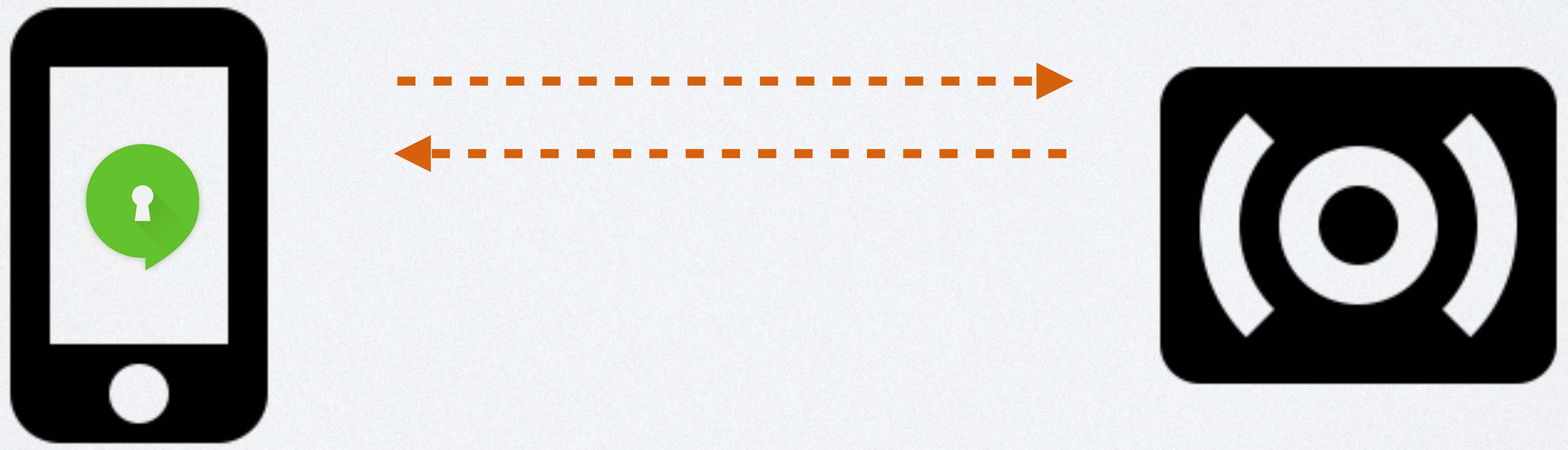
REGISTRATION

account establishment {

1) phone requests to verify number

REGISTRATION

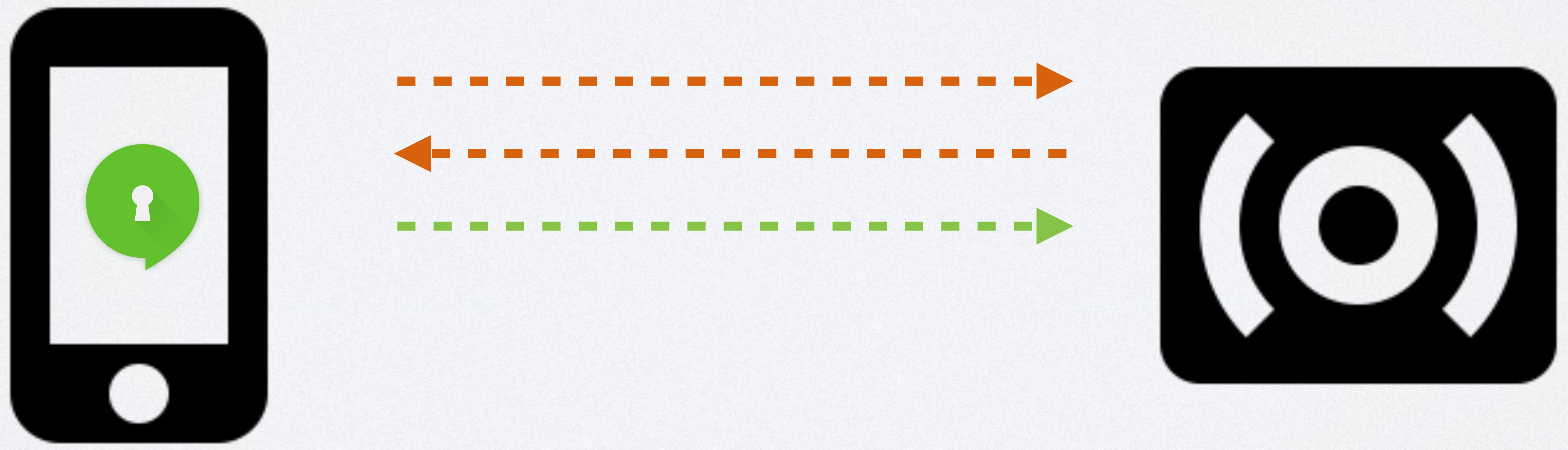
account establishment {
1) phone requests to verify number
2) server sends challenge via SMS/call



