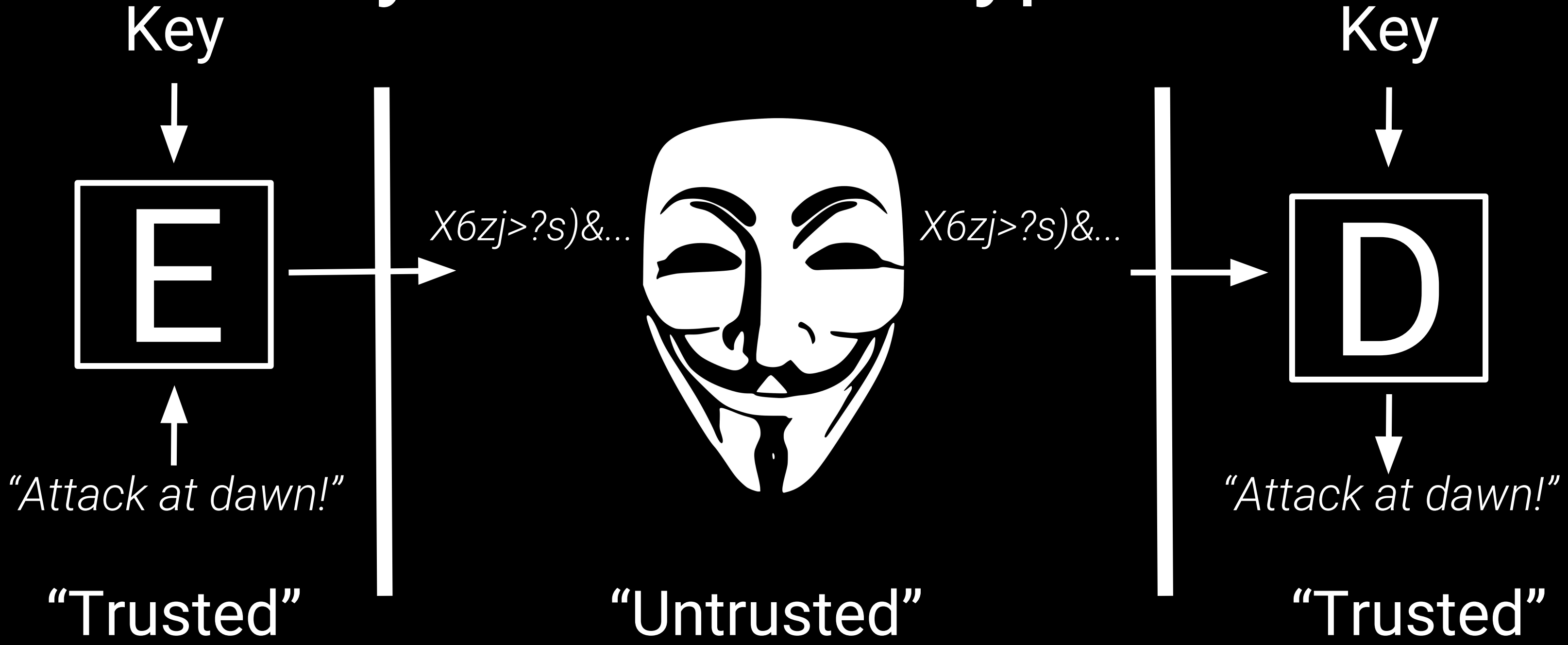
Let’s Encrypt

Sagi Kedmi

Battleground

“Trusted” | “Untrusted” | “Trusted”

Symmetric Encryption

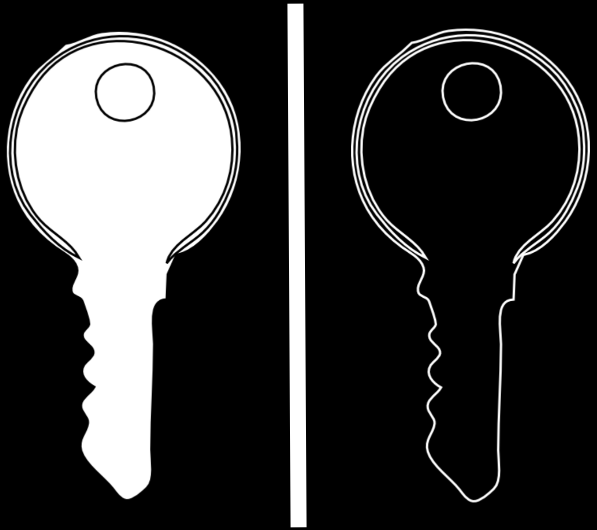


The 70s

Public Key Crypto

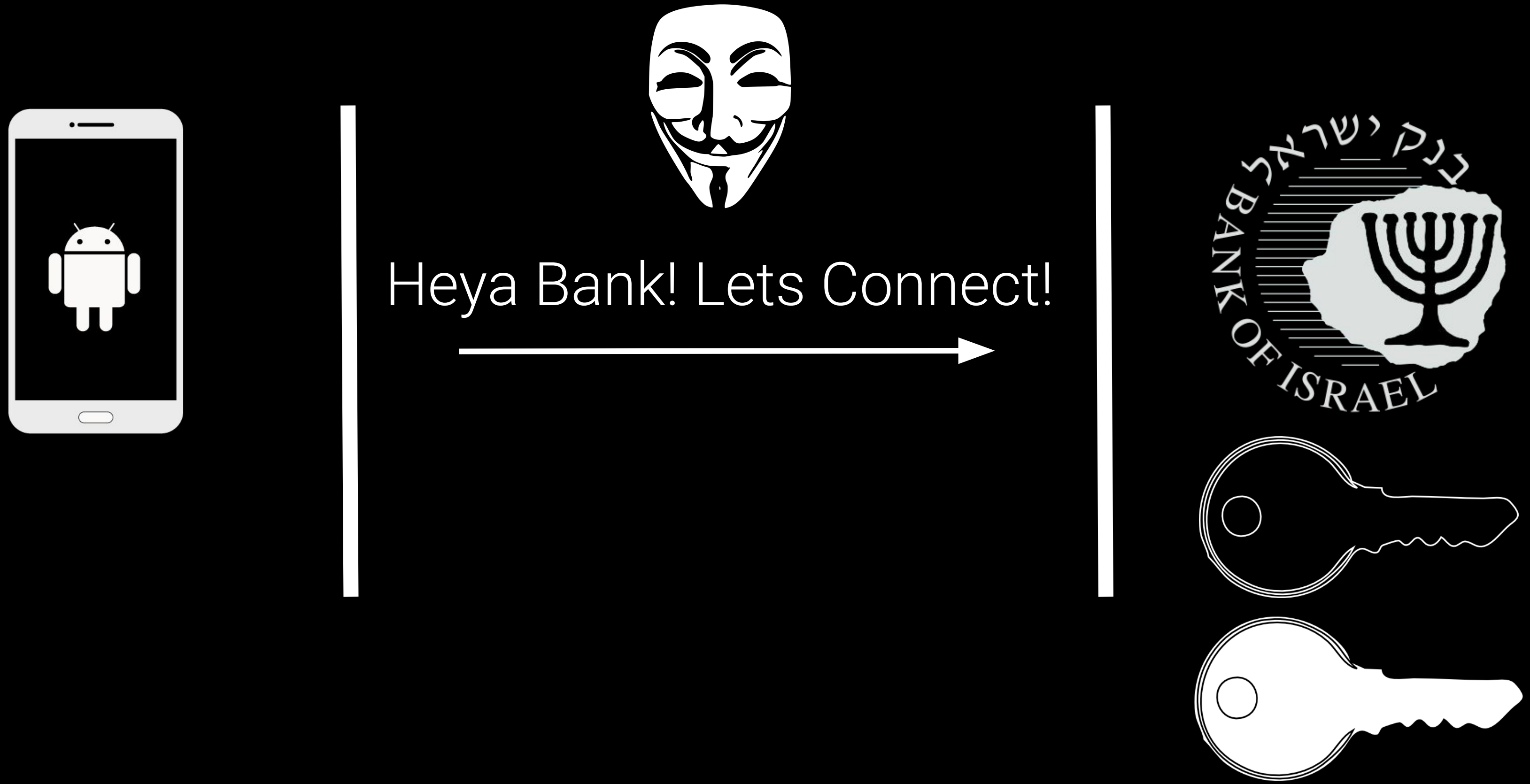
Public Key Crypto
(Asymmetric Encryption)

