650.445: Practical Cryptographic Systems

Protocols

Instructor: Matthew Green

Housekeeping: A2

- Extension to Sunday night (11:59pm)
- Make sure you're using the most recent client
- What have we learned about cryptographic libraries?

Reading and Midterm

- 26th of October
- Do the reading, please
  - Anderson, Physical Tamper Resistance
  - Gutmann, Lessons Learned
  - Adam Langley, GotoFail
  - Cache timing attacks on AES
  - Kocher: Timing attacks
  - Bardou et al.: Efficient Padding Oracle Attacks

Reading and Midterm

- 26th of October [will not cover Bitcoin]
- Do the reading, please
  - Anderson, Physical Tamper Resistance
  - Gutmann, Lessons Learned
  - Adam Langley, GotoFail
  - Cache timing attacks on AES
  - Kocher: Timing attacks
  - Bardou et al.: Efficient Padding Oracle
  - Checkoway et al. Systematic analysis
  - ZRTP Protocol

SSL/TLS

- Transport-layer security protocol
  - Often used to secure reliable protocols (TCP)
  - Does not require pre-shared keys
  - Most common usage: https

-E-commerce ($200bn/2008), Banking, etc.

Bank of America

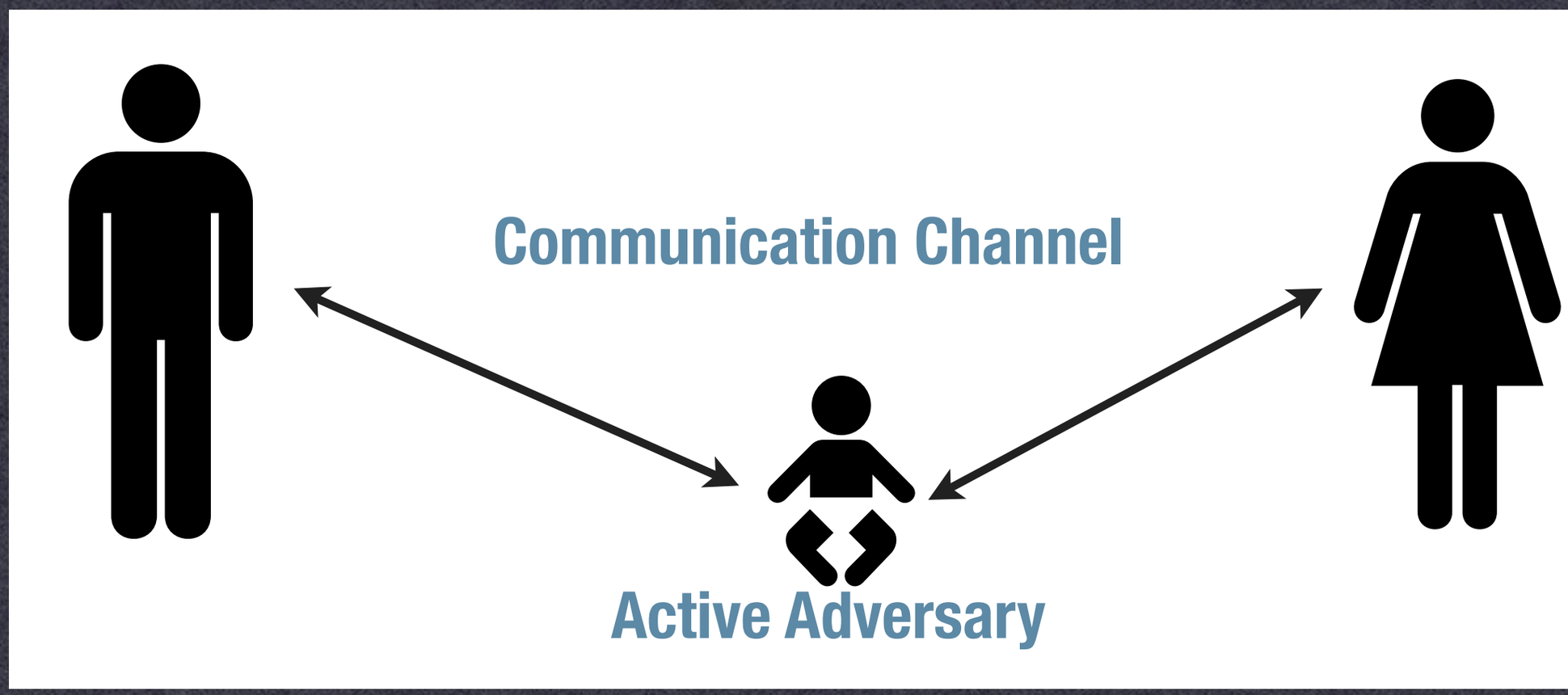
Bank of Opportunity™

https://bofa.com

https

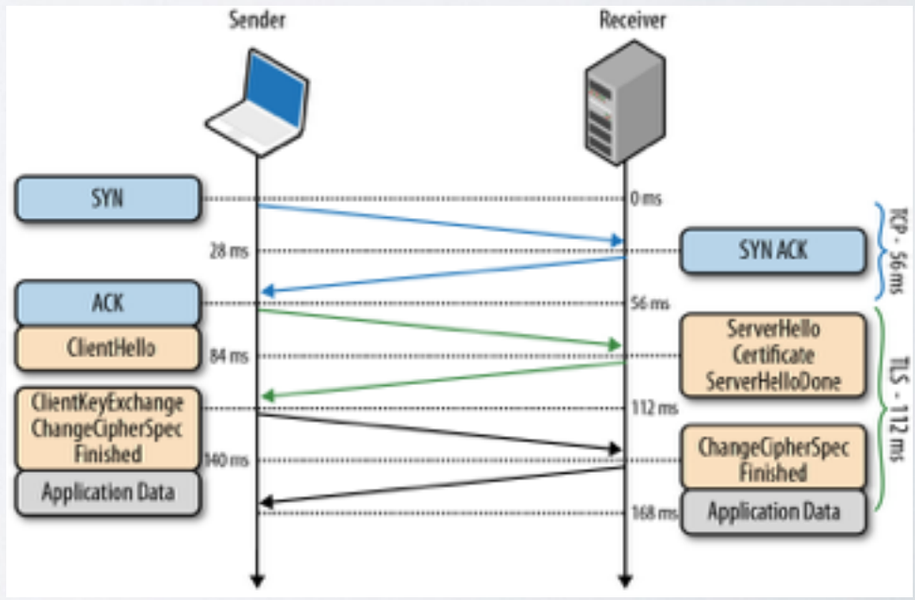
Threat Model

Threat Model



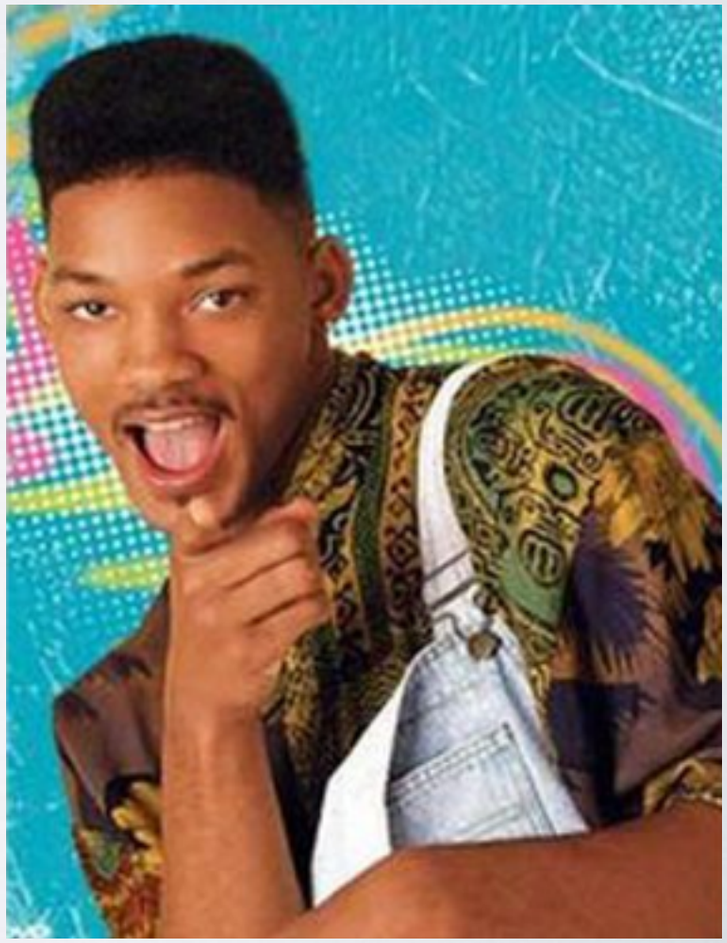
SSL/TLS

- **Most important security protocol on the Internet**
  - Allows secure connections between clients & servers
  - Current version: TLS 1.2
    - (But browsers still support SSL 3, TLS 1.0/1.1) plus 1.3 coming soon!
  - Not just web browsing!



