650.445: Practical Cryptographic Systems

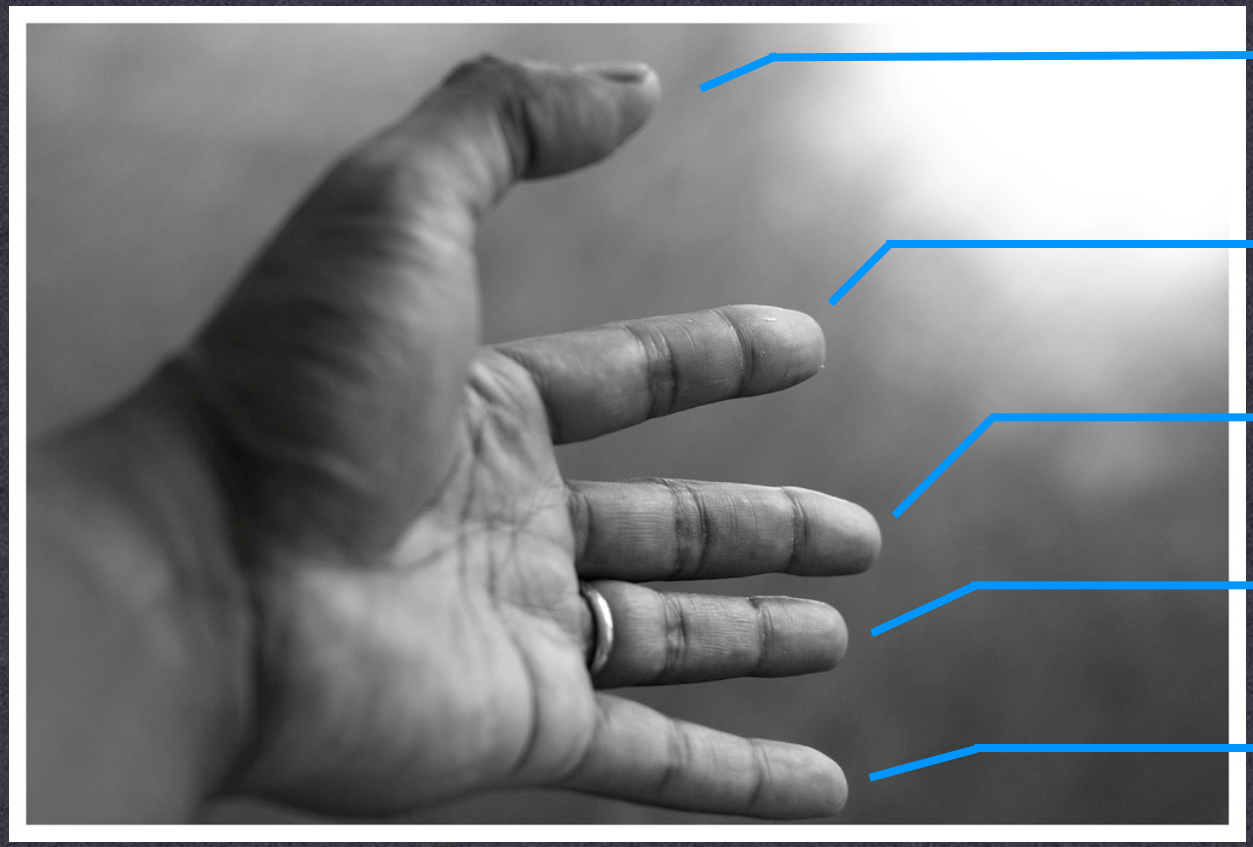
Cryptographic Protocols I

Instructor: Matthew Green

Review

- Housekeeping:
  - Midterm date
  - Assignment extension
  - Piazza: piazza.com/jhu/spring2015/650445600454

Today's class



- Concept
- Primitives
- **Protocols**
- Implementation
- Usage

Truism

- (Distributed) systems almost always use complex protocols:
  - Take a look at any RFC!
- Many use cryptographic protocols:
  - Peer authentication
  - Key establishment
  - Encryption/message authentication
  - Device pairing, etc.

Protocols

- When evaluating a product, this is fertile ground
  - Colossally easy to screw up
- Typical problems:
  - Too many options for implementers
  - Backwards compatibility
  - Assuming components do more than they actually do
  - Assuming implementers will add checks beyond the spec

Case Studies

- Examples:
  - SSL/TLS - securing internet transactions
  - IPSEC - secure IP packets
  - DECT - cordless phones
  - GSM - cellphones

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.