REGISTRATION

account establishment {
1) phone requests to verify number
2) server sends challenge via SMS/call

identity/key management {
3) phone generates long-term identity key and uploads
4) phone generates 100 ephemeral “pre-keys”, uploads pre keys w/ IDs

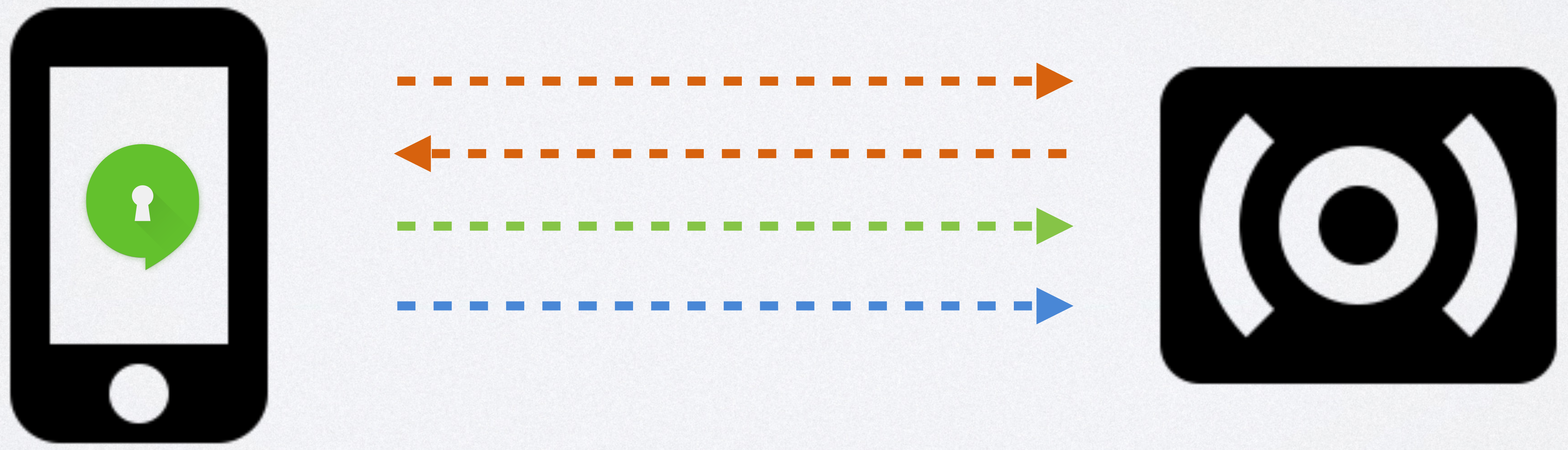


REGISTRATION

account establishment {
1) phone requests to verify number
2) server sends challenge via SMS/call

identity/key management {
3) phone generates long-term identity key and uploads
4) phone generates 100 ephemeral "pre-keys", uploads pre keys w/ IDs