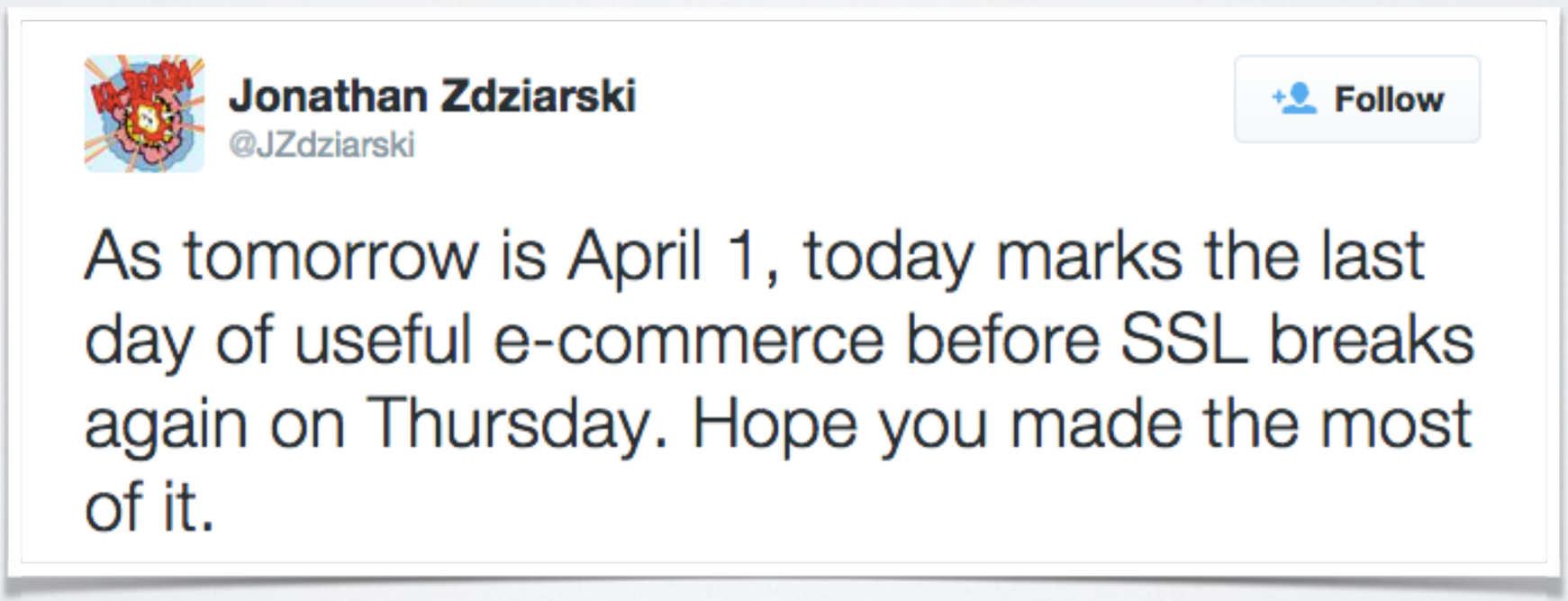
A brief history

- **SSLv1 born at Netscape. Never released. (~1994)**
- **SSLv2 released one year later**
- **SSLv3 (1996)**
- **TLS 1.0 (1998)**
  - Still widely deployed
- **TLS 1.1 (2006)**
- **TLS 1.2 (2008)**



How secure is TLS?

- **Many active attacks and implementation vulnerabilities**
  - Heartbleed, Lucky 13, FREAK, CRIME, BEAST, RC4



Why these problems?

- Many problems result from TLS's use of "*pre-historic cryptography*" (- Eric Rescorla)
  - Export grade encryption
  - RSA-PKCS#1v1.5 encryption padding
  - RC4
  - DH parameter generation
  - Horrifying backwards compatibility requirements

Quite a bit

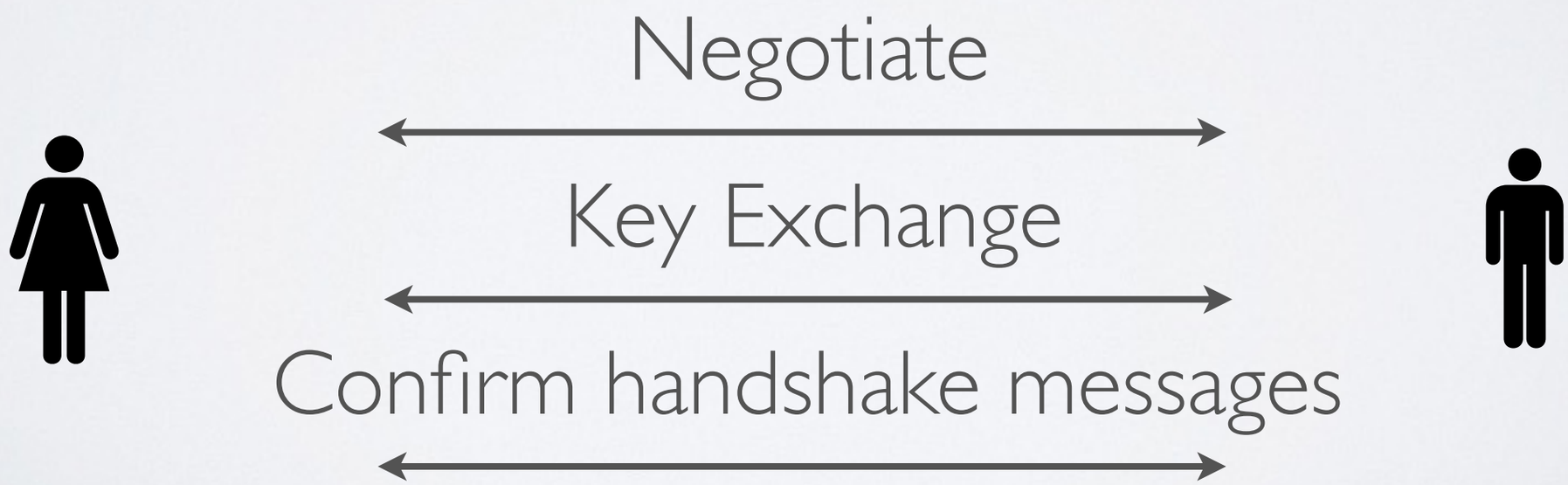
- Many problems result from TLS's use of "*pre-historic cryptography*" (- Eric Rescorla)
  - **Export grade encryption**

1995-~2000 (and onward)

Weakened "ciphersuites" with limited security
(e.g., 512-bit DH/RSA, 40-bit RC4)

TLS Negotiation

Each TLS handshake begins with a cipher suite negotiation that determines which key agreement protocol (etc.) will be used.



SSL/TLS

1. Negotiate peer capabilities
2. Exchange certificates
3. Session key establishment
4. Secure communications
5. Session expiration/rekeying

SSL/TLS

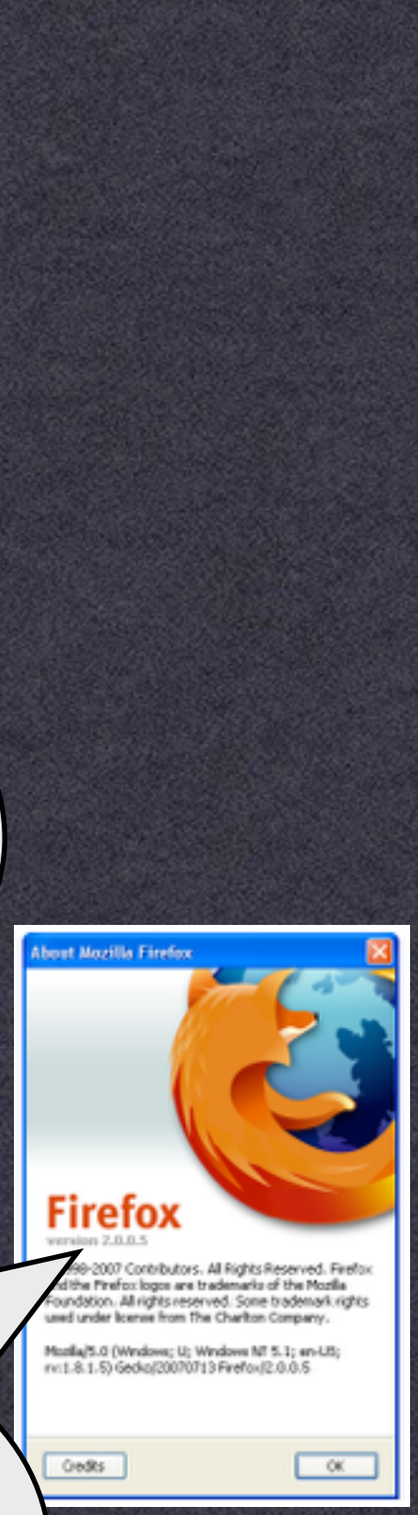
- **Negotiation:**

I choose TLS 1.0
I choose ciphersuite X.

