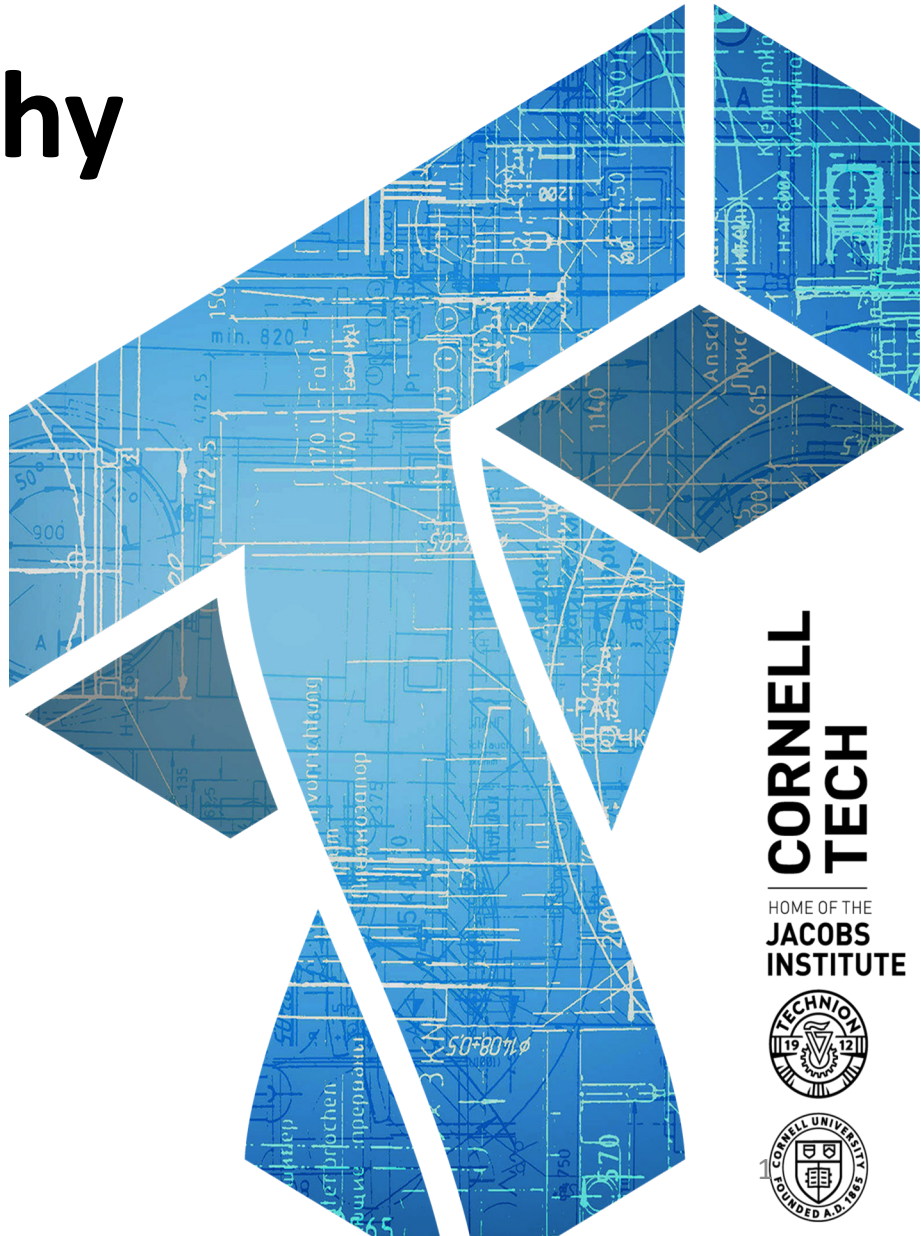
CS 6831:
Designing Secure Cryptography

Tom Ristenpart

Cornell CIS
Computer Science



CORNELL TECH
HOME OF THE
JACOBS
INSTITUTE

Cryptography: “Hidden writing”

- Study and practice of building security protocols that resist adversarial behavior
- Blend of mathematics, engineering, computer science

Crypto is foundational

Failures highlight dependency on good crypto:

- WWII
- Government sabotage
- Playstation 3 crack
- Wireless keys for cars (Tesla)
- Password cracking
- WEP attacks
- Many TLS vulnerabilities
- Stuxnet malware attack
  on Iranian nuclear program
- …

This class

- It is really, really, really hard to get crypto right
- Need principled approaches to help us

design secure cryptography

- Approaches themselves are fascinating

A bit of history

Classical cryptography

1940's Shannon "Communication Theory of Secrecy Systems"

1970's digital asymmetric cryptography & symmetric cryptography
(Diffie-Hellman, RSA, DES)