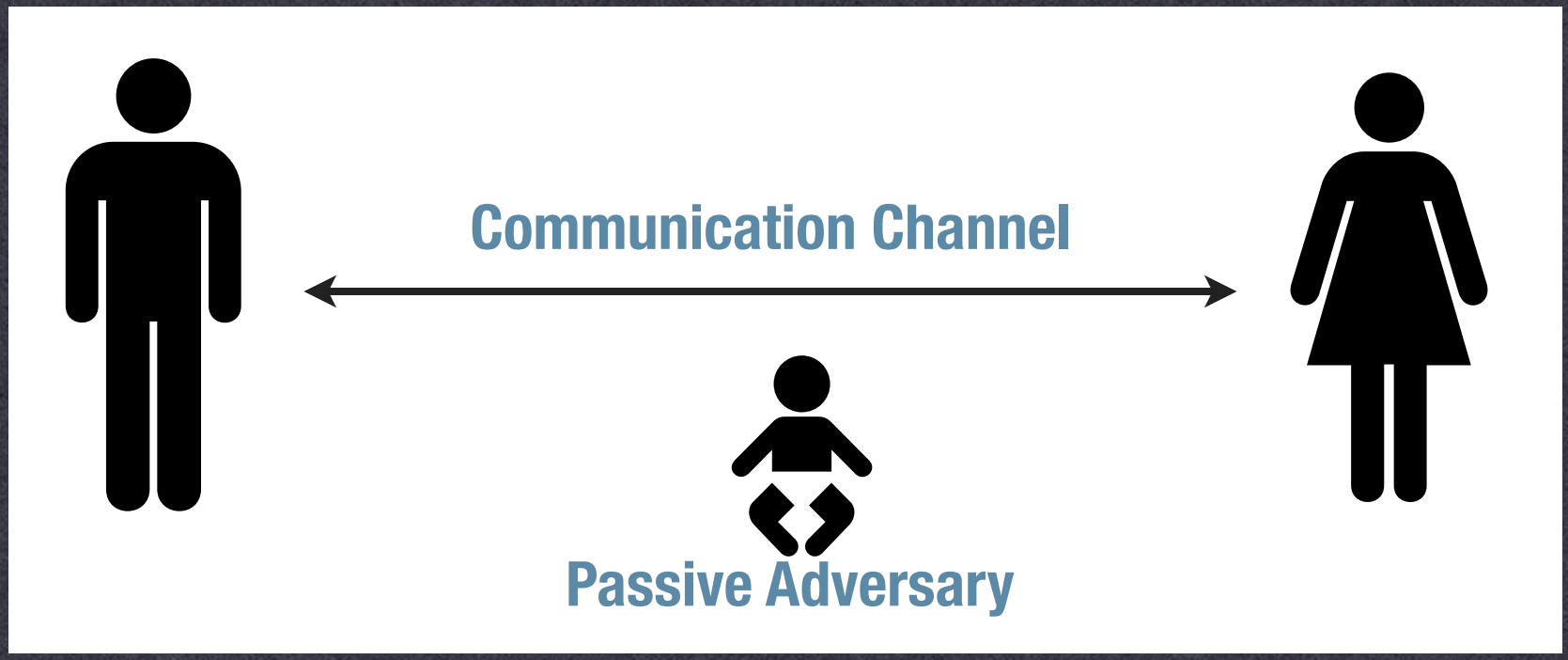
Bank of America

Bank of Opportunity™

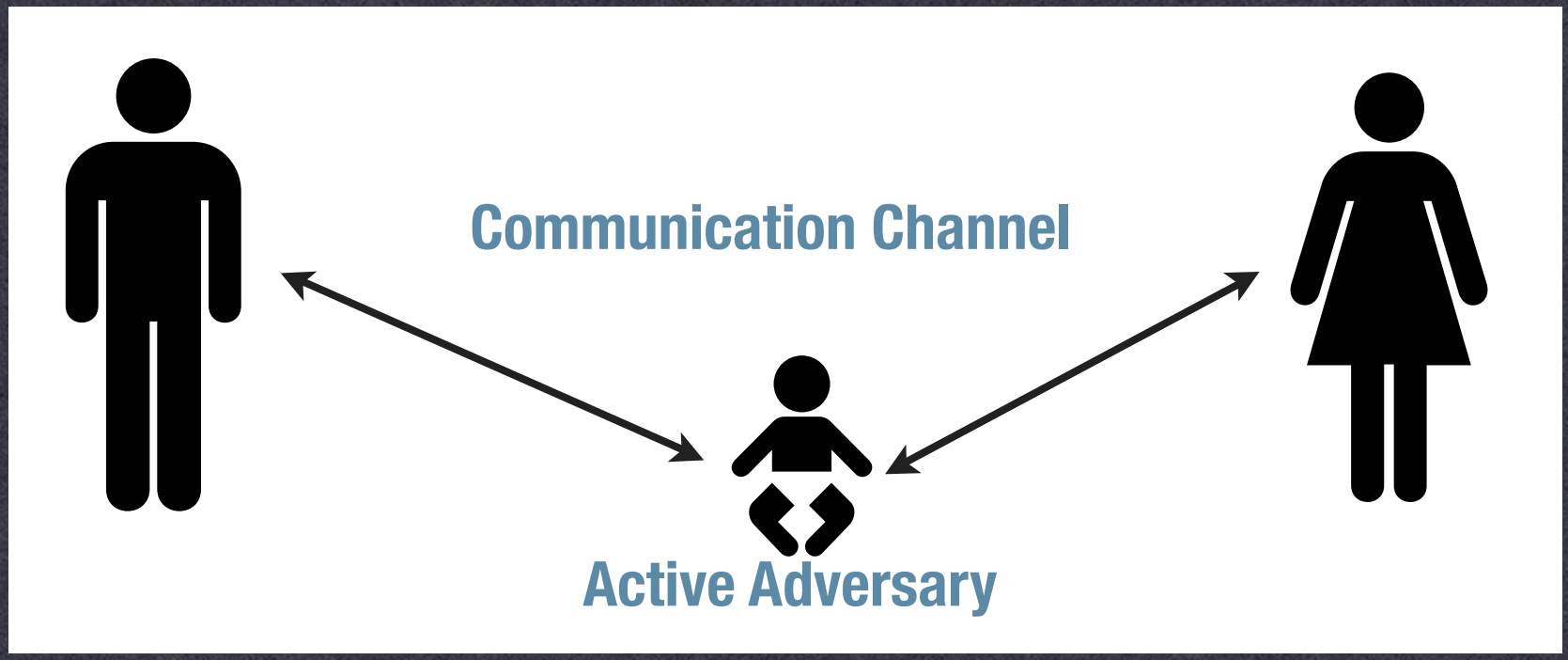
https://bofa.com

https

Threat Model



Threat Model



SSL/TLS

- Version 1.0 (pre-1995)
  - Non-public Netscape protocol
- Version 2.0 (1995)
  - Many flaws: export-weakened keys, rollback
- Version 3.0 (1996)
  - Some flaws (Wagner, Schneier)
- TLS Version 1.0 (1999)
  - Version 1.2 is current standard (as of 8/08)

SSL/TLS

- Extremely flexible protocol
  - Lots of options
  - Backwards compatibility (TLS 1->SSL 3->SSL 2)

Bank of America

Bank of Opportunity™

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

SSL/TLS

- **Negotiation:**

I speak SSL 2.0, 3.0.
I support ciphersuites X,Y.
I don't require a client cert.