Bank of America
Bank of Opportunity™

1. Negotiate peer capabilities

2. F



I speak SSL 3.0, TLS 1.0.
I support cipher suites X, Y.
I don't have a client cert.

SSL/TLS

- **Certificate Exchange**

$seed_1$

2. Exchange certificates

Here's my cert:
{ *pk*, bofa.com,
Not a CA, Expires 10/1/2008,
signed by $pk_{Verisign}$ } &
$seed_2$

SSL/TLS

- Session key establishment
  - Various options
  - Common approach: RSA based

$C = RSA\text{-}ENC_{pk}(seed_3)$

1. Negotiate peer capabilities
2. Exchange certificates
3. Session key establishment
4. Secure communi
5. Session expirati

$seed_3 = RSA\text{-}DEC(sk, C)$

$k_s = H(seed_1 \,||\, seed_2 \,||\, seed_3)$

$k_s = H(seed_1 \,||\, seed_2 \,||\, seed_3)$

SSL/TLS

- **Secure communication**
  - **In practice, we derive separate MAC & encryption keys**

Communicate under k_s

1. Negotiate peer capabilities
2. Exchange certificates
3. Session key establishment
4. **Secure communications**
5. Session expiration/rekeying

SSL/TLS

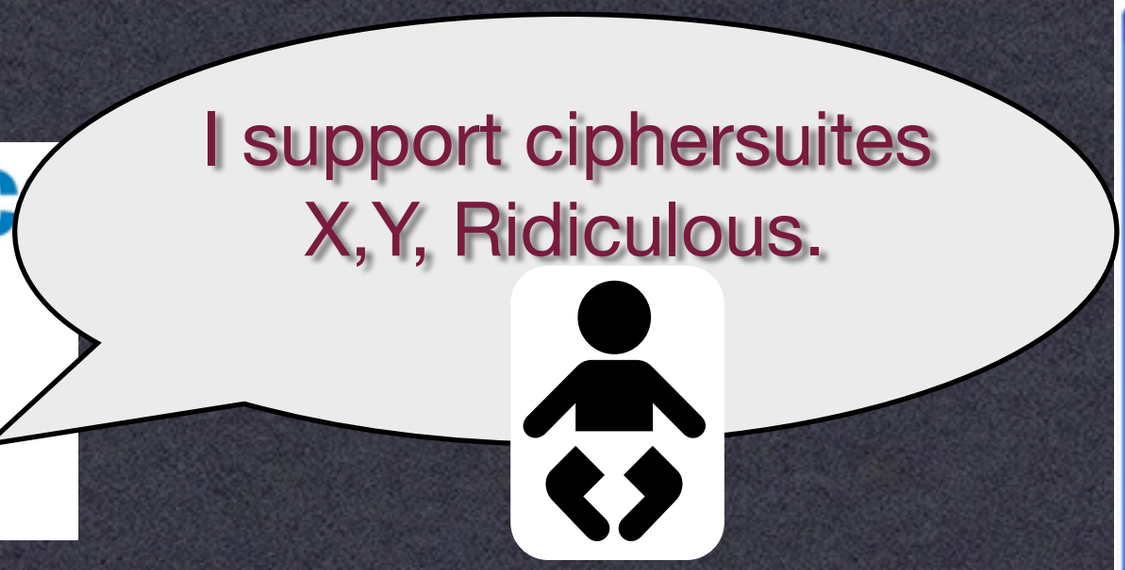
- Key expiration/rekeying
  - Key has a defined lifetime
  - If session drops within that lifetime, we restart:

-This shortcut saves PK operations

1. Negotiate peer capabilities
2. Exchange certificates
3. Session key establishment
4. Secure communications
5. Session expiration/rekeying

Attacks on SSL2

- Many and varied...
- Major vulnerability:
  - Ciphersuite list not authenticated
  - Active attacker could modify the message to specify export-weakened ciphers



SSL3

- All of the problems with SSL2 fixed!
- Well, not quite:
  - Ciphersuite rollback attack (weaker)
  - Key-exchange algorithm rollback
  - Version rollback
  - (Weak) traffic analysis
  - Also, uses some non-standard primitives

Wagner, Schneier: Analysis of the SSL 3.0 Protocol

SSL3 Handshake

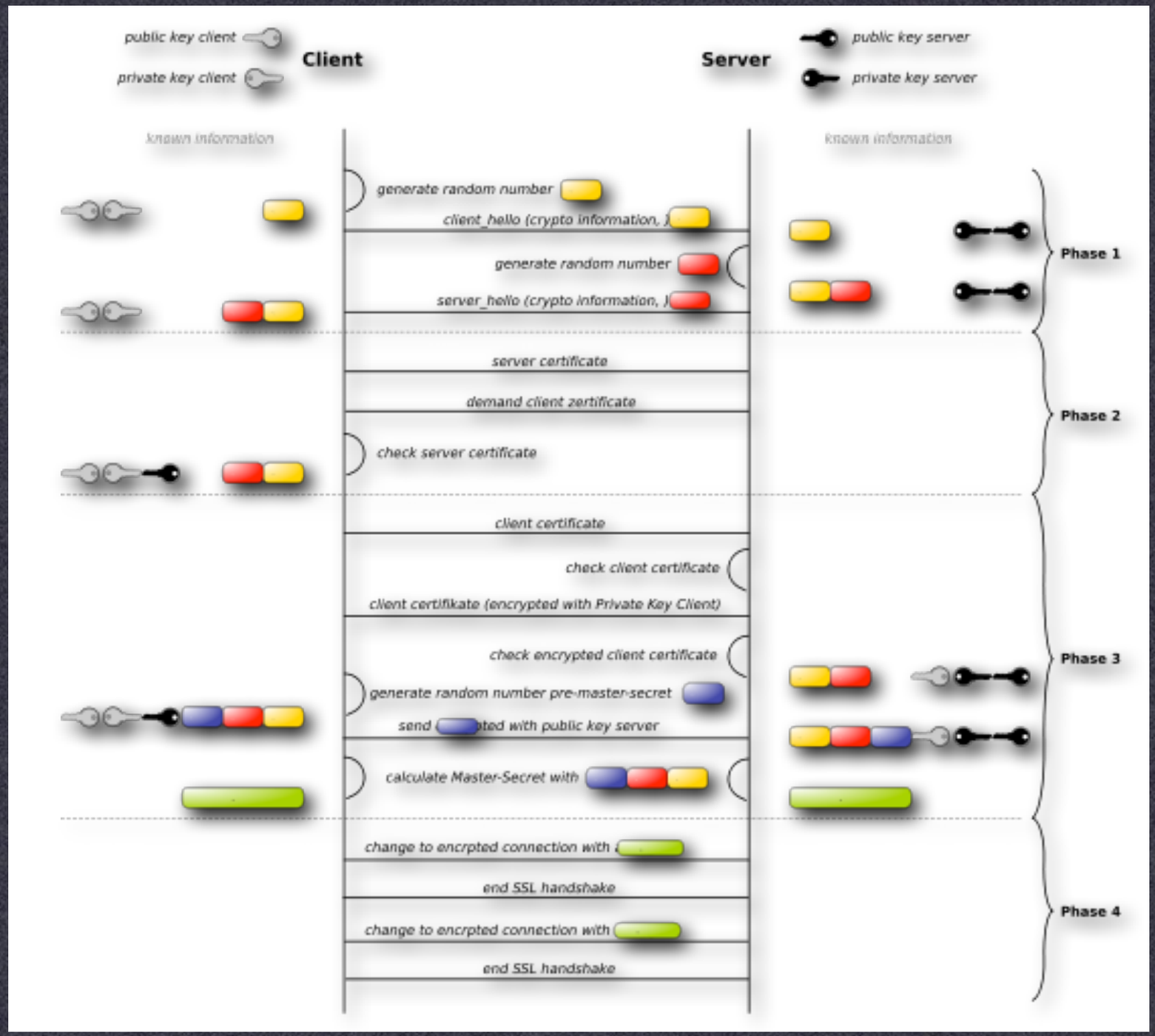


Image from Wikimedia Commons used under a Creative Commons license:
http://commons.wikimedia.org/wiki/File:Ssl_handshake_with_two_way_authentication_with_certificates.svg

CCS Rollback

- Most messages sent during client/server handshake are authenticated
  - Final MAC is sent at finish message
  - However, [change cipher spec] message is not included in the MAC
  - Tells the other party to start using encryption/ authentication
  - Attacker can modify/drop this message!