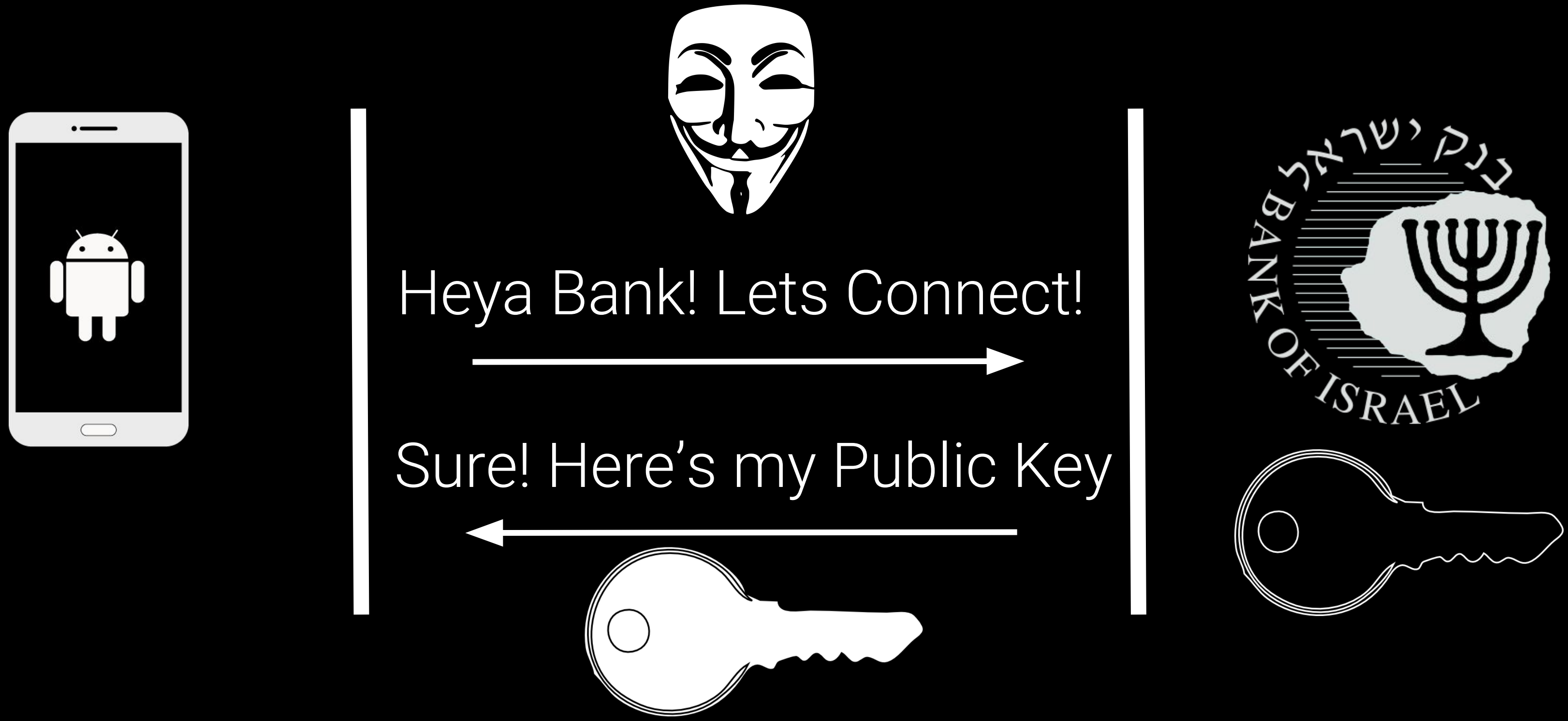
- Public knowledge
- Anything **encrypted** with it can only be decrypted using the **Private Key**



- Kept secret
- Anything “**encrypted”*** with it can only be decrypted using the **Public Key**