1980's computational complexity security analyses (Micali-Goldwasser)
symbolic analyses approaches (Dolev-Yao)

1990's concrete security analyses (Bellare-Rogaway)

Gaining confidence in security

Design A

↓ break

Design B

↓ break

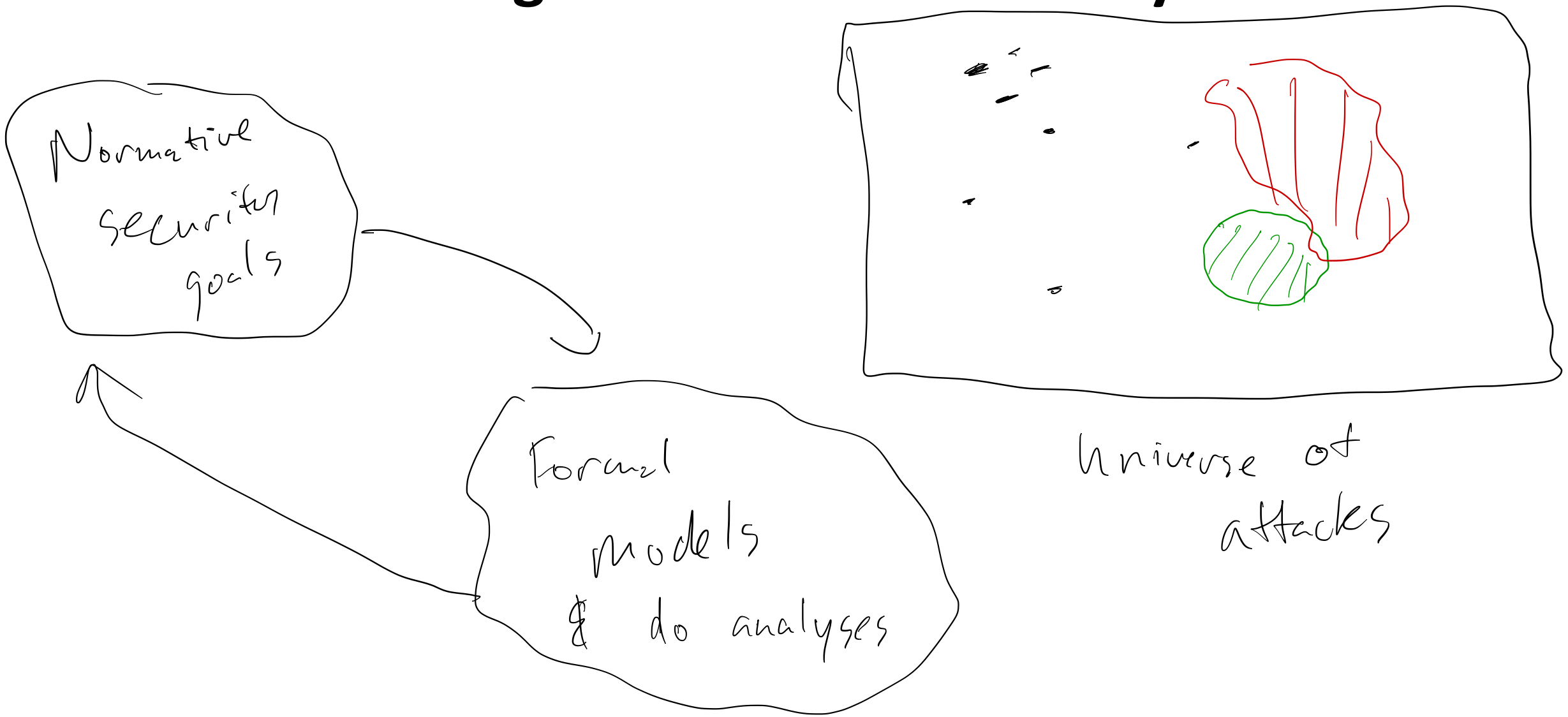
⋮

Provable security

Formal definition of security

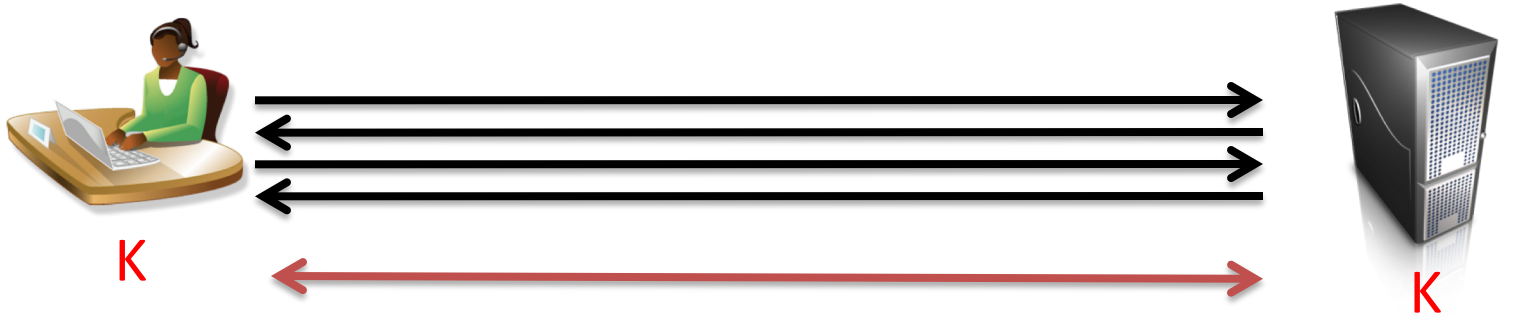
Proof that Design A meets it

Gaining confidence in security



How TLS works (high level view)

https://amazon.com



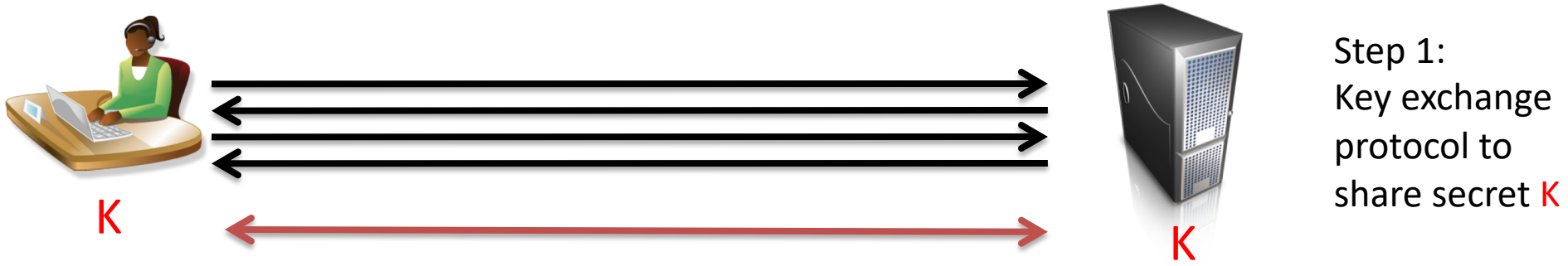
Step 1:
Key exchange protocol to share secret K

Step 2:
Send data via secure channel

Goals of handshake (key exchange protocol):

- Negotiate version
- Negotiate parameters (crypto to use)
- Authenticate server (Is server actually Amazon.com?)
  - Digital signatures and certificates
- Establish shared secret
  - Asymmetric encryption primitives

How TLS works (high level view)



Step 1:
Key exchange protocol to share secret K

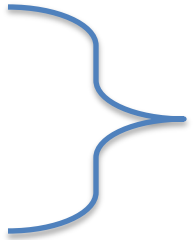
Step 2:
Send data via secure channel

Goals of secure channel (record layer protocol):

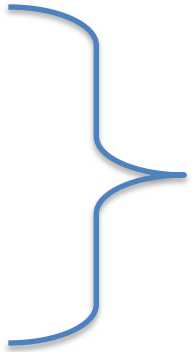
- Confidentiality
  - Only sender/recipient can learn information about plaintext
- Authenticity / integrity
  - Only sender/recipient can generate valid ciphertext

Crypto primitives used in TLS

- Symmetric encryption
- Cryptographic hashing
- Public-key encryption
- Key-exchange
- Digital signatures