Wagner, Schneier: Analysis of the SSL 3.0 Protocol

CCS Rollback

- **Normal protocol:**

$$
\begin{array}{lll}
\ldots & & \\
1. & C \rightarrow S: & [\textbf{change cipher spec}] \\
2. & C \rightarrow S: & [\textbf{finished:}]\,\{a\}_k \\
3. & S \rightarrow C: & [\textbf{change cipher spec}] \\
4. & S \rightarrow C: & [\textbf{finished:}]\,\{a\}_k \\
5. & C \rightarrow S: & \{m\}_k \\
\ldots & &
\end{array}
$$

Wagner, Schneier: Analysis of the SSL 3.0 Protocol

CCS Rollback

- **MITM attack:**

$$
\begin{array}{lll}
\ldots & & \\
1. & C \rightarrow M: & [\textbf{change cipher spec}] \\
2. & C \rightarrow M: & [\textbf{finished:}]\ \{a\}_k \\
2'. & M \rightarrow S: & [\textbf{finished:}]\ a \\
3. & S \rightarrow M: & [\textbf{change cipher spec}] \\
4. & S \rightarrow M: & [\textbf{finished:}]\ \{a\}_k \\
4'. & M \rightarrow C: & [\textbf{finished:}]\ a \\
5. & C \rightarrow M: & \{m\}_k \\
5'. & M \rightarrow S: & m \\
\ldots & &
\end{array}
$$

Wagner, Schneier: Analysis of the SSL 3.0 Protocol

Key-Exchange Rollback

- SSL3 standard supports two ephemeral key exchange modes:
  - 1. Server publishes ephemeral RSA parameters (signed under its certified signing key)
  - 2. Server publishes ephemeral DH parameters
  - Client may be able to pick which to use
- Why ephemeral key exchange?
  - Advantages of Diffie-Hellman? RSA?

Wagner, Schneier: Analysis of the SSL 3.0 Protocol

Key Exchange - RSA

Bank of America

Bank of Opportunity™

I'd like to use RSA-ephemeral

RSA Parameters (N,e), signature

Wagner, Schneier: Analysis of the SSL 3.0 Protocol

The Switch

‘FREAK’ flaw undermines security for Apple and Google users, researchers discover

By Craig Timberg March 3, 2015

Most Read

Wagner, Schneier: Analysis of the SSL 3.0 Protocol

Ciphersuite Negotiation

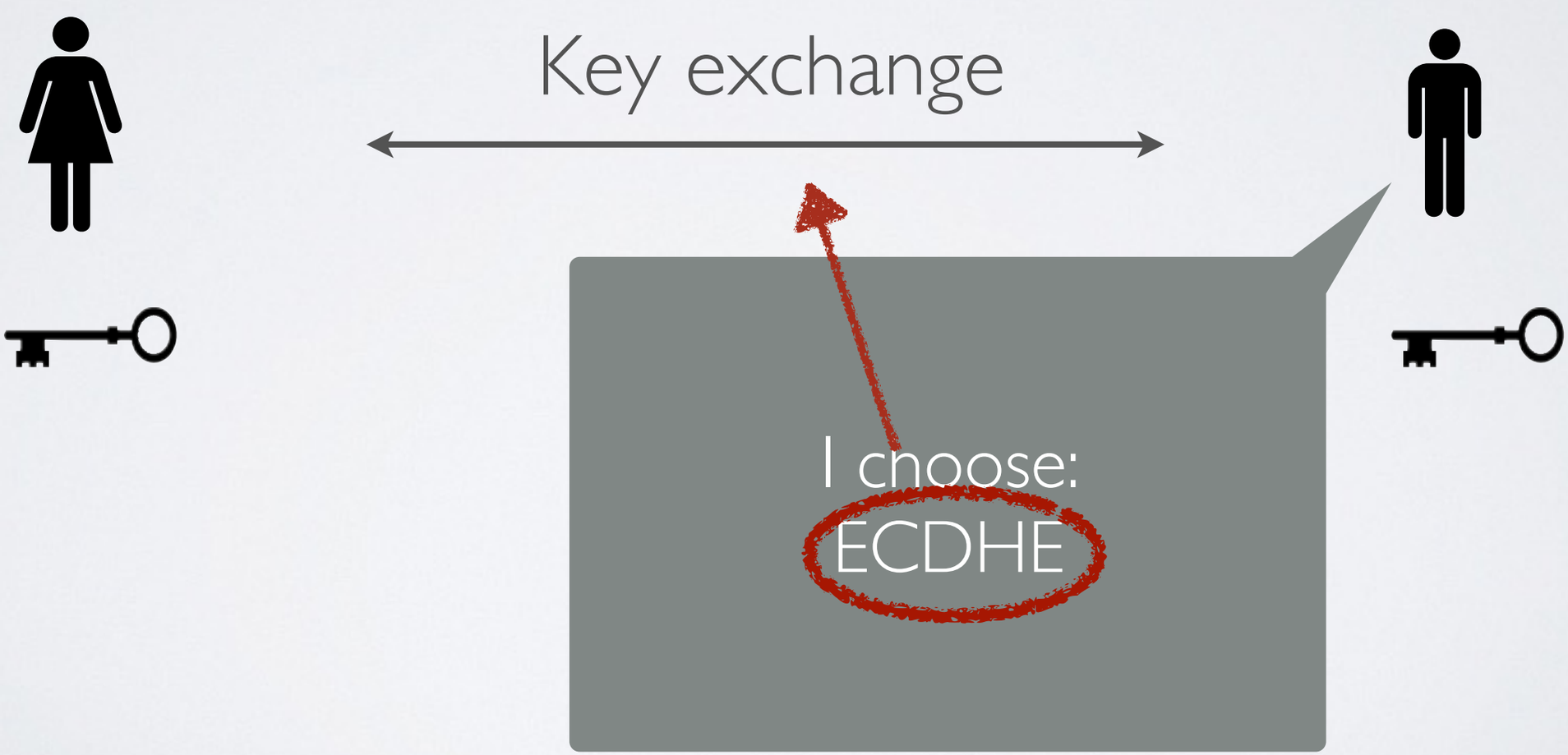
I support:
RSA, DHE, ECDHE,
RSA_EXPORT

Negotiate

I choose:
ECDHE

Ciphersuite Negotiation

I support:
RSA, DHE, ECDHE,
RSA_EXPORT



Ciphersuite Negotiation

I support:
RSA, DHE, ECDHE,
RSA_EXPORT

Confirm handshake messages

I choose:
ECDHE

MITM Negotiation

MITM Negotiation

I support:
~~RSA, DHE, ECDHE,~~
RSA_EXPORT

MITM Negotiation

I support:
~~RSA, DHE, ECDHE,~~
RSA_EXPORT

I choose:
RSA_EXPORT

MITM Negotiation

I support:
~~RSA, DHE, ECDHE,~~
RSA_EXPORT

MITM Negotiation

I support:
~~RSA, DHE, ECDHE,~~
RSA_EXPORT

Attacker can break RSA export key

MITM Negotiation

I support:
~~RSA, DHE, ECDHE,~~
RSA_EXPORT

… and forge confirmation messages

Confirm handshake messages

I choose:
RSA_EXPORT

MITM Negotiation

I support:
~~RSA, DHE, ECDHE,~~
RSA_EXPORT

As of Mar '15,
30+% of TLS hosts supported
export suites!

MITM Negotiation

I support:

Solution:

Modern clients won't offer broken cipher suites like RSA_EXPORT

(unless they're wget or curl!)

I choose:
A_EXPORT

As of Mar '15,
30+% of TLS hosts supported export suites!

Key Exchange - DH

I'd like to use Diffie-Hellman

DH Parameters (p, g, g^a), signature

Wagner, Schneier: Analysis of the SSL 3.0 Protocol

Key Exchange - DH

Security Response

Symantec Official Blog

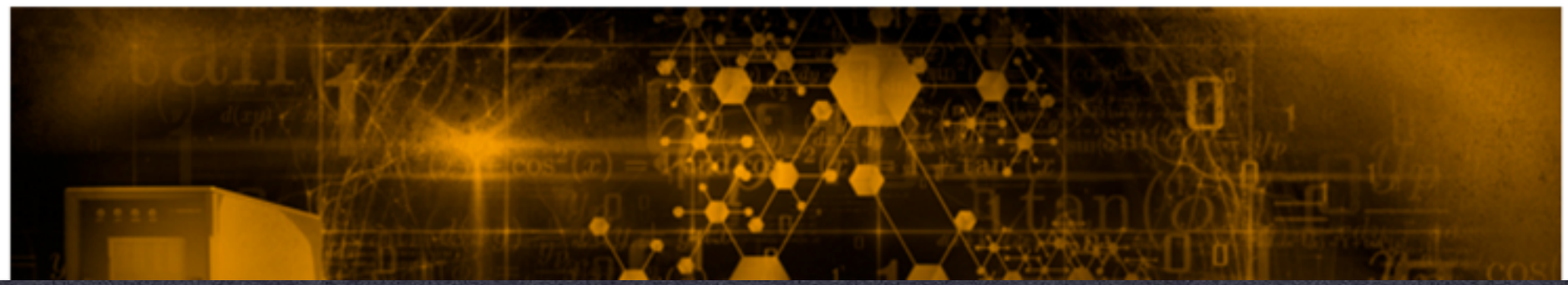
Logjam is latest security flaw to affect secure communication protocols

Similar to FREAK, Logjam is a newly discovered attack that is focused on subverting secure communications on the internet.