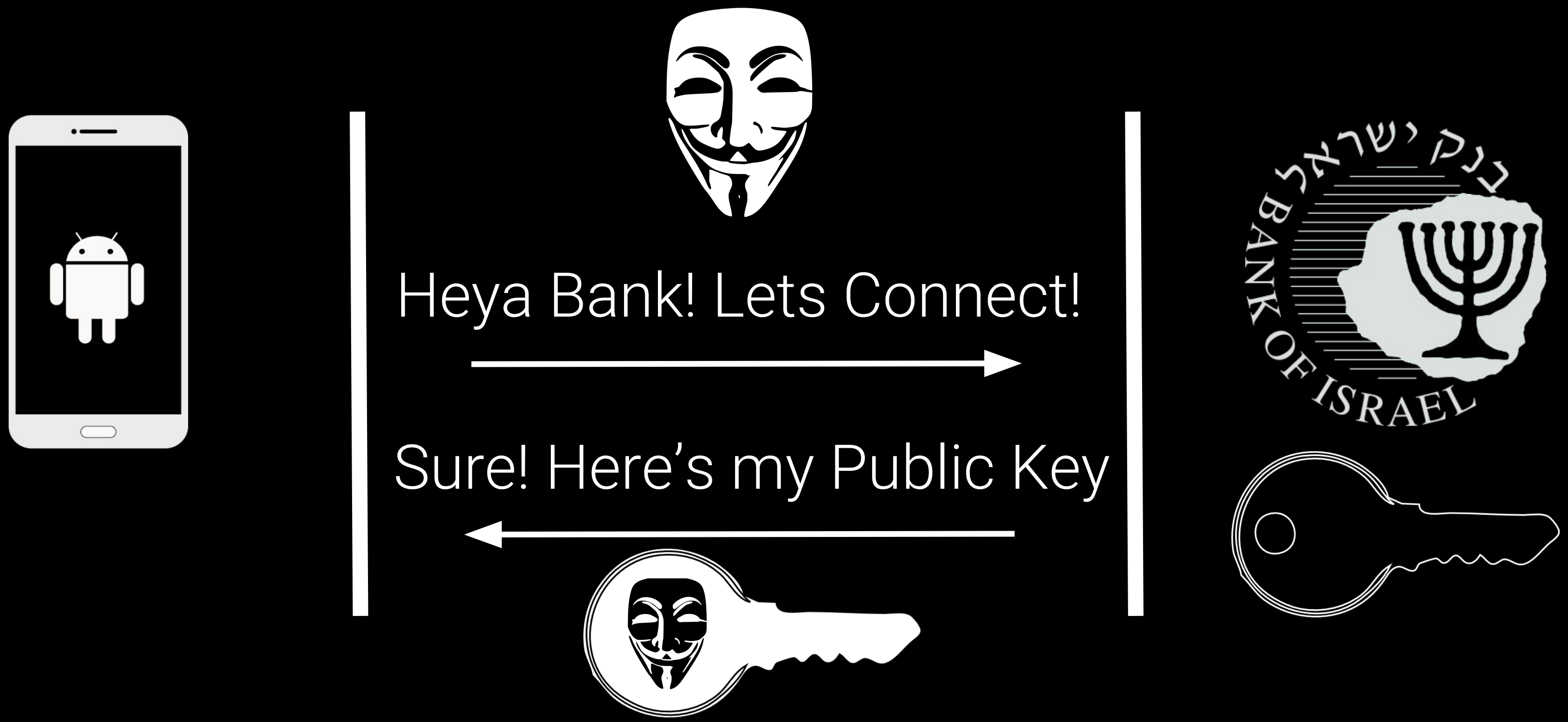
** Digital Signature*


Heya Bank! Lets Connect!
בנק ישראל
BANK OF ISRAEL



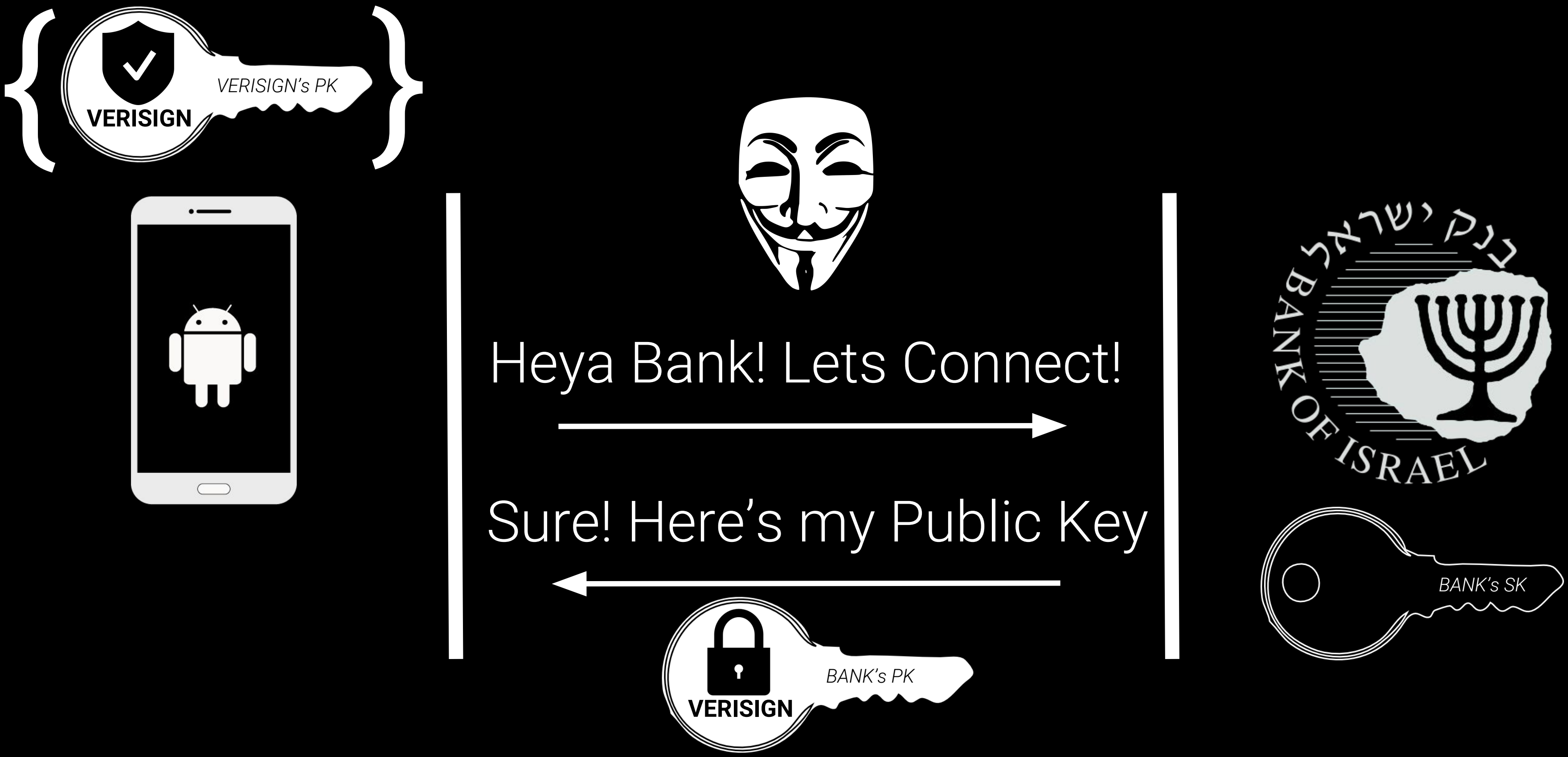
Heya Bank! Lets Connect!