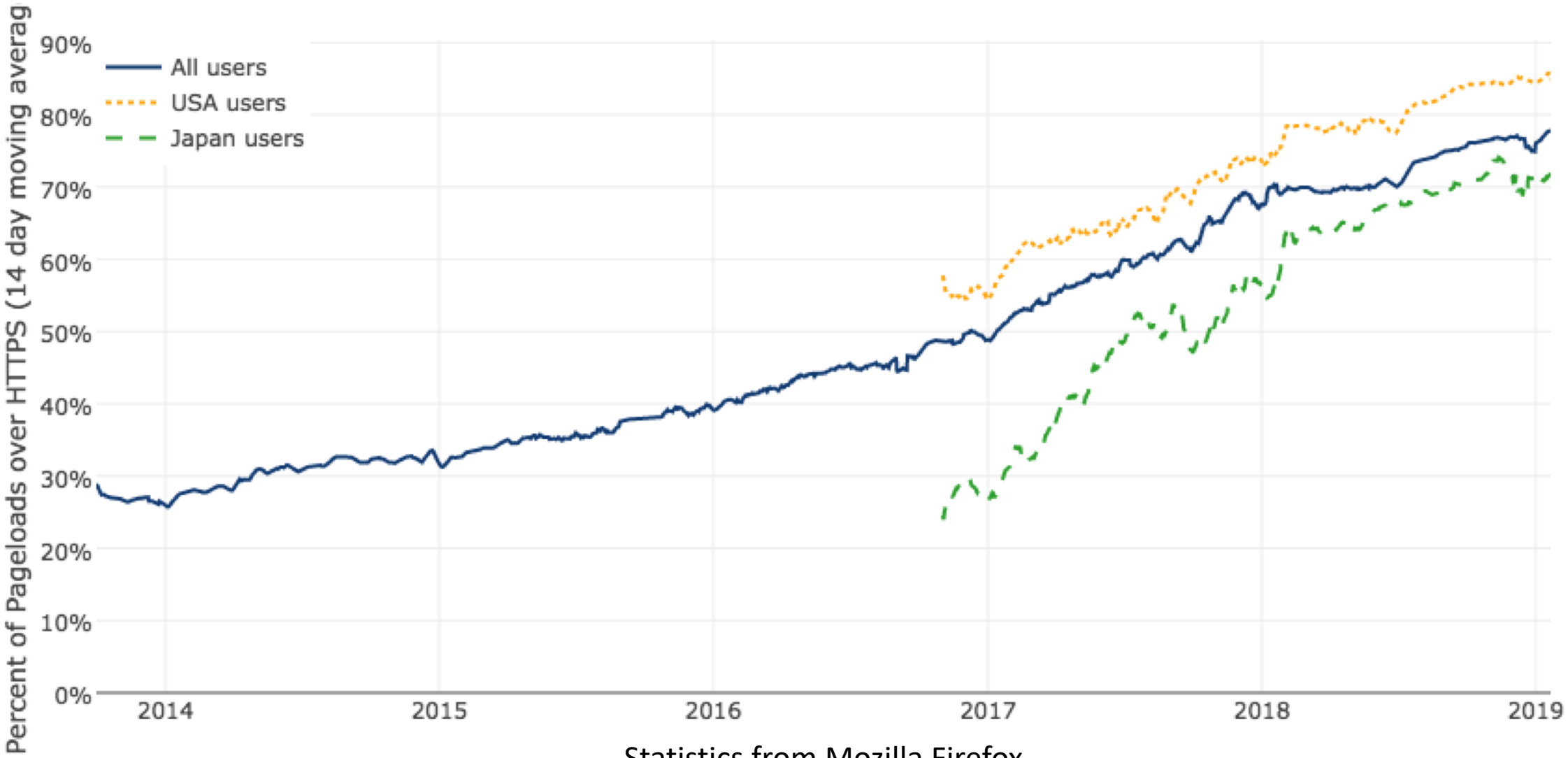


Symmetric crypto



Asymmetric crypto

% of Firefox pageloads using HTTPS



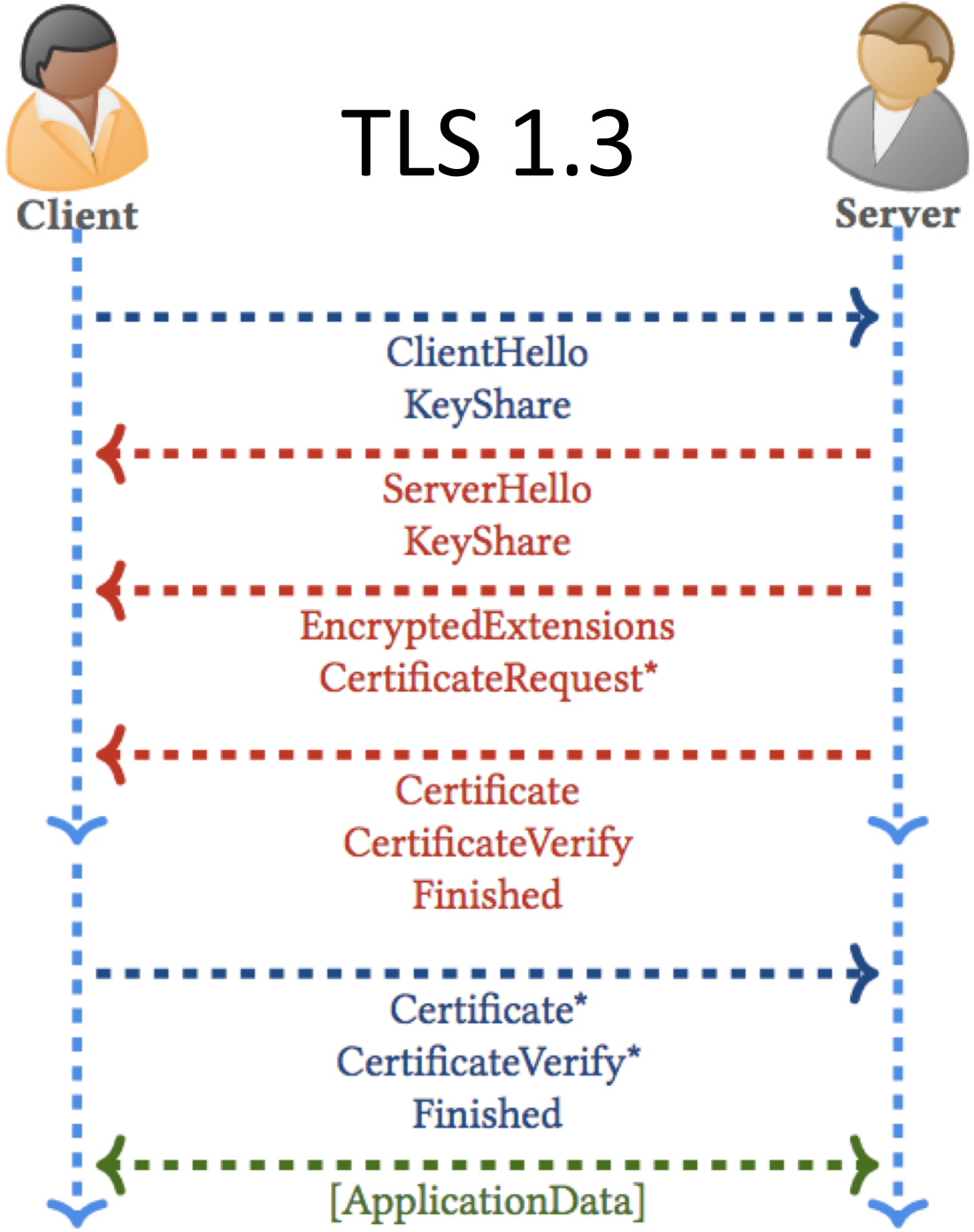
Statistics from Mozilla Firefox
https://letsencrypt.org/stats/

A brief history of TLS

1994-1996	SSL 1.0, 2.0, 3.0
1999	TLS 1.0
2006	TLS 1.1
2008	TLS 1.2