By: **Hon Lau** SYMANTEC EMPLOYEE

Created 20 May 2015 | 0 Comments | : 日本語

43 305



Wagner, Schneier: Analysis of the SSL 3.0 Protocol

Key Exchange Rollback

I'd like to use RSA

I'd like to use DH

DH Parameters (p, g, g^a)

RSA Encrypt: $k^g \bmod p$

Normal RSA parameters: (N,e)

I assume *p* is the RSA modulus, and *g* is the RSA exponent. I ignore the extra value.

Since *p* is a prime, we can compute inverses. Recover k.

Version Rollback

- Release of SSL3 didn't make SSL2 browsers go away
  - Servers still accepted SSL2 requests
  - Attacker could modify [client hello] message to specify SSL2
  - Server continues with SSL2 connection, attacker uses SSL2 attacks

Wagner, Schneier: Analysis of the SSL 3.0 Protocol

Version Rollback

- Version rollback is a big problem!
  - SSL, SSH, IPSEC...
  - Example: PPTP

-Can disable encryption, force use of a weaker password authentication protocol

  - Example: L2TP

-Better! But many implementations automatically downgrade to PPTP if L2TP connection fails

Traffic Analysis: SSL3

- Example:
  - First HTTP request typically looks like:

```
GET / HTTP/1.1
Host: cnn.com
User-Agent: Mozilla/5.0 (Macintosh; U; Intel
Mac OS X 10_5_6; en-us) AppleWebKit/525.27.1
 (KHTML, like Gecko) Version/3.2.1 Safari/
525.27.1
```

  - From ciphertext length, we may be able to work out URL information

Traffic Analysis++

- Digression: The case of encrypted VoIP
  - Some VoIP protocols use VBR encoding, size of data packets depends on signal
  - Also include “silence suppression” (VAD)
  - Therefore, total traffic is highly correlated to the contents of the line.

Good news:
Most VoIP implementations don't actually use VBR/ supression

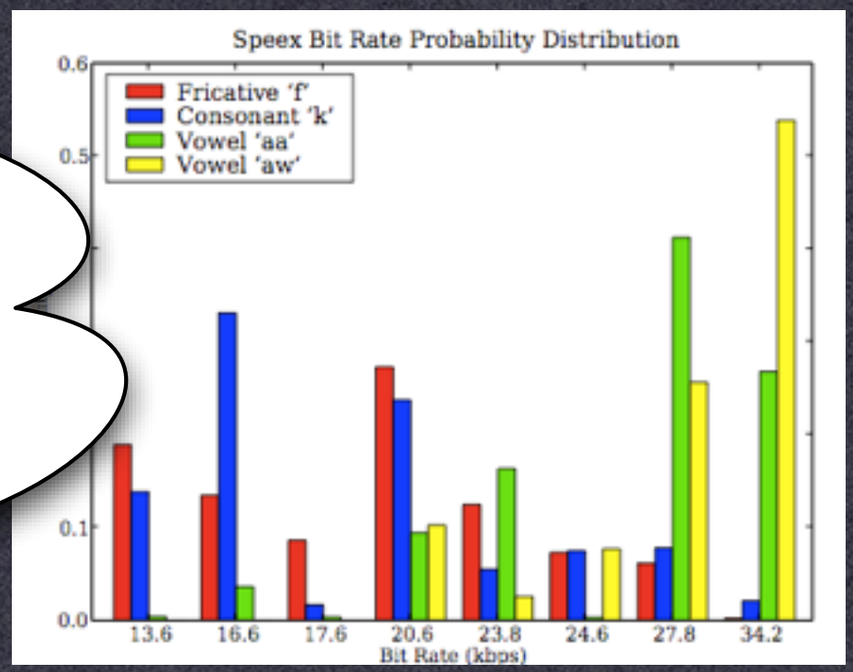


Image from Wright, Ballard, Coull, Monrose, Masson: Spot me if you Can, Uncovering spoken phrases in VoIP conversations

Recent Events

- Black Hat Federal 2009
  - Moxie Marlinspike: SSLStrip

-(now in the wild)

  - Does not break SSL
  - Instead: takes advantage of the way SSL is used

HTTP->HTTPS

- Typical Banking Experience:
  - SSL URLs begin with https://
  - But users rarely type the prefix

User enters: americanexpress.com

GET http://americanexpress.com

REDIRECT https://americanexpress.com

GET https://americanexpress.com

SSL secured web page

User login info

American Express Credit Cards, Travel Services, & Business Credit Cards

https://home.americanexpress.com/home/mt_personal_cm.shtml?

Google

Login page: https

Global Sites | Help | Contact Us | Need He

AMERICAN EXPRESS

WELCOME TO
AMERICAN EXPRESS

PERSONAL CARDS | TRAVEL | SMALL BUSINESS | CORPORATES | MERCHANTS

FIRST-TIME USER?

- Create an Account | Learn More
- Activate a Card

User ID

Remember Me What's this?

Password

Cards -- Check and Pay Bil

LOG IN

Forgot ID or Password?

IMPORTANT ANNOUNCEMENTS

- Delta and AXP Announce Extension of Co-Branded SkyMiles Credit Card

AMERICAN EXPRESS EXCLUSIVE OFFERS

← Car Rental Pro

← Share the Ben

← Only in San Fran

← Travel your wa

← American Expr

← Shop Online w

YOUR CARD BENEFITS

American Express® Gift Card

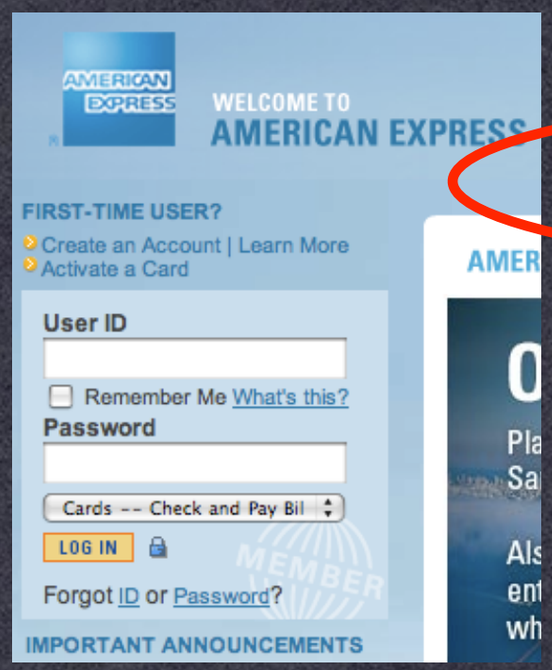
FIND ANOTHER CARD

- Personal
- Small Business
- Corporate
- Gift Cards

Get

HTTP->HTTPS

- If you can intercept the user's connection:
  - Don't redirect, or:
  - Redirect to malicious site, unsecured (http)



User enters: americanexpress.com

GET http://amex.com

REDIRECT http://secure.amex.com

GET https://secure.amex.com

SSL secured web page

User login info

HTTP->HTTPS

- If you can intercept the user's connection:
  - Homograph site: paypal.com (with a capital i), or:
  - Use clever IDN tricks e.g.,

https://www.gmail.com/accounts/ServiceLogin!f.ijjk.cn

HTTP->HTTP->HTTPS

- It can be worse:
  - Some sites give an http page with a form that submits via https

User enters: americanexpress.com

GET http://americanexpress.com

Unsecured http web page

User login info

Wachovia – Personal Finance and Business Financial Services

http://wachovia.com/

Google

Login page: http

Customer Service | Contact Us | Locations

WACHOVIA

Great News
about Free Online Statements—
Now with up to 7 years of
Online Statement history.

See More >

LOGIN

User ID:

Remember my User ID

Password:

(case sensitive)

Service:

Choose a service...

Login

Forgot User ID or Password?

Retirement Plan Participants: Login
Education Loan Customers: Login

Online Security
Wachovia Security Plus℠
Online Services Guarantee

Sign Up for Online Banking
Sign Up | Learn More | Demo

LOCATIONS

ZIP: Find

More Search Options

PERSONAL FINANCE

En español

Online Services
Online Banking with BillPay
Mobile Banking
Online Brokerage
More...

Retirement Planning
Tools & information for Lifetime Retirement Planning

Investing
Accounts & Services
IRAs
More...

Insurance
Life, Auto, Home, Health

Banking
Checking
Savings & CDs
Credit Cards
Check Cards
More...