Sure! Here’s my Public Key



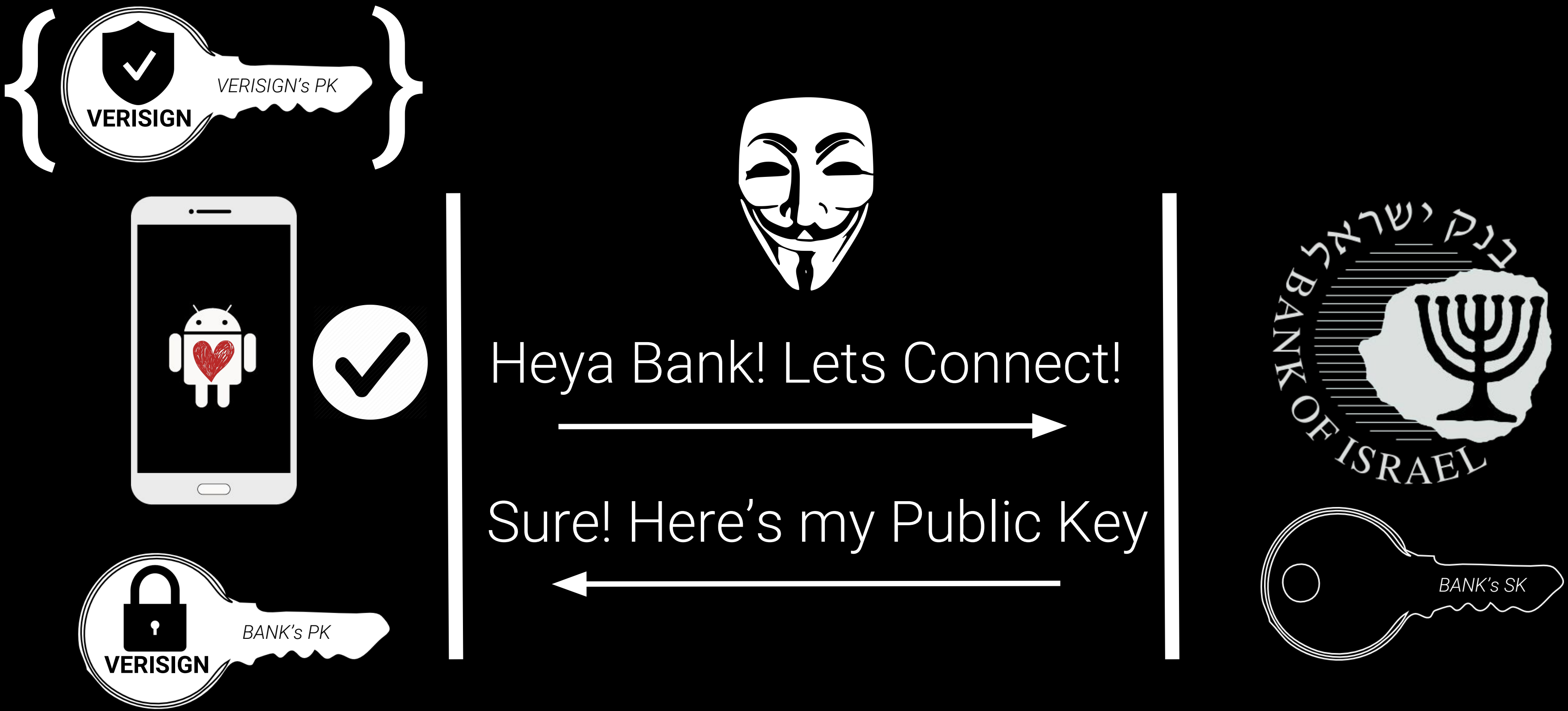
Heya Bank! Lets Connect!

Sure! Here’s my Public Key



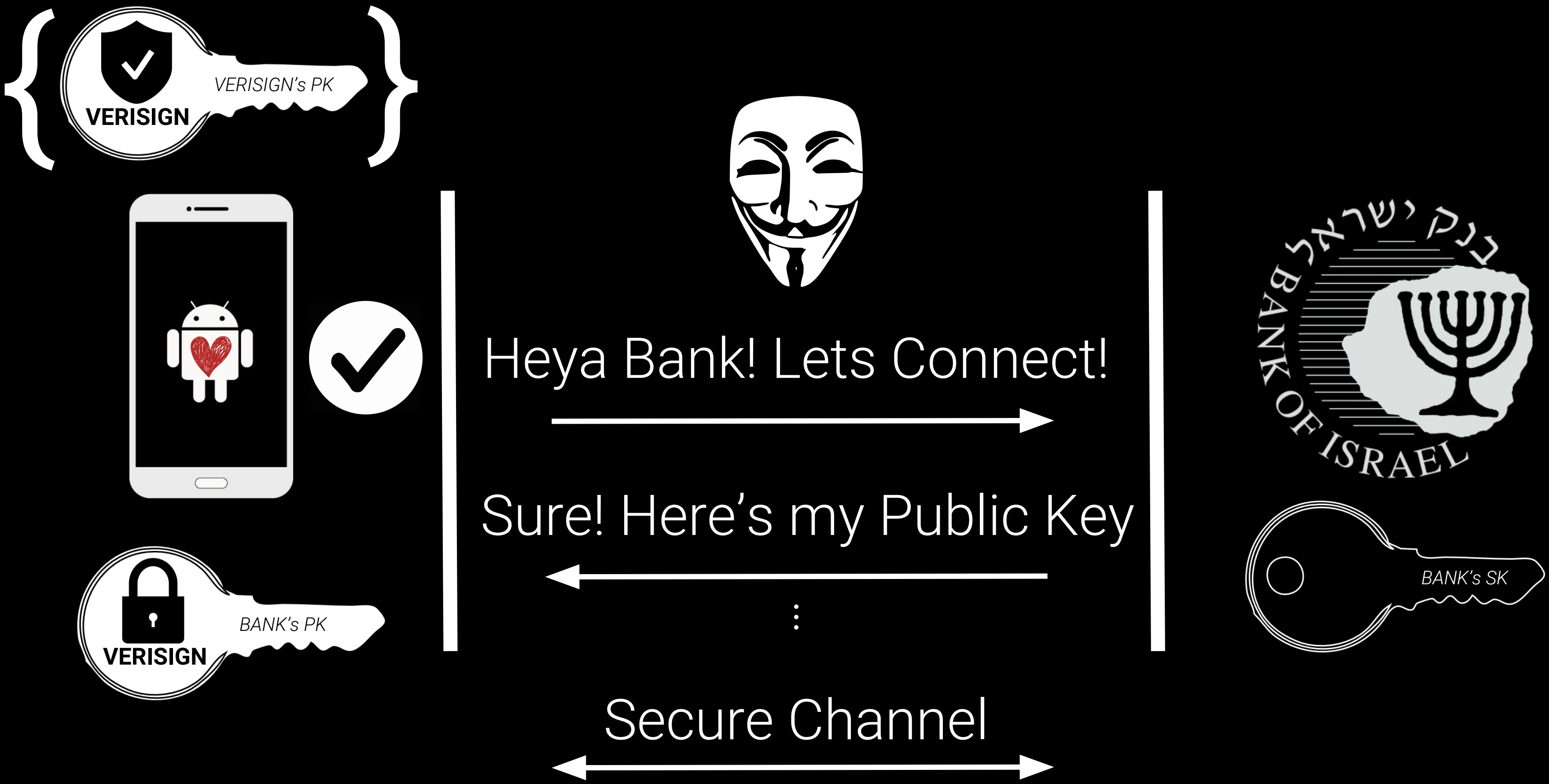
Heya Bank! Lets Connect!