2018	TLS 1.3

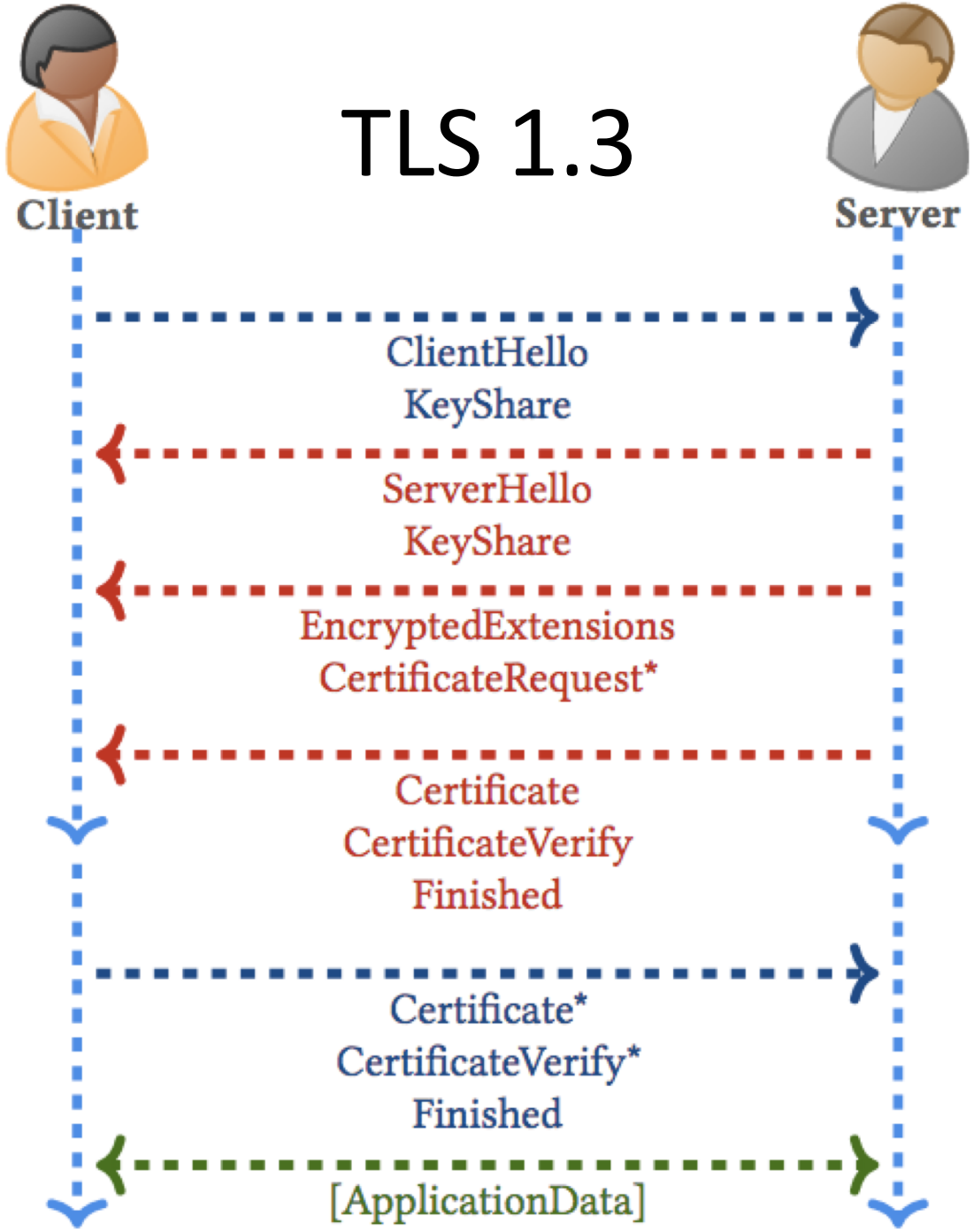
1996 Bleichanbacher RSA attack
2002 Vaudeny CBC padding oracle
2002 Rogaway IV-reuse theory vuln
2011 BEAST attack
2012 CRIME/BREACH compression attack
2013 RC4 attacks
2013 Lucky13 CBC padding oracle
2014 FREAK downgrade attack
2015 Logjam downgrade attack
…