Lending
Mortgage
Home Equity **New!**
Education Loans
Vehicle Loans

Rates
Mortgage Rates
Home Equity Rates
Credit Card Rates

Payment Challenges?
Explore your loan options

Save up to 30% on TurboTax.
Small Business customers save big on the #1 rated tax software.
Save Now >>

The time is now.
Mortgage rates are at an all-time low.
Refinance Today >>

Search

Search Tips

What to Expect:
Homeowner Affordability & Stability Plan
Learn More >>

WACHOVIA SECURITIES

An industry leader in investment and advisory services for individuals, corporations and institutions.

SMALL BUSINESS

The tools, services, and research to manage your company.
Small Business Login

ONLINE BANKING.
Securely manage your business finances online.
Wachovia Business Online.

CORPORATE & INSTITUTIONAL

Wachovia Securities Corporate and

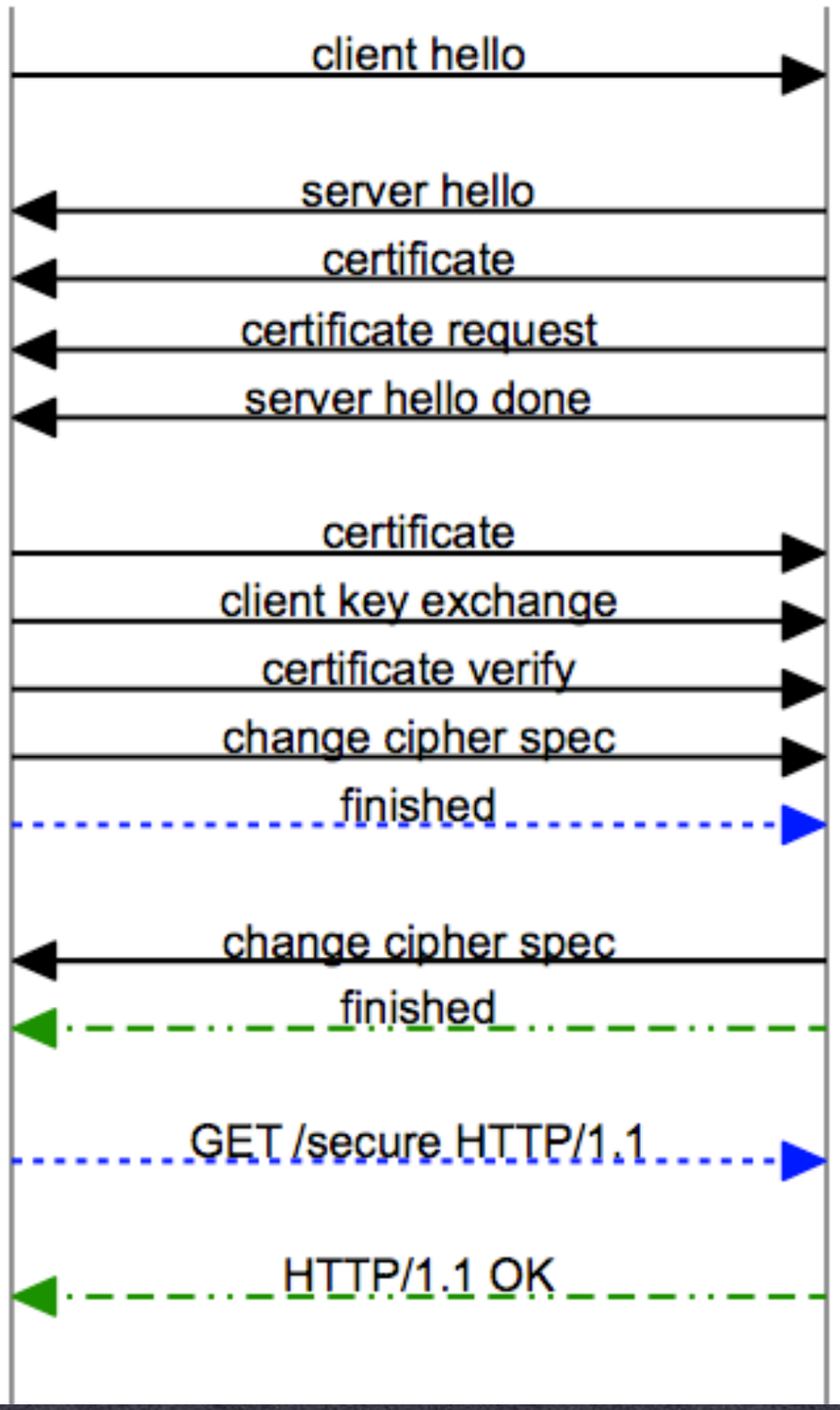
Injecting Prefixes

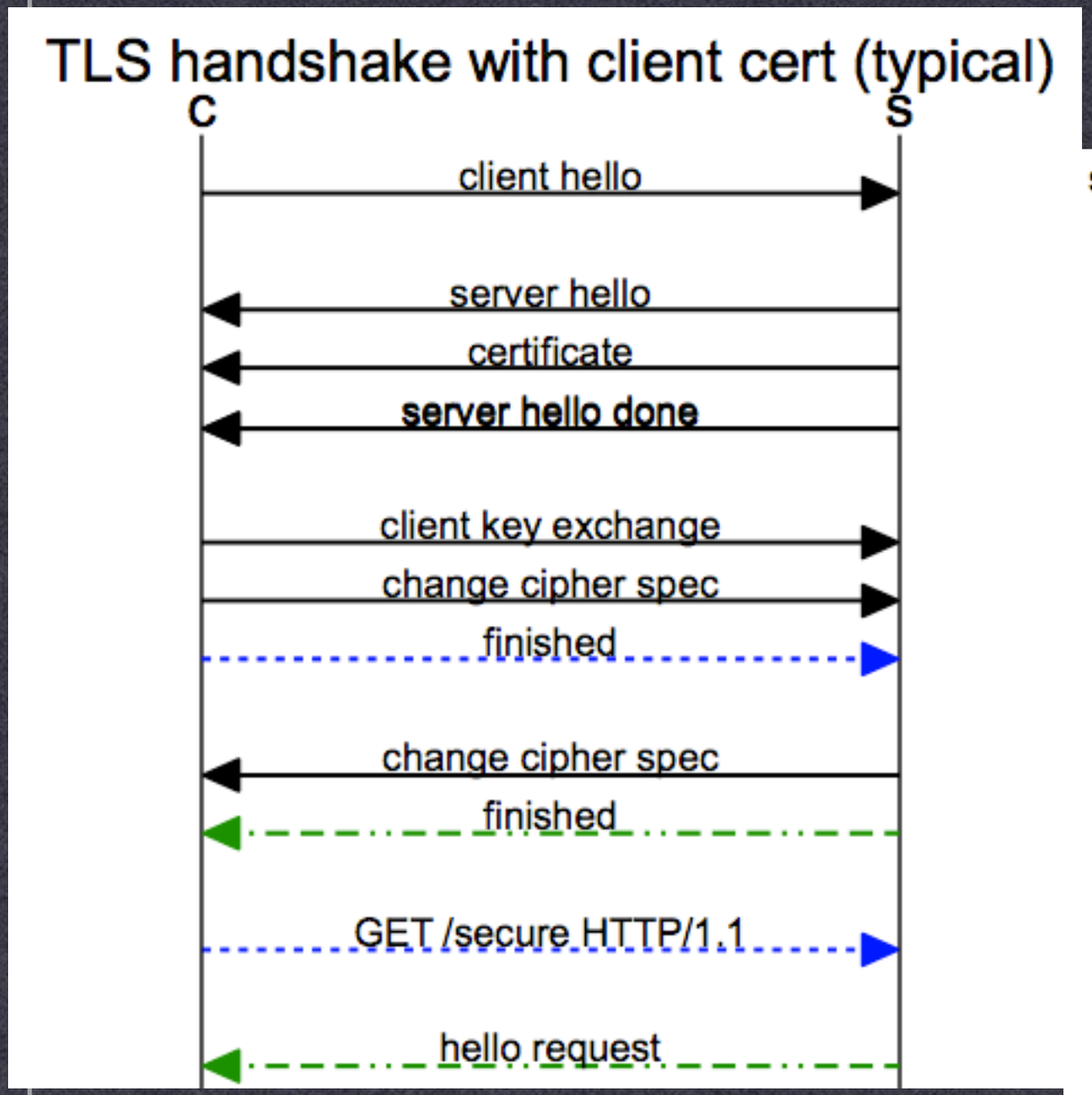
- Ray, Dispensa, 11/2009:
  - Many web servers require client-side auth, but only for <u>certain</u> resources