Sure! Here's my Public Key



Heya Bank! Lets Connect!

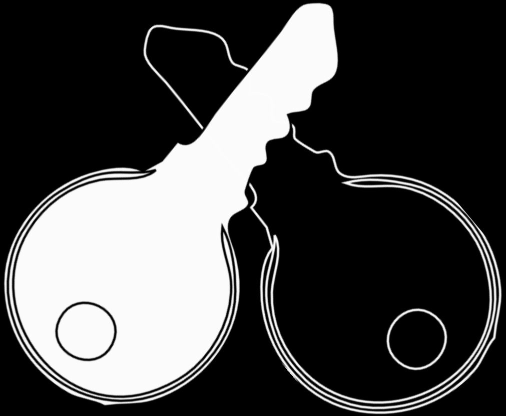
Sure! Here's my Public Key


VERISIGN's PK
VERISIGN
Heya Bank! Lets Connect!
Sure! Here's my Public Key
⋮
Secure Channel
BANK's PK
VERISIGN
בנק ישראל
BANK OF ISRAEL
BANK's SK

Key Generation*

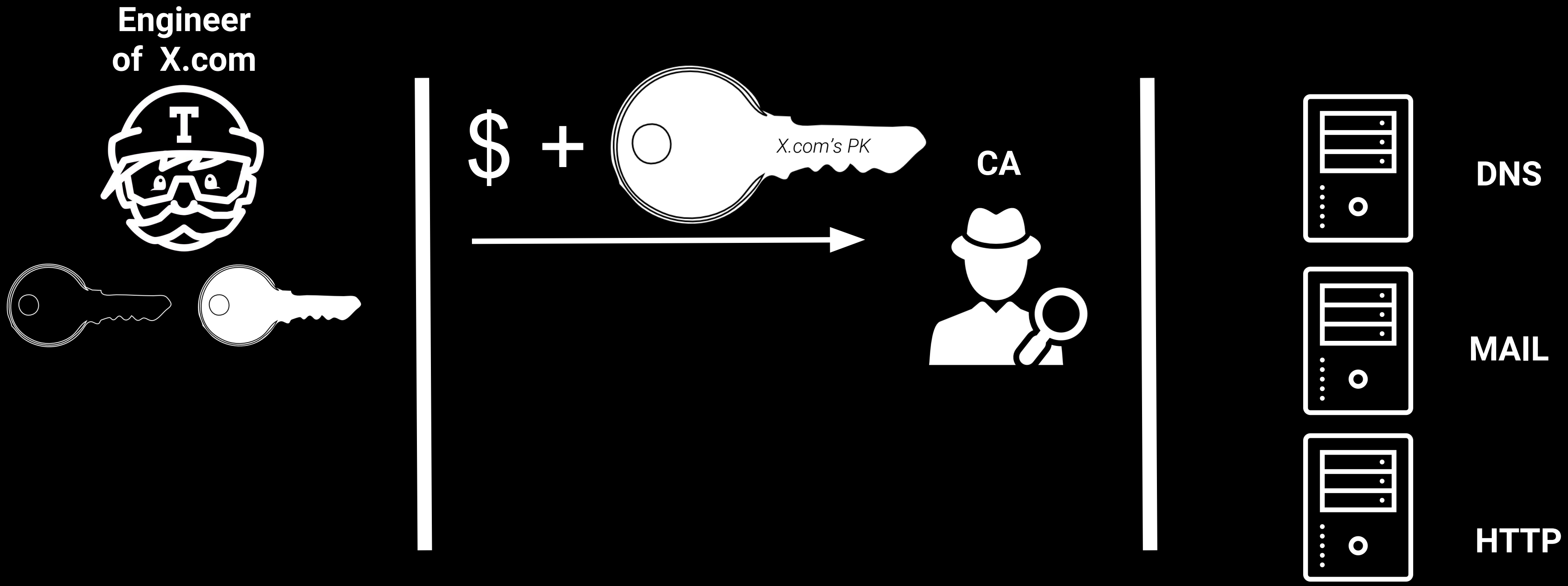
**Before Let's Encrypt*

```
$ openssl req -nodes
-newkey rsa:4096 \
-keyout secret.key \
-out request.csr \
-subj \
"/C=IL/ST=Tel-Aviv/L=Tel-Av
iv/O=Rumors/OU=Engineering/
CN=rumors.io"
```