TLS 1.3



From [Cremers et al. 2017]
https://tls13tamarin.github.io/TLS13Tamarin/docs/tls13tamarin-draft21.pdf

TLS 1.3



From [Cremers et al. 2017]
https://tls13tamarin.github.io/TLS13Tamarin/docs/tls13tamarin-draft21.pdf

Process for TLS 1.3

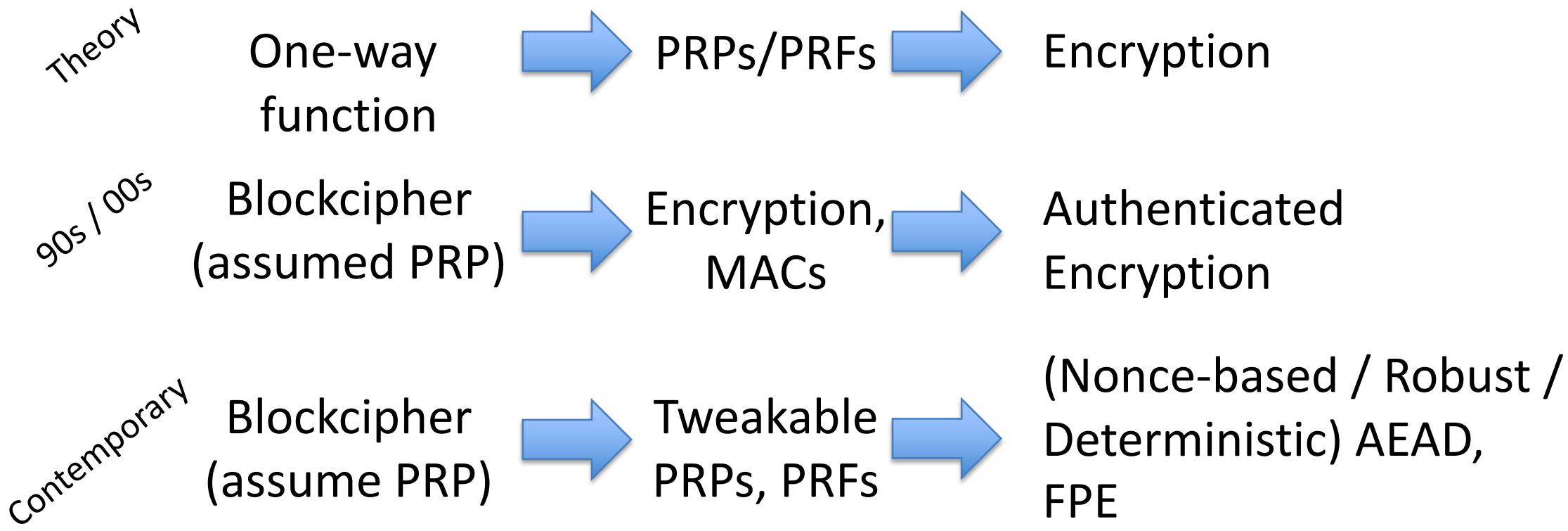
- Unprecedented collaboration
- Almost 100 contributors to RFC 8446