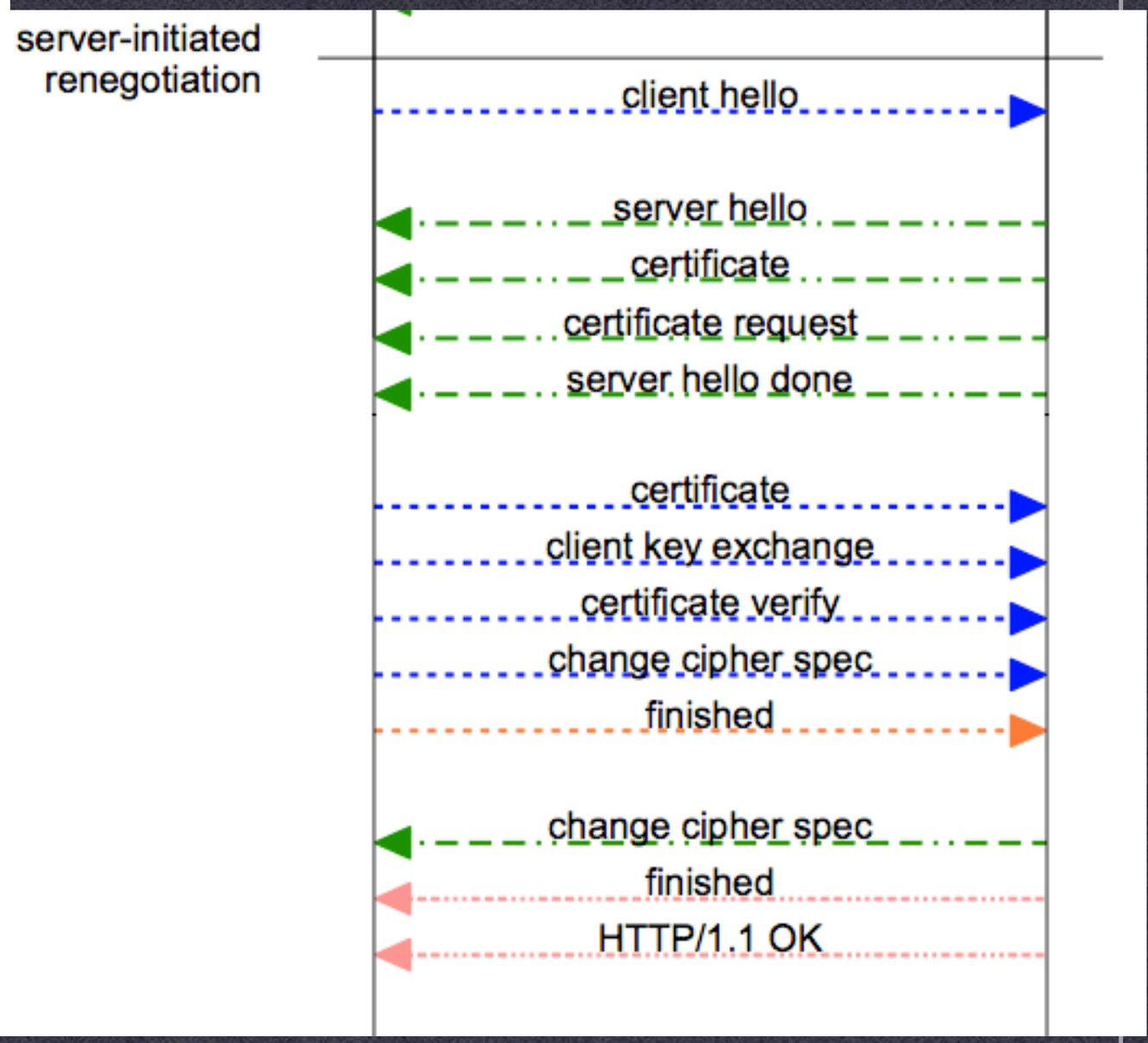
```
GET /highsecurity/index.html HTTP/1.1
Host: example.com
Connection: keep-alive
```

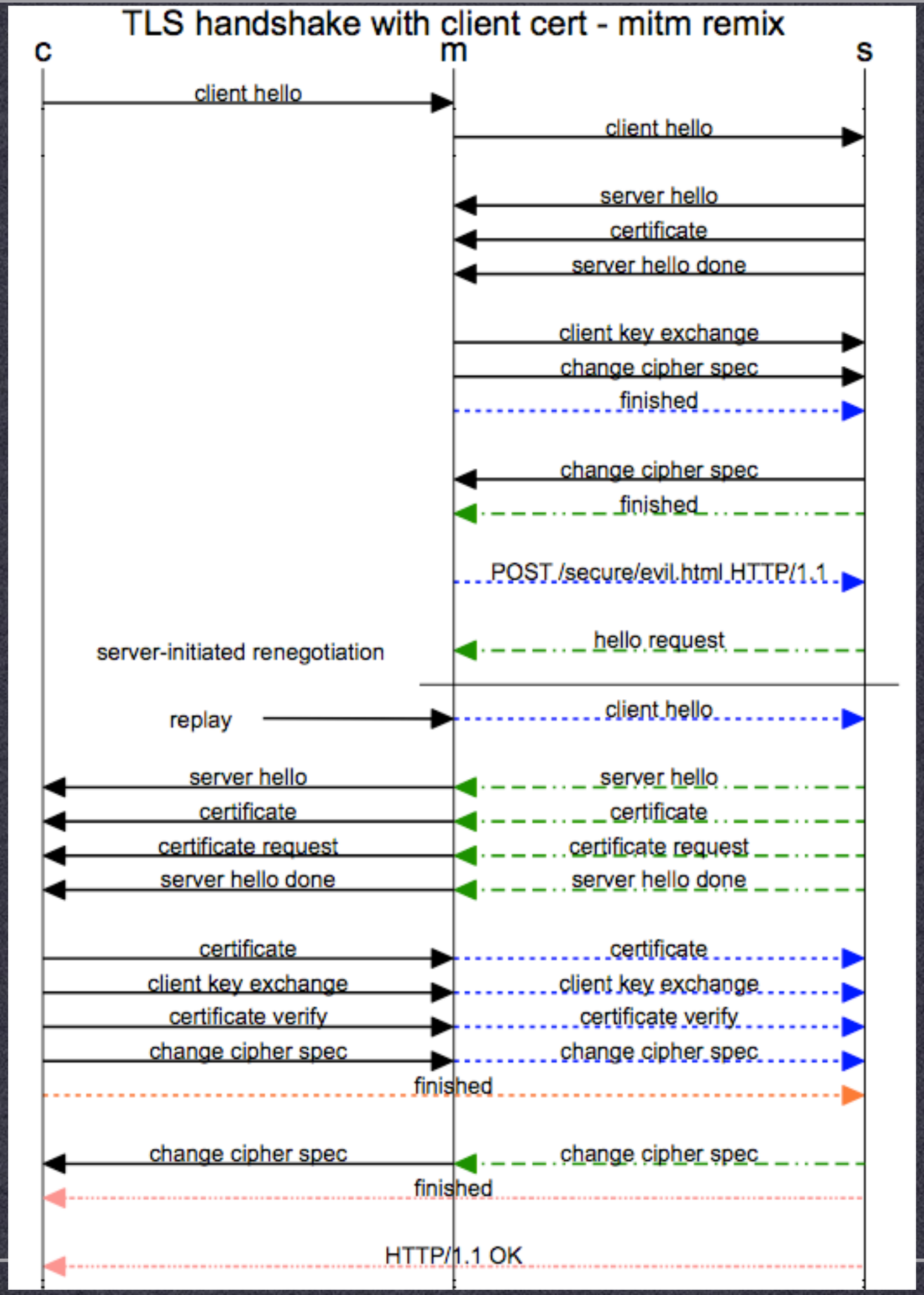
  - This may require an on-the-fly TLS re-negotiation

TLS handshake with client cert (ideal)




TLS handshake with client cert (typical)
C
S
client hello
server hello
certificate
server hello done
client key exchange
change cipher spec
finished
change cipher spec
finished
GET /secure HTTP/1.1
hello request


server-initiated
renegotiation
client hello
server hello
certificate
certificate request
server hello done
certificate
client key exchange
certificate verify
change cipher spec
finished
change cipher spec
finished
HTTP/1.1 OK


TLS handshake with client cert - mitm remix
c
m
s
client hello
client hello
server hello
certificate
server hello done
client key exchange
change cipher spec
finished
change cipher spec
finished
POST /secure/evil.html HTTP/1.1
hello request
server-initiated renegotiation
replay
client hello
server hello
server hello
certificate
certificate
certificate request
certificate request
server hello done
server hello done
certificate
certificate
client key exchange
client key exchange
certificate verify
certificate verify
change cipher spec
change cipher spec
finished
change cipher spec
change cipher spec
finished
HTTP/1.1 OK

DECT

- Digital Enhanced Cordless Telephone protocol
  - European standard, now in US
  - Interoperable devices
  - Connects base station (FT) to handsets (PT)
- Tools:
  - DECT Standard Cipher (DSC)
  - DECT Standard Authentication Algorithm (DSAA)

Lucks, Schuler, Tews, Weinmann, Wenzel: Attacks on the DECT authentication mechanisms

DECT

- **Step 1: Pairing**
  - User enters a 4-digit PIN into handset and base
  - Base generates a 64-bit seed, combined with PIN to generate shared key (UAK)
  - Base and handset conduct challenge/response handshake

Total entropy of UAK:
77.288 bits (64-bit seed + PIN)
Much less if PRNG is bad!