contact discovery {
5) phone uploads list of hashed contact numbers

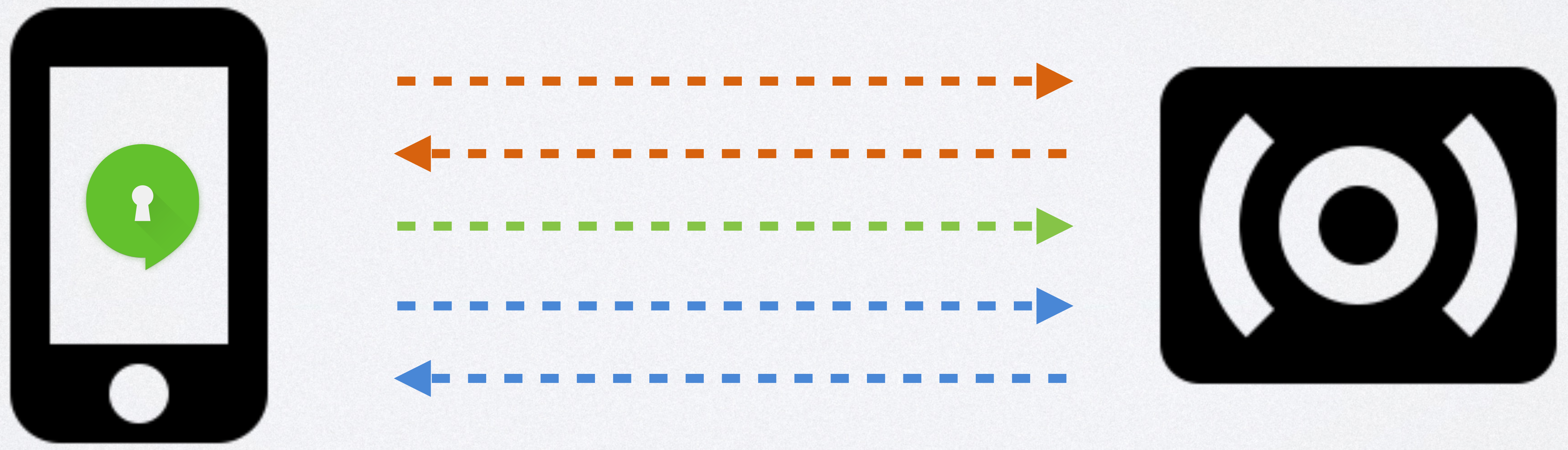


REGISTRATION

account establishment {
1) phone requests to verify number
2) server sends challenge via SMS/call

identity/key management {
3) phone generates long-term identity key and uploads
4) phone generates 100 ephemeral "pre-keys", uploads pre keys w/ IDs

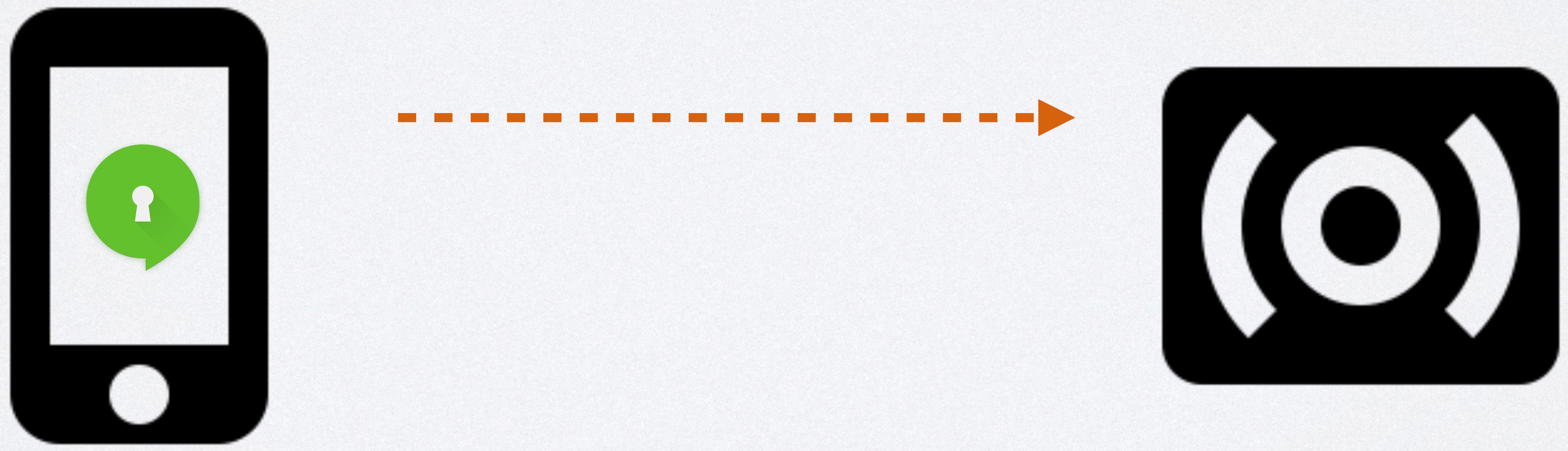
contact discovery {
5) phone uploads list of hashed contact numbers
6) server responds back with list of contacts registered on it