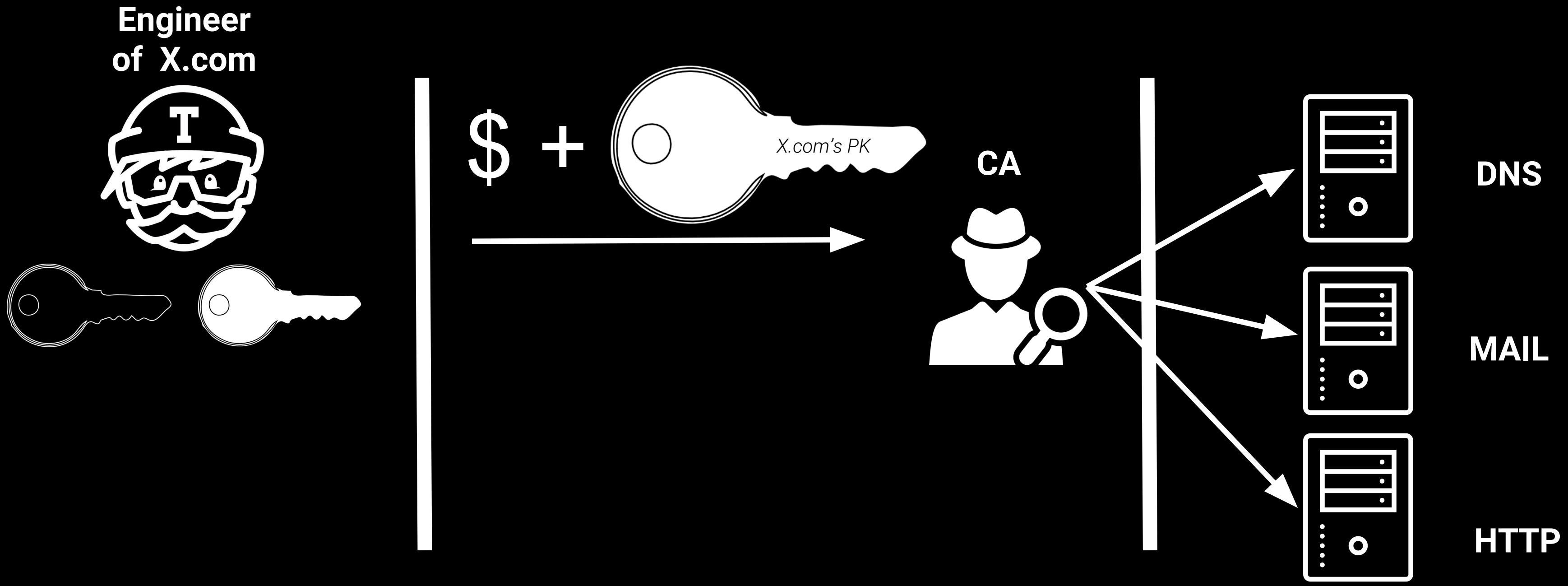


```
*View SK/PK: $ openssl rsa -noout -text -in secret.key
*View CSR:   $ openssl req -noout -text -in request.csr
```

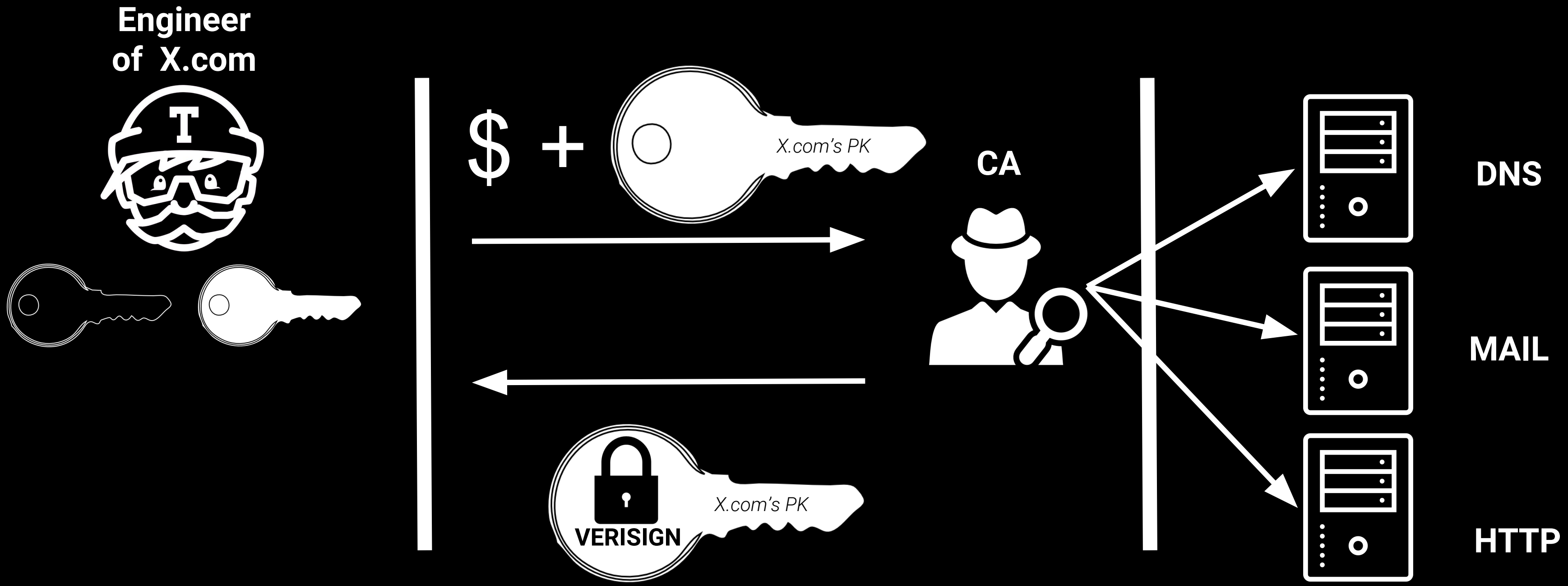
CA Domain Validation



CA Domain Validation



CA Domain Validation



Let’s Encrypt

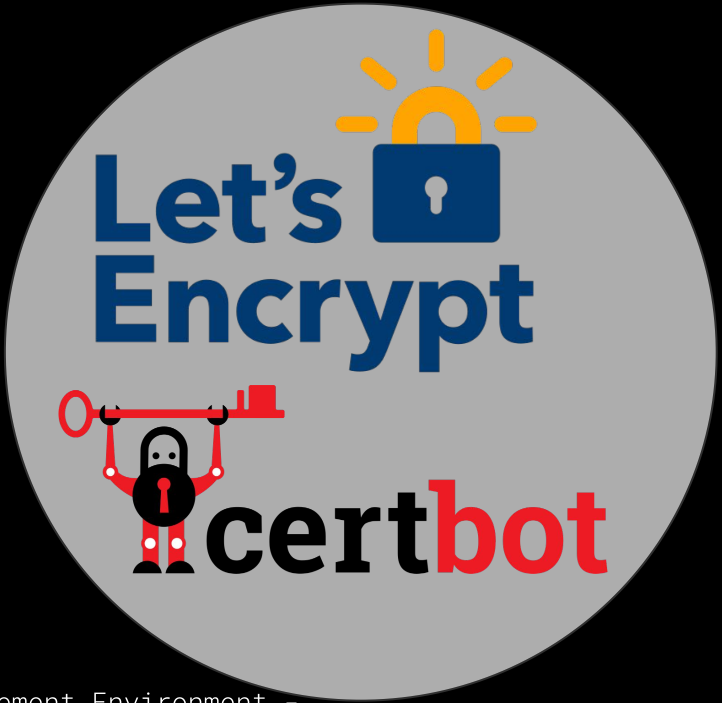
- A FREE and Automated CA, gets you a browser-trusted certificate if one can prove domain ownership.
- Speaks the ACME* protocol
- Many clients** exists, **certbot** (aka Let’s Encrypt client) is the recommended one.