Lucks, Schuler, Tews, Weinmann, Wenzel: Attacks on the DECT authentication mechanisms

DECT

- **Step 2: Authentication**
  - **Two protocols, recommended one:**

AS = A11(UAK, RS)
k, SRES = A12(AS, RAND_F)

RS, RAND_F →

← **SRES**

In common mode, only the handset is authenticated!

AS = A11(UAK, RS)
k, XRES = A12(AS, RAND_F)

SRES == XRES?

Lucks, Schuler, Tews, Weinmann, Wenzel: Attacks on the DECT authentication mechanisms

DECT Attack

- **Step 2: Authentication**
  - **Two protocols, recommended one:**

AS = A11(UAK, RS)
k, SRES = A12(AS, RAND_F)

RS, RAND_F →

← **SRES**

Switch to Encrypted Mode? →

← **Nope.**

AS = A11(UAK, RS)
k, XRES = A12(AS, RAND_F)

SRES == XRES?

Lucks, Schuler, Tews, Weinmann, Wenzel: Attacks on the DECT authentication mechanisms

DECT, other

- A11, A12 built from weak cipher
  - Authors show how this cipher can be inverted using some clever attacks
  - Leaves room for attacks <u>even if</u> protocol bug fixed
  - Eerily reminiscent of GSM...

-Weak protocols

-Weak homebrew ciphers

Lucks, Schuler, Tews, Weinmann, Wenzel: Attacks on the DECT authentication mechanisms

Example: DTCP

- BluRay & HD-DVD Disks
  - Contains “protected” area that can’t be read using normal Drive protocol
  - Embeds secret “Binding Nonce”

DTCP Protocol

- Digital Transmission Content Protection
  - Runs between Drive and Host
  - Encrypts & Authenticates Communications

DTCP

- One layer of protection for HD-DVD/BluRay
  - Encrypts/authenticates content traversing unprotected bus lines

Player ("Sink")

Attacker

Content Source

IEEE 1394 / IP

Image of "DVD Jon" Johansen from http://nanocr.eu/picture.php, used under a Creative Commons license. XBox product photos by Microsoft.

DTCP AKE

- Authenticated Key Exchange
  - EC Diffie-Hellman Protocol
  - Each device has a certificate & secret key
  - Devices also have a certificate revocation list, to prevent communication with hacked devices

DTCP AKE (v1.4)

$$r_A, Cert_A \longleftarrow$$

$$\longrightarrow r_B, Cert_B$$

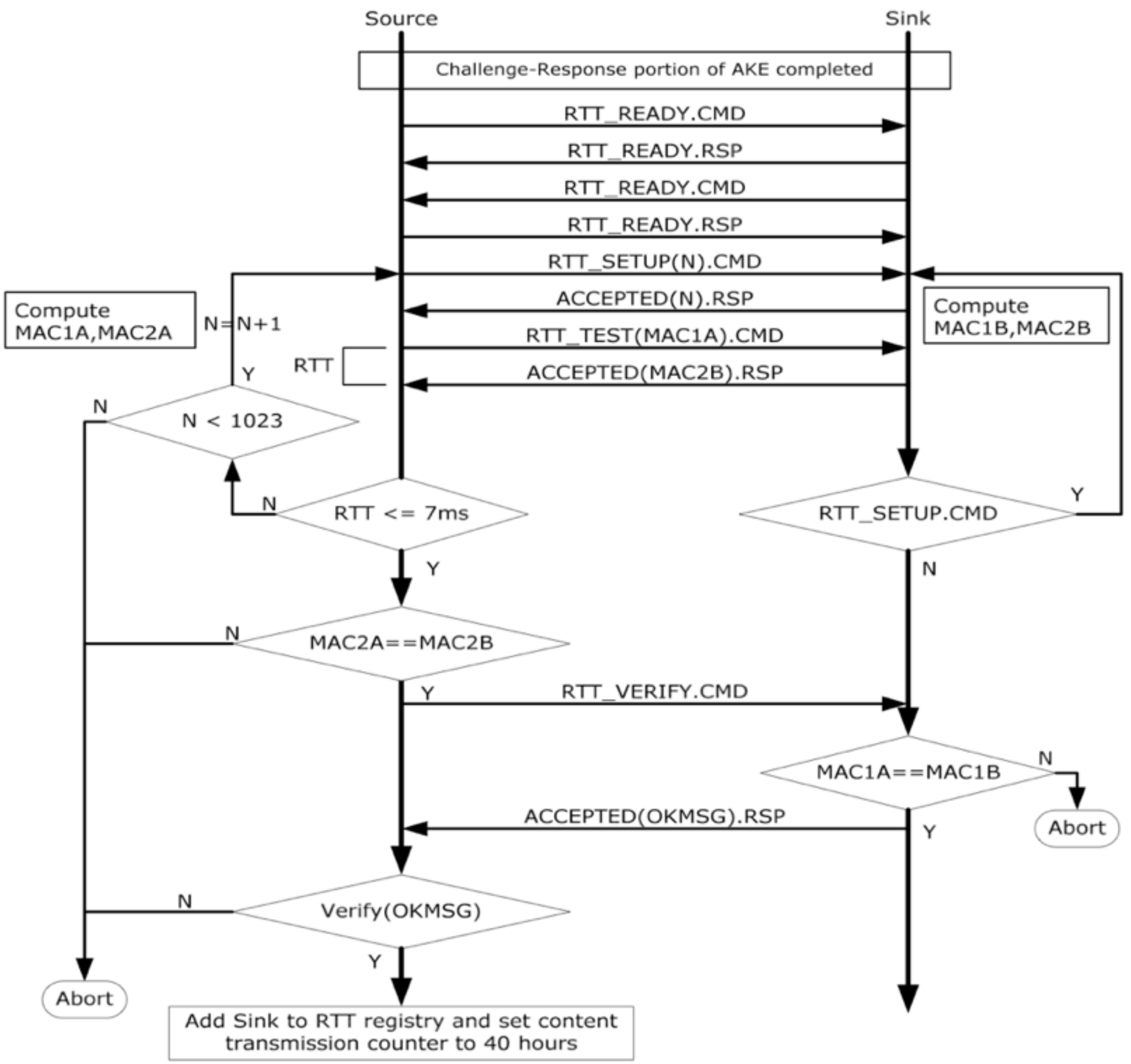
$$g^a, Text_A, Sign_{sk_A}(r_A, g^a, Text_A) \longleftarrow$$

$$\longrightarrow g^b, Text_B, Sign_{sk_B}(r_B, g^b, Text_B)$$

Compute shared key

Verify Shared Key (Localization)

Talk!


Source
Sink
Challenge-Response portion of AKE completed
RTT_READY.CMD
RTT_READY.RSP
RTT_READY.CMD
RTT_READY.RSP
RTT_SETUP(N).CMD
ACCEPTED(N).RSP
RTT_TEST(MAC1A).CMD
ACCEPTED(MAC2B).RSP
Compute MAC1A,MAC2A
N=N+1
Compute MAC1B,MAC2B
RTT
Y
N
N < 1023
N
RTT <= 7ms
Y
RTT_SETUP.CMD
Y
N
N
MAC2A==MAC2B
Y
RTT_VERIFY.CMD
MAC1A==MAC1B
N
ACCEPTED(OKMSG).RSP
Y
Abort
N
Verify(OKMSG)
Y
Abort
Add Sink to RTT registry and set content transmission counter to 40 hours

Other Attacks

- Replay Attacks
  - Attacker replays older messages
  - Can be countered with timestamps, nonces and sequence counters
- Cut & Paste
  - Malleable encryption scheme like CBC
  - Can be countered with MACs
- Reflection
  - If party A sends a message, just bounce it back

Discussion

- We've seen standards with problems
  - Usually the cryptanalysis comes after the standard is released, and products in the field
  - Why?

Next Time

- **Next lecture:**
  - **How do we design secure protocols?**

END

Ciphersuite Rollback

Bank of America

Bank of Opportunity™

[change cipher spec to Auth] ←

[finalize: MAC] ←

Wagner, Schneier: Analysis of the SSL 3.0 Protocol

Ciphersuite Rollback

Wagner, Schneier: Analysis of the SSL 3.0 Protocol

Ciphersuite Rollback

- **Big caveat:**
  - **Only works when client asks for authentication <u>without</u> encryption**

[change [illegible]ec to Auth]

[finalize: <u>Encrypted MAC</u>]

Server thinks encryption is disabled, but gets an encrypted MAC

Wagner, Schneier: Analysis of the SSL 3.0 Protocol