SENDING MESSAGES

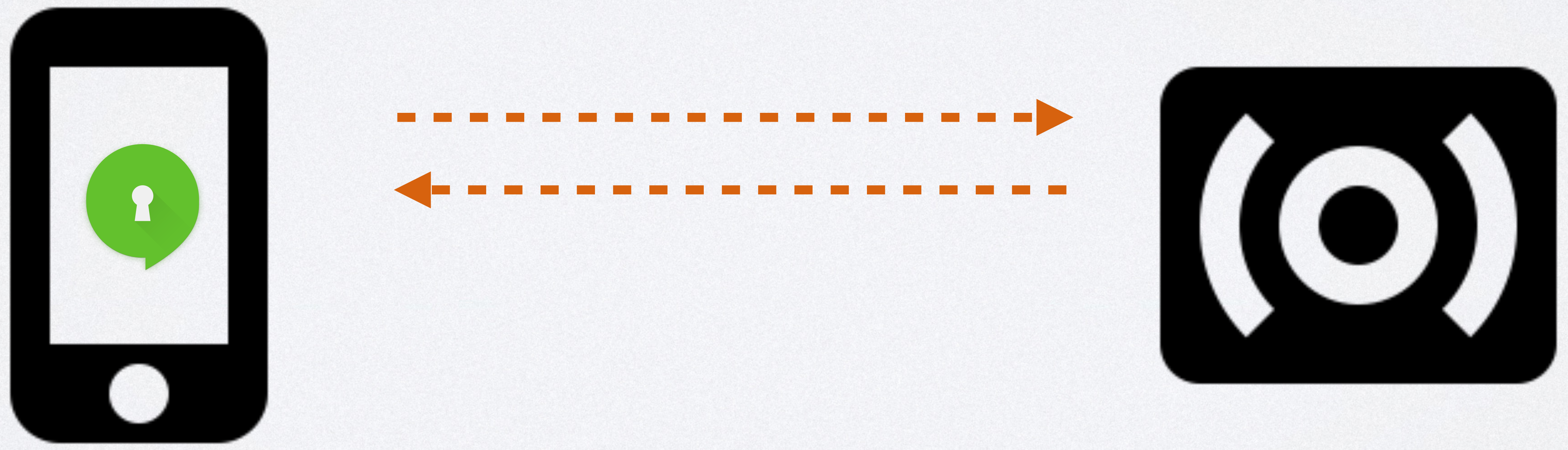
SENDING MESSAGES

keys { 1) phone asks for public key and pre-key if one is not cached.



SENDING MESSAGES

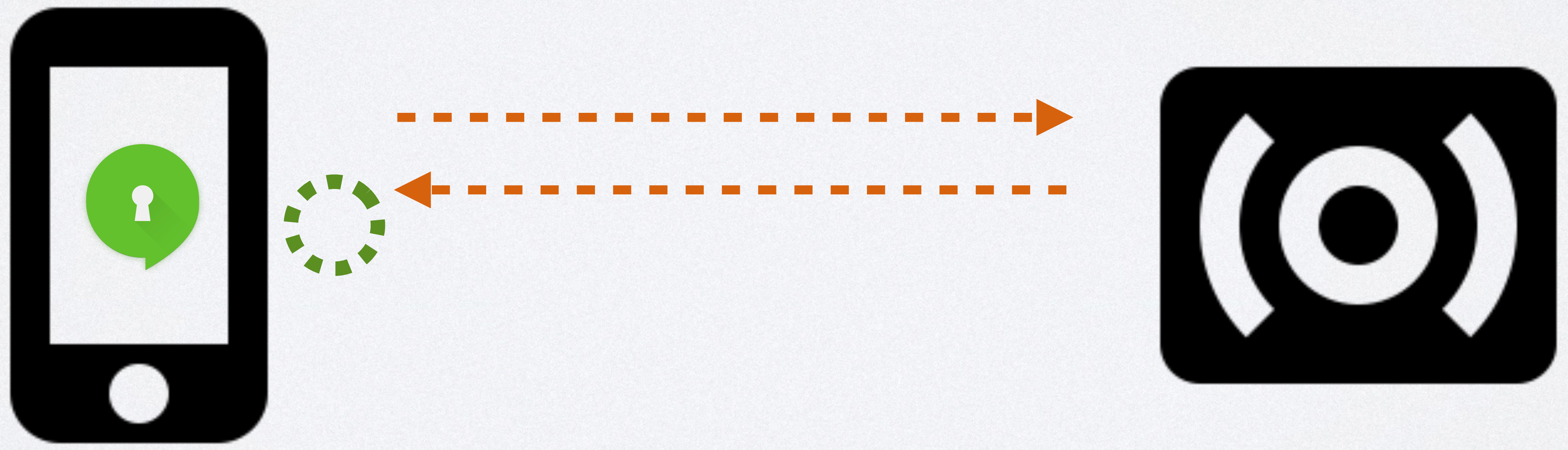
keys {
1) phone asks for public key and pre-key if one is not cached.
2) server sends back public key and an unused pre-key