- Measurement
- Symbolic analyses (Dolev-Yao)
- Reductionist security analyses
- Cryptanalytic work

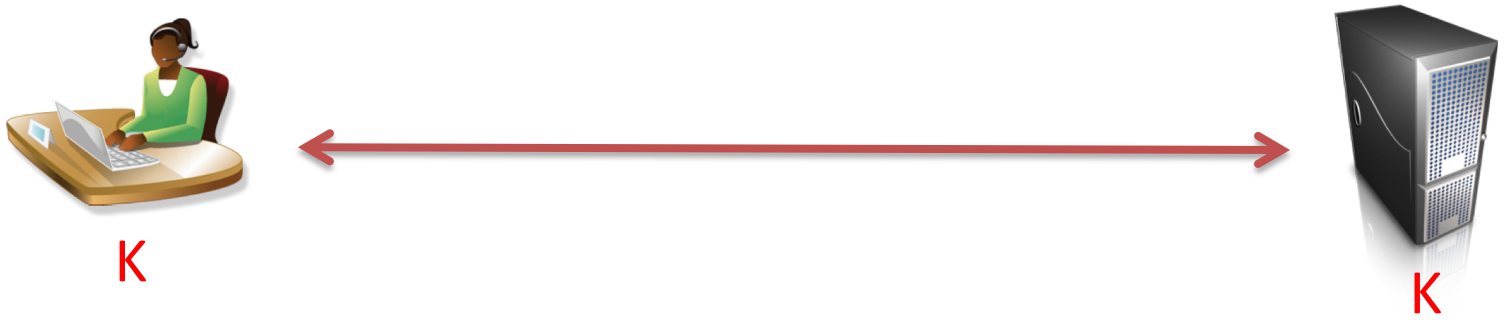
Rough game plan

- Symmetric encryption
- Asymmetric encryption
- More esoteric stuff, time allowing

High level views of building symmetric encryption



Symmetric encryption



- Ciphers, basic definitions and computational viewpoint
- Block ciphers & tweakable block ciphers
  - PRPs / PRFs ; Feistel constructions ; Fast tweakable block ciphers
- Modes of operation, e.g., OCB “core”
- Active security & modern AEAD viewpoints

Ciphers

$\Sigma = (E, D)$ is a pair of algs.

E is enciphering $E: K \times M \to C$

D is deciphering $D: K \times C \to M$

Associated to Σ:

K is key space

M is message space

C is ctxt space

Correctness: $\forall m \in M, \forall k \in K$

$D(k, E(k, m)) = m$

$E_k(m) = E(k, m)$

$D_k(c) = D(k, c)$

Ciphers

Simple OTP cipher $K = M = C = \{0,1\}^n$

$E(K, m) = m \oplus K$

$D(K, C) = C \oplus K$

(1) keys should remain secret

(2) Message confidentiality maintained

Target key recovery security

Identity cipher

$E_K(m) = m$

$D_K(C) = C$

$\underline{\mathcal{A}^{\mathrm{Fn}}}$

$C \leftarrow \mathrm{Fn}(0^n)$

Ret C

$\Leftrightarrow \exists \left[\forall \mathcal{A} \; \mathbf{Adv}^{\mathrm{tkr}}_{\mathcal{E}_{\mathrm{ID}}}(\mathcal{A}) \leq \frac{1}{2^n} \right]$

$\underline{\mathrm{TKR}^{\mathcal{A}}_{\mathcal{E}}}$

$K \xleftarrow{\$} \mathcal{K}$

$K^* \xleftarrow{\$} \mathcal{A}^{\mathrm{Fn}}$

Ret $(K = K^*)$

$\underline{\mathrm{Fn}(M)}$

$C \leftarrow E_K(M)$

Ret C

Chosen plaintext attack

$$\mathbf{Adv}^{\mathrm{tkr}}_{\mathcal{E}}(\mathcal{A}) = \Pr\left[\, \mathrm{TKR}^{\mathcal{A}}_{\mathcal{E}} \Rightarrow \mathsf{true} \,\right]$$

"$K = 1^n$"

$\Pr[K = 1^n] = \frac{1}{2^n}$

Key recovery security

$\underline{\mathrm{KR}_{\mathcal{E}}^{\mathcal{A}}}$

$K \xleftarrow{\$} \mathcal{K}$

$K^* \xleftarrow{\$} \mathcal{A}^{\mathrm{Fn}}$

win $\leftarrow$ true

For $M \in \mathcal{X}$:

    If $E_{K^*}(M) \neq E_K(M)$ then

        win $\leftarrow$ false

Ret win

$\underline{\mathrm{Fn}(M)}$

$\mathcal{X} \leftarrow \mathcal{X} \cup \{M\}$

$C \leftarrow E_K(M)$

Ret C

$$\mathbf{Adv}_{\mathcal{E}}^{\mathrm{kr}}(\mathcal{A}) = \Pr\left[\, \mathrm{KR}_{E}^{\mathcal{A}} \Rightarrow \mathsf{true} \,\right]$$

Comparing security definitions

We can formally compare security definitions

$$\mathrm{DEF1} \not\Rightarrow \mathrm{DEF2}$$

We can show a ***counter-example***:

- Scheme such that show that no (reasonable) DEF1-adversary gets good advantage
- We give DEF2-adversary that gets good DEF2 advantage

$$\mathrm{DEF1} \Rightarrow \mathrm{DEF2}$$

We can show a ***reduction***:

- Convert DEF2-adversary A into DEF1-adversary B s.t. B's DEF1 advantage upper bounds A's DEF2 advantage

$\underline{\mathrm{TKR}^{\mathcal{A}}_{\mathcal{E}}}$

$K \leftarrow\!\!\$\ \mathcal{K}$

$K^* \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{Fn}}$

Ret $(K = K^*)$

$\underline{\mathrm{Fn}(M)}$

$C \leftarrow E_K(M)$

Ret C

$\underline{\mathrm{KR}^{\mathcal{A}}_{\mathcal{E}}}$

$K \leftarrow\!\!\$\ \mathcal{K}$

$K^* \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{Fn}}$

win $\leftarrow$ true

For $M \in \mathcal{X}$:

    If $E_{K^*}(M) \neq E_K(M)$ then

        win $\leftarrow$ false

Ret win

$\underline{\mathrm{Fn}(M)}$

$\mathcal{X} \leftarrow \mathcal{X} \cup \{M\}$

$C \leftarrow E_K(M)$

Ret C

Theorem 1 *Let $\mathcal{E}$ be a cipher. For any $\mathrm{TKR}_{\mathcal{E}}$-adversary $\mathcal{A}$, we give a $\mathrm{KR}_{\mathcal{E}}$-adversary $\mathcal{B}$ such that* $\mathbf{Adv}^{\mathrm{tkr}}_{\mathcal{E}}(\mathcal{A}) = \mathbf{Adv}^{\mathrm{kr}}_{\mathcal{E}}(\mathcal{B})$.

Key recovery doesn’t imply message privacy

- Say we have cipher that meets KR definition. Does this seem to imply message privacy?

Towards message privacy definitions

$\underline{\mathrm{otIND}^{\mathcal{A}}_{\mathcal{E}}}$

$K \leftarrow\!\!\$\ \mathcal{K}$

$b \leftarrow\!\!\$\ \{0,1\}$

$b' \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{Fn}}$

$\mathrm{Ret}\ (b = b')$

$\underline{\mathrm{Fn}(M_0, M_1)}$

$C \leftarrow E_K(M_b)$

$\mathrm{Ret}\ C$

$$\mathbf{Adv}^{\text{ot-ind}}_{\mathcal{E}}(\mathcal{A}) = 2 \cdot \Pr\left[\,\mathrm{otIND}^{\mathcal{A}}_{E} \Rightarrow \mathsf{true}\,\right] - 1$$

OTP otIND security

Theorem 1 *Let $\mathcal{E}$ be the OTP cipher. Then for any single-query* $\mathrm{otIND}_{\mathcal{E}}$*-adversary $\mathcal{A}$ it holds that* $\mathbf{Adv}_{E}^{\mathrm{ot\text{-}ind}}(\mathcal{A}) = 0$.

Shannon's theorem

Theorem 1 *Let $\mathcal{E}$ be a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ such that for any* otIND_E*-adversary $\mathcal{A}$ it holds that* $\mathbf{Adv}_E^{\mathrm{ot\text{-}ind}}(\mathcal{A}) = 0$. *Then* $|\mathcal{K}| \geq |\mathcal{M}|$.

Computational security

- Big paradigm shift is to focus on computationally bound adversaries
- Which adversaries that we've seen so far are computationally efficient?
- How do we measure computational costs?
  - Simple solution: assume abstract unit costs of (most) operations
  - Course-grained but useful

Summary

- Even for simple primitives, security models have many subtleties
  - Multiple versions of KR
  - Easy to end up with mathematically fine, but security-wise uninteresting definitions
- Can formally compare definitions
- Can informally explore definitions
  - Build scheme that formally meets goal, but normatively insecure

Logistics

- https://github.com/cornelltech/CS6831-Spring2019
- Grading:
  - Participation (20%)
  - Scribe notes (40%)
  - Project (40%)
- Class will involve some paper-reading, and you will get out of it what you put into it in terms of content
- I will set scribing schedule and milestones for project shortly, announce end of the week
  - Please let me know if you (not) enrolling ASAP