* Automated Certificate Management Environment - https://tools.ietf.org/html/draft-ietf-acme-acme-07
** LE Clients: https://letsencrypt.org/docs/client-options/

certbot

- Developed by the EFF

- What does it do?
  - Generates a key-pair
  - Uses ACME to validate domain ownership via Let's Encrypt's CA
  - Installs the legit Cert
  - Sets **secure** ciphersuites
  - Allows other security settings
    - HSTS, OCSP Stapling/Must-Staple, HTTPS Redirection, CSP: Upgrade-Insecure-Reqs



* Automated Certificate Management Environment - https://tools.ietf.org/html/draft-ietf-acme-acme-07
** LE Clients: https://letsencrypt.org/docs/client-options/

SSL/TLS Attacks

- **CA Compromise -** e.g. DigiNotar
- **PRNG Fails -** e.g. Debian OpenSSL Debacle
- **Broken Crypto -** e.g. Flame Malware (MD5 Collision), RC4, DES
- **Weakened Crypto -** e.g. EXPORT ciphersuites (FREAK)
- **Protocol -** CRIME, TIME, BREACH, BEAST, DROWN LOGJAM, POODLE (many more...)

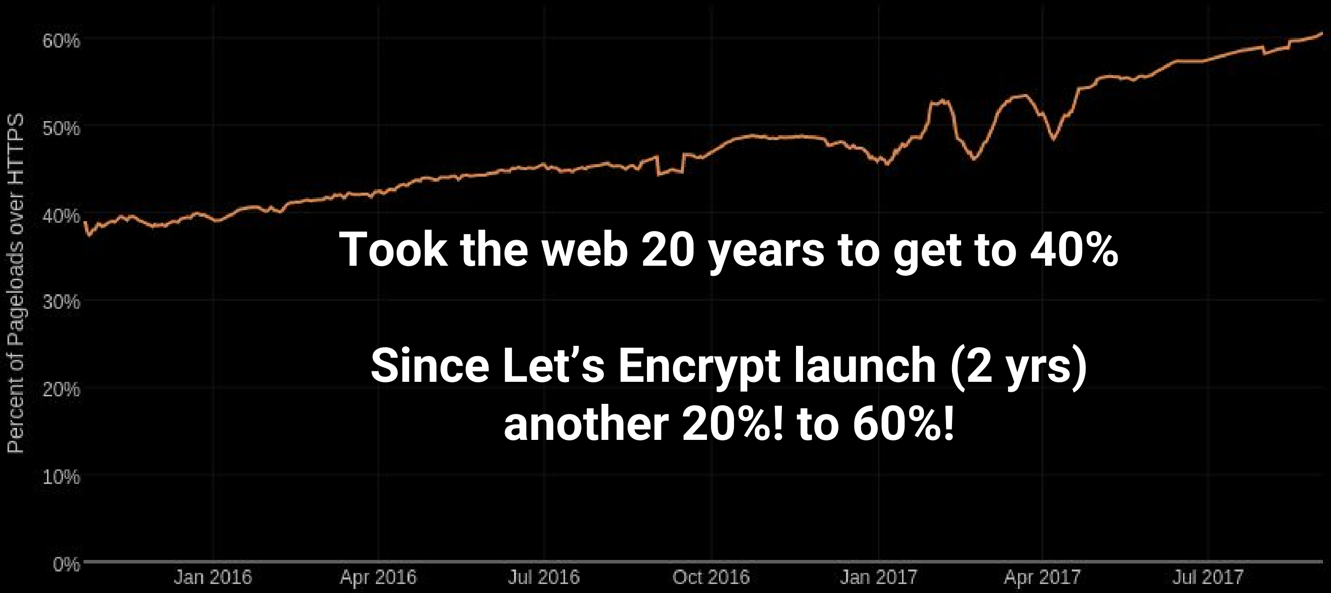
Not just the USA. Many other nation states and other sophisticated attackers.