SENDING MESSAGES

keys {
1) phone asks for public key and pre-key if one is not cached.
2) server sends back public key and an unused pre-key

ratchet {
3) phone generates an ephemeral key
4) phone does 3DHE to derive master secret

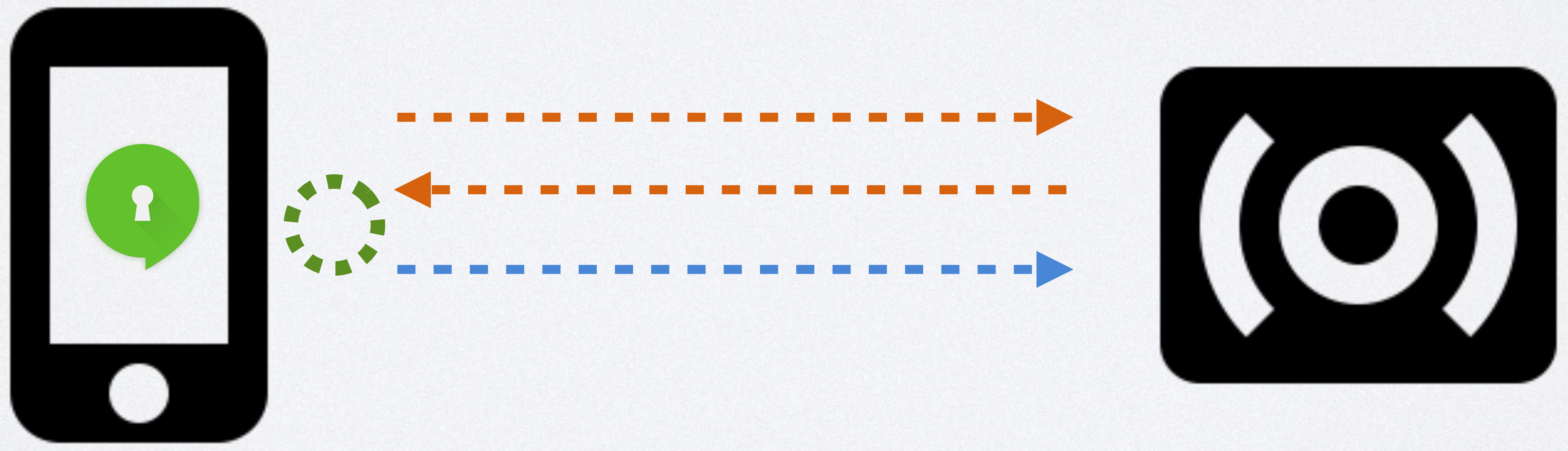


SENDING MESSAGES

keys {
1) phone asks for public key and pre-key if one is not cached.
2) server sends back public key and an unused pre-key

ratchet {
3) phone generates an ephemeral key
4) phone does 3DHE to derive master secret

delivery {
5) phone sends encrypted message for server to pass along

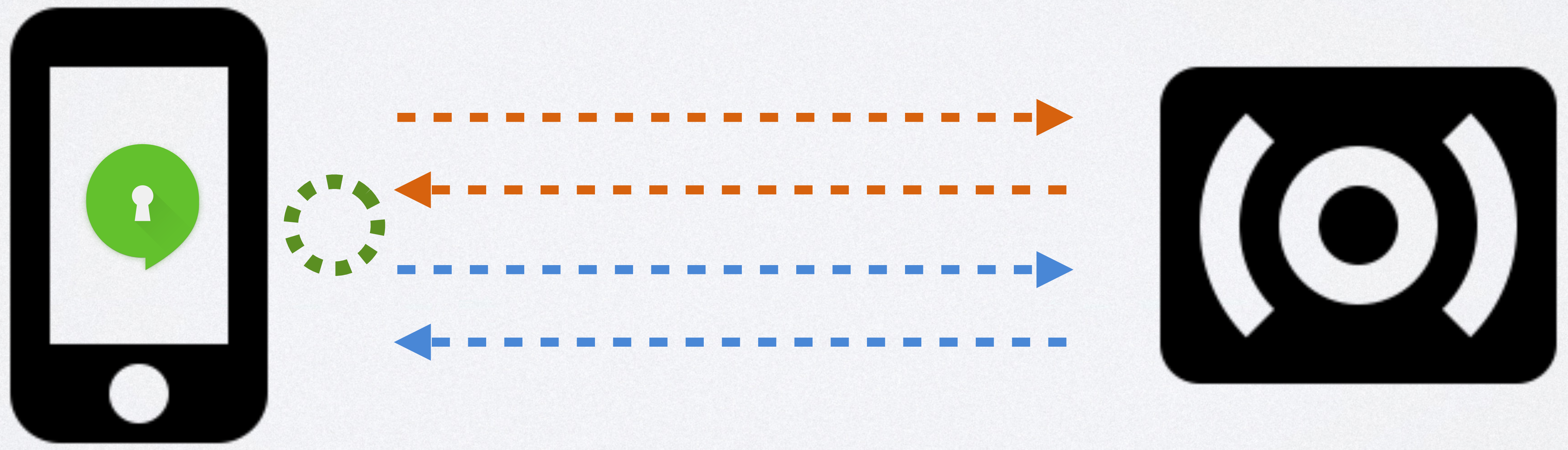


SENDING MESSAGES

keys {
1) phone asks for public key and pre-key if one is not cached.
2) server sends back public key and an unused pre-key

ratchet {
3) phone generates an ephemeral key
4) phone does 3DHE to derive master secret

delivery {
5) phone sends encrypted message for server to pass along
6) server responds with status



SENDING MESSAGES

KEY VALIDATION MODEL

SENDING MESSAGES

KEY VALIDATION MODEL

raw, uncut TOFU

SENDING MESSAGES

KEY VALIDATION MODEL

1) At first retrieval of a user's identity key, we Trust On First Use. (**TOFU**)
   think SSH without the initial fingerprint notification that nobody ever verifies

2) If their public key changes, we alert the user and await their approval

3) We provide a UI for fingerprint verification via a side-channel

SENDING MESSAGES

KEY VALIDATION MODEL

because of this simplification:

users don’t need to know what a key is
users don’t need to know what a fingerprint is