Ciphersuites

- “Good Ciphersuites” : *at least for now … :)*

  -
    ```
    ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:AES:CAMELLIA:DES-CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA
    ```
- Disable TLS compression

Impact

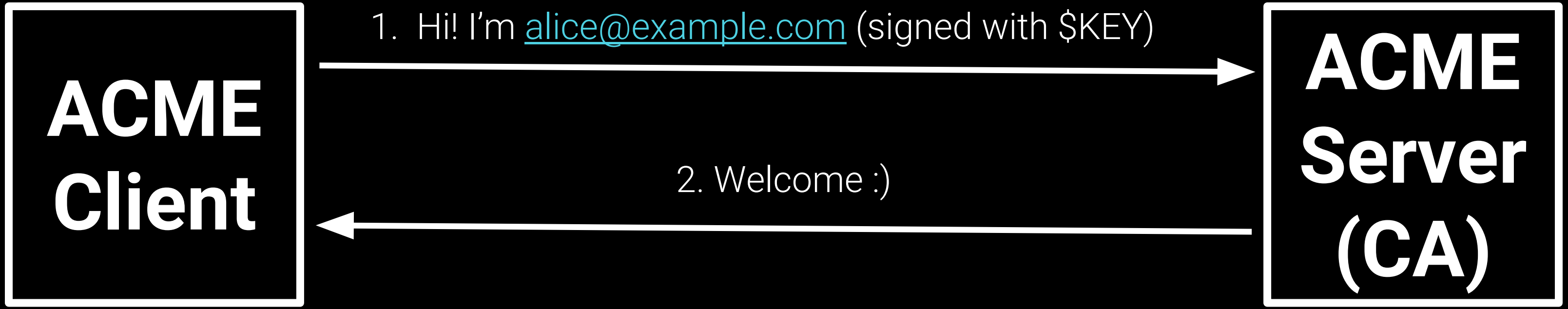


Took the web 20 years to get to 40%

Since Let's Encrypt launch (2 yrs) another 20%! to 60%!

Let’s Encrypt - How?
Create an Account

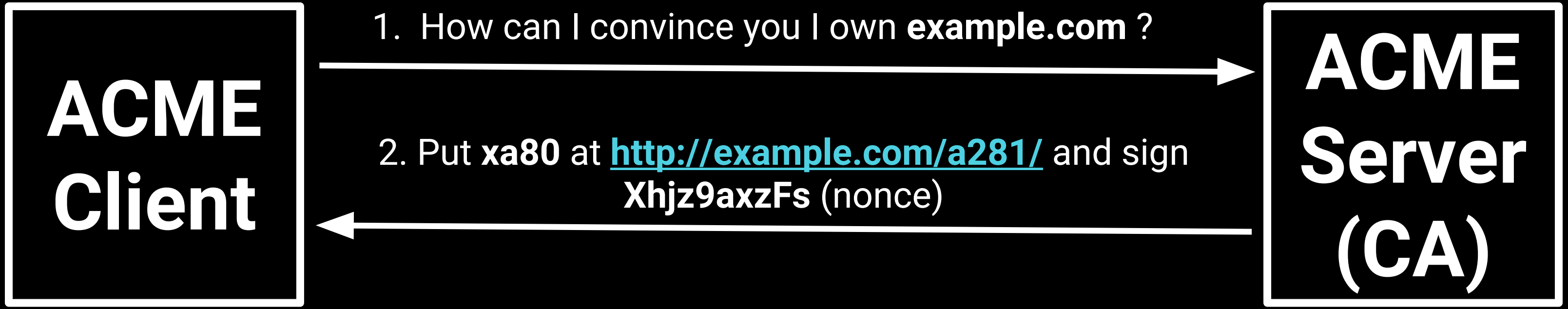
- Creates a key-pair (all future messages will be signed with it)
- Registers the key-pair with the CA



Let’s Encrypt - How?

Get a Challenge

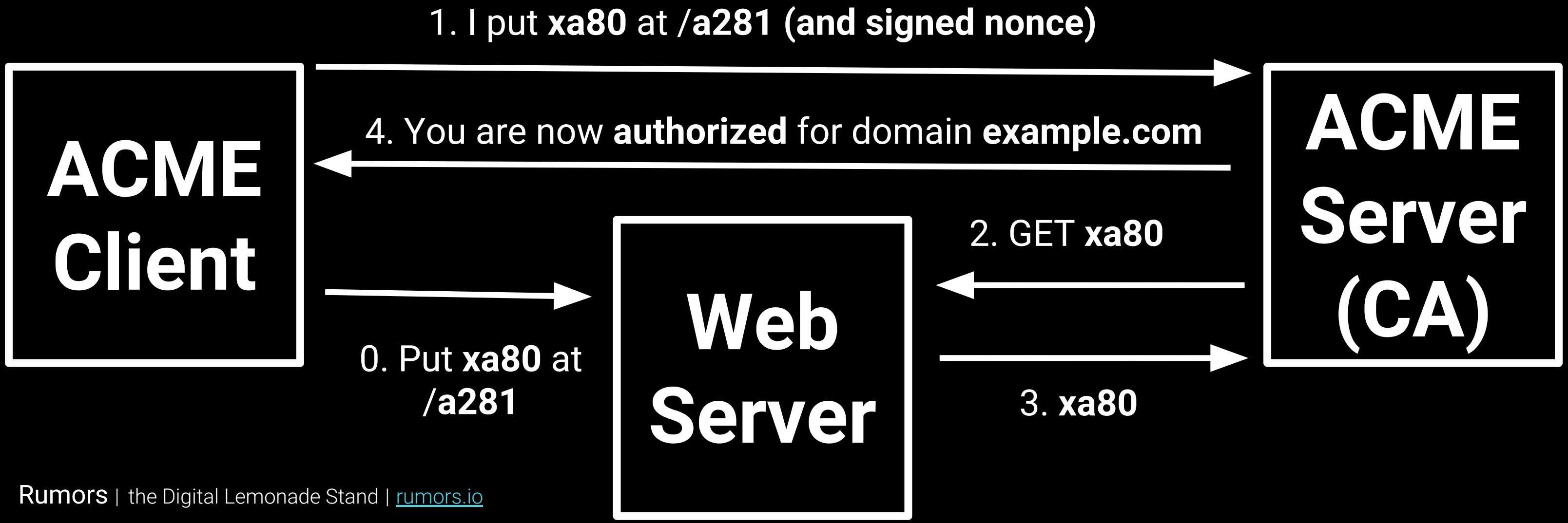
- You tell the CA you’d like to be authorized for a **example.com**
- The CA will give you a challenge to prove you own **example.com**


ACME Client
1. How can I convince you I own example.com ?
2. Put xa80 at http://example.com/a281/ and sign Xhjz9axzFs (nonce)
ACME Server (CA)

Let’s Encrypt - How?

Domain Validation

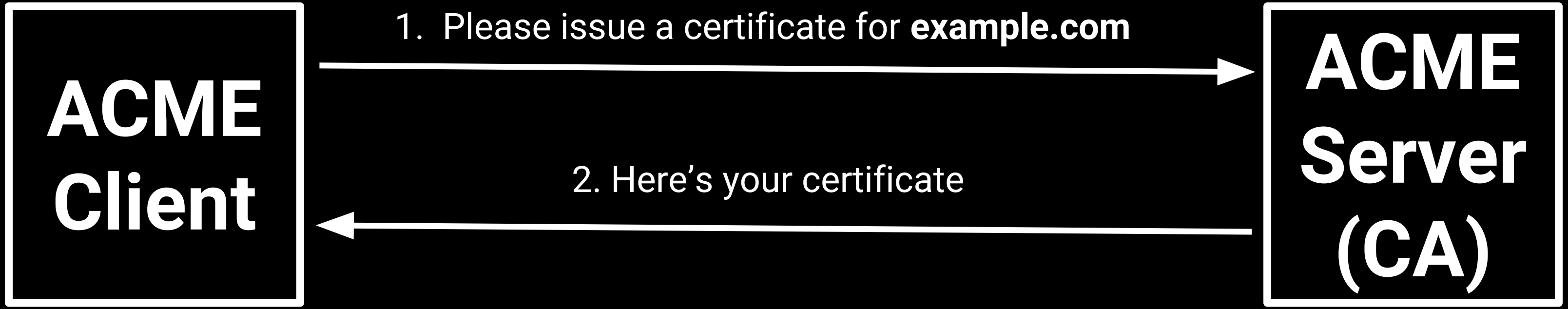
- Once you fulfill the challenge, you let the CA know, and it checks
- If all is well, your account is authorized to manage certs for the domain


1. I put **xa80** at /**a281 (and signed nonce)**
ACME Client
4. You are now **authorized** for domain **example.com**
ACME Server (CA)
2. GET **xa80**
Web Server
0. Put **xa80** at /**a281**
3. **xa80**

Let’s Encrypt - How?

Certificate Issuance

- Client is now authorized for **example.com**
- Client sends a Certificate Signing Request to the Server


ACME Client
1. Please issue a certificate for **example.com**
2. Here’s your certificate
ACME Server (CA)

DEMO

Let's Encrypt

certbot

Thanks!

@sagikedmi

@sagi

sagi.io