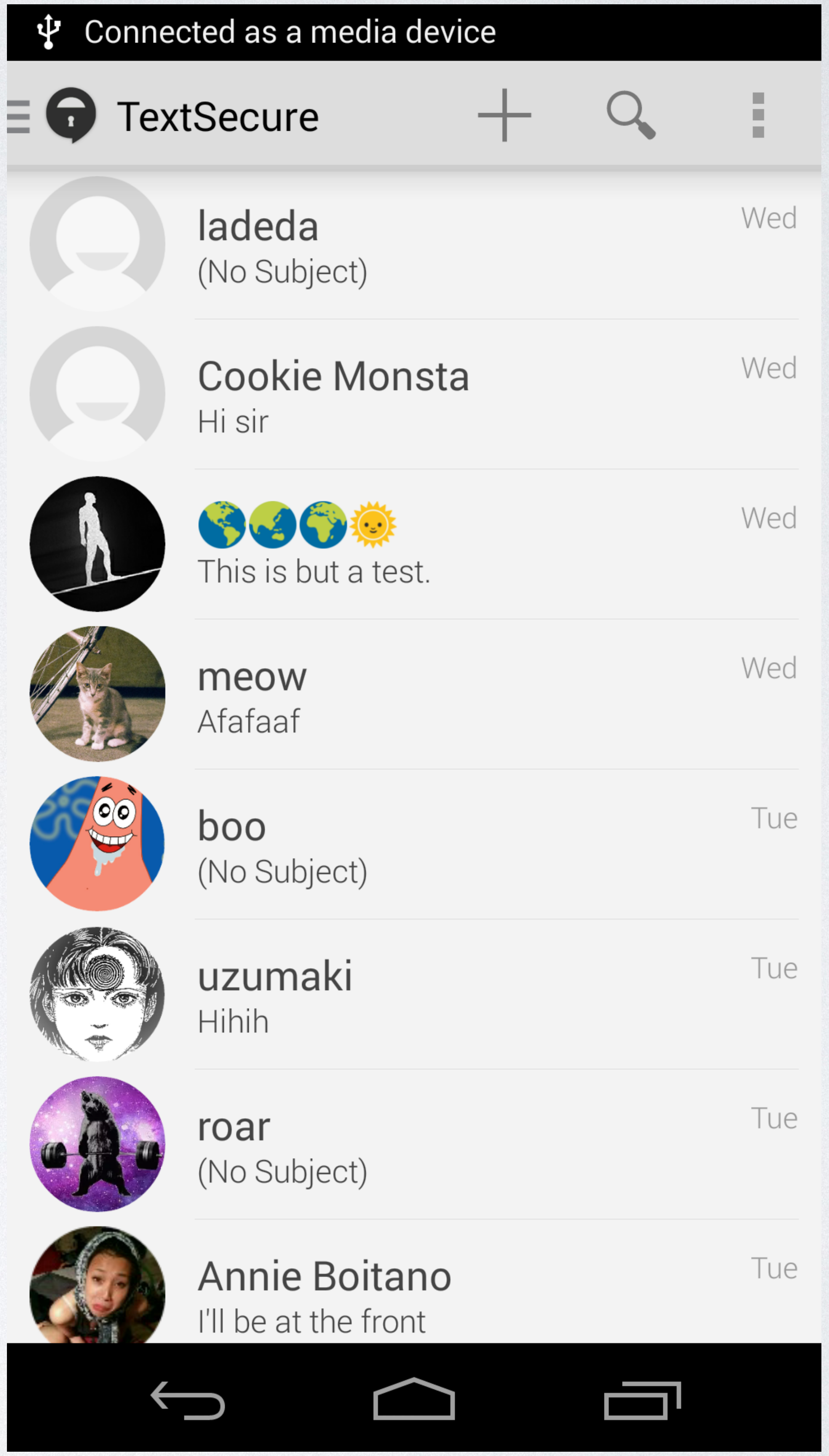
… if they don’t want to.
(and most people in the world don’t want to)

WHAT DO WE WIN?!

SIMPLIFICATION

something that looks just like every other messenger,

which is exactly the starting point we want for encrypted messaging.


Connected as a media device

ladeda (No Subject)	Wed
Cookie Monsta Hi sir	Wed
🌎🌏🌍🌞 This is but a test.	Wed
meow Afafaaf	Wed
boo (No Subject)	Tue
uzumaki Hihih	Tue
roar (No Subject)	Tue
Annie Boitano I'll be at the front	Tue

SIMPLIFICATION

all this was about simplification the whole time

STARTING FROM SCRATCH

By designing both the cryptographic ***and*** network protocol from scratch, we get better stability and usability.

Crypto on SMS is painful.
Crypto on XMPP is painful.
Crypto on transports you don't control is painful.

ALL THE SIMPLIFICATIONS

- An entire transport that's always encrypted and **just fucking works**.
- A **drastically simplified UX**, and that's the golden victory.
- Grandma no longer needs to know crypto lingo to benefit from end-to-end privacy.

ALL THE SIMPLIFICATIONS

All these technical choices build up to a system that may actually be ready for mass adoption.

Mass adoption is what pisses off the surveillance state.

ANGERING EVE

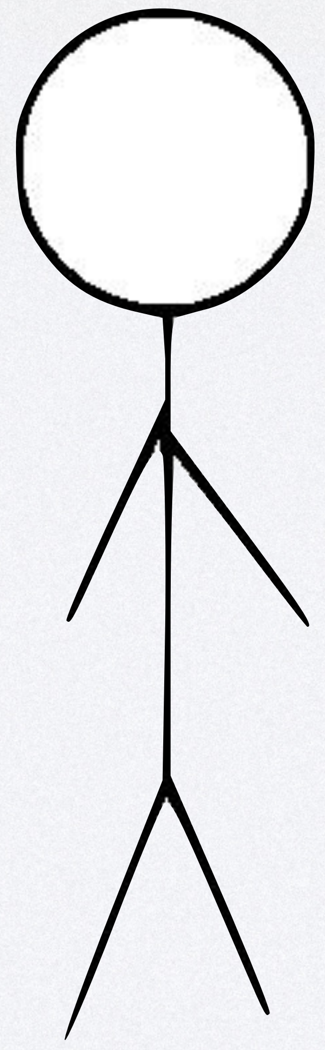
Eve is afraid of ubiquitous end-to-end encryption that isn’t broken.

ANGERING EVE

Usability issues are security issues.

CALL TO ARMS

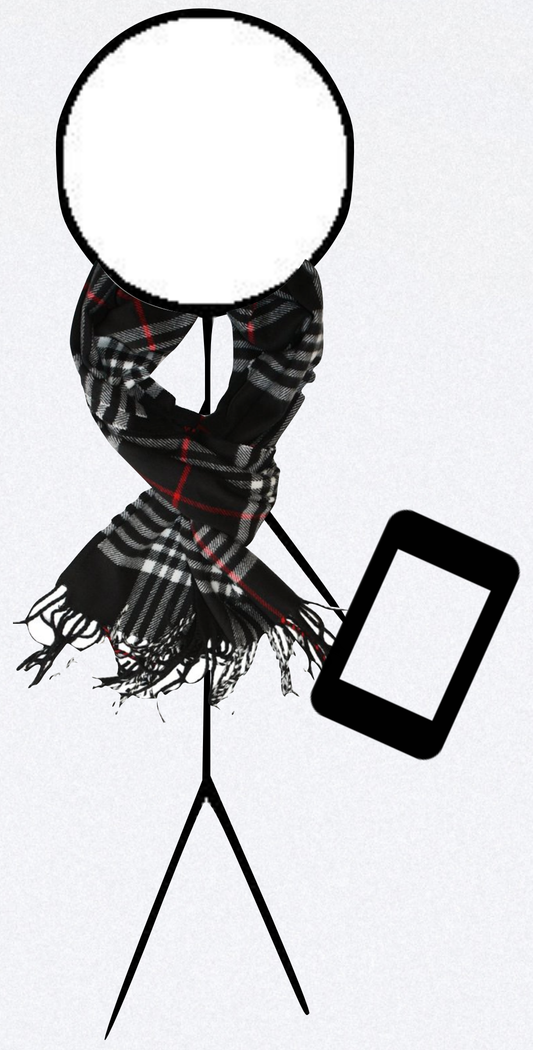
IF YOU CODE,



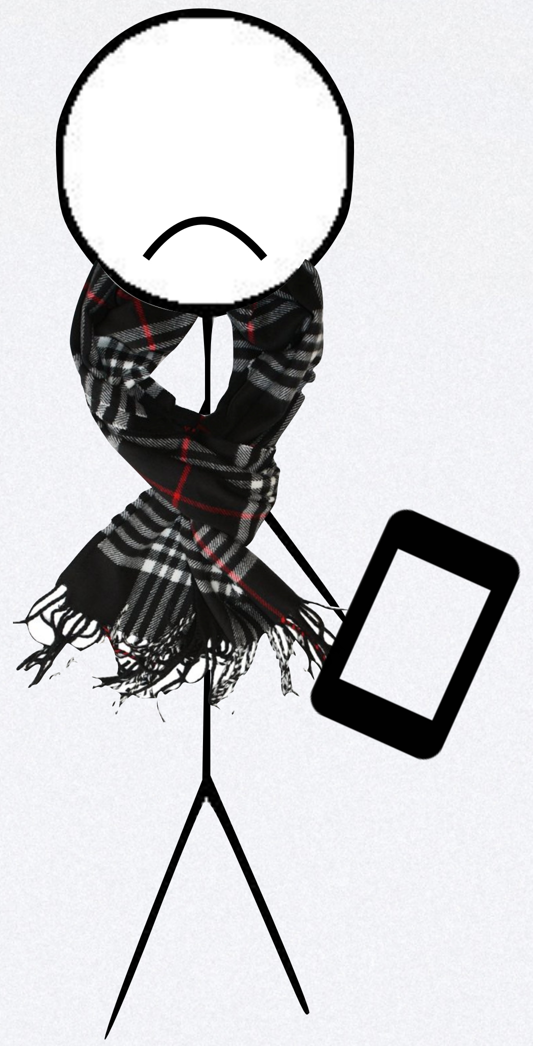
IF YOU CODE, YOU DESIGN

IF YOU DESIGN,

IF YOU DESIGN, USABILITY MATTERS.



IF YOUR SOFTWARE ISN'T USABLE, NOBODY USES IT.



USABLE CRYPTO FUCKS WITH THE SURVEILLANCE STATE.

thanks from OWS