Bank of America

Bank of Opportunity™

1. Negotiate peer capabilities

2. F

I speak SSL 3.0.
I support ciphersuite X.
I don't have a client cert.

SSL/TLS

- **Certificate Exchange**

Bank of America

Bank of Opportunity™

$seed_1$

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**

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

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

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

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

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

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

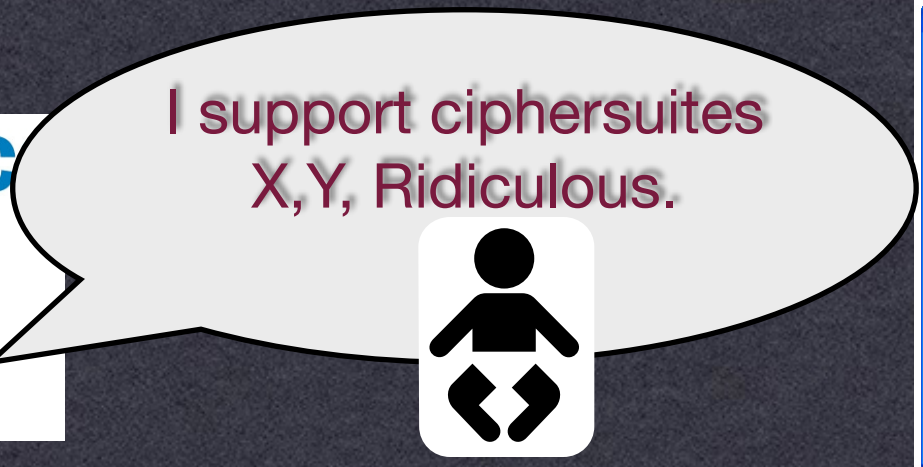
SSL/TLS

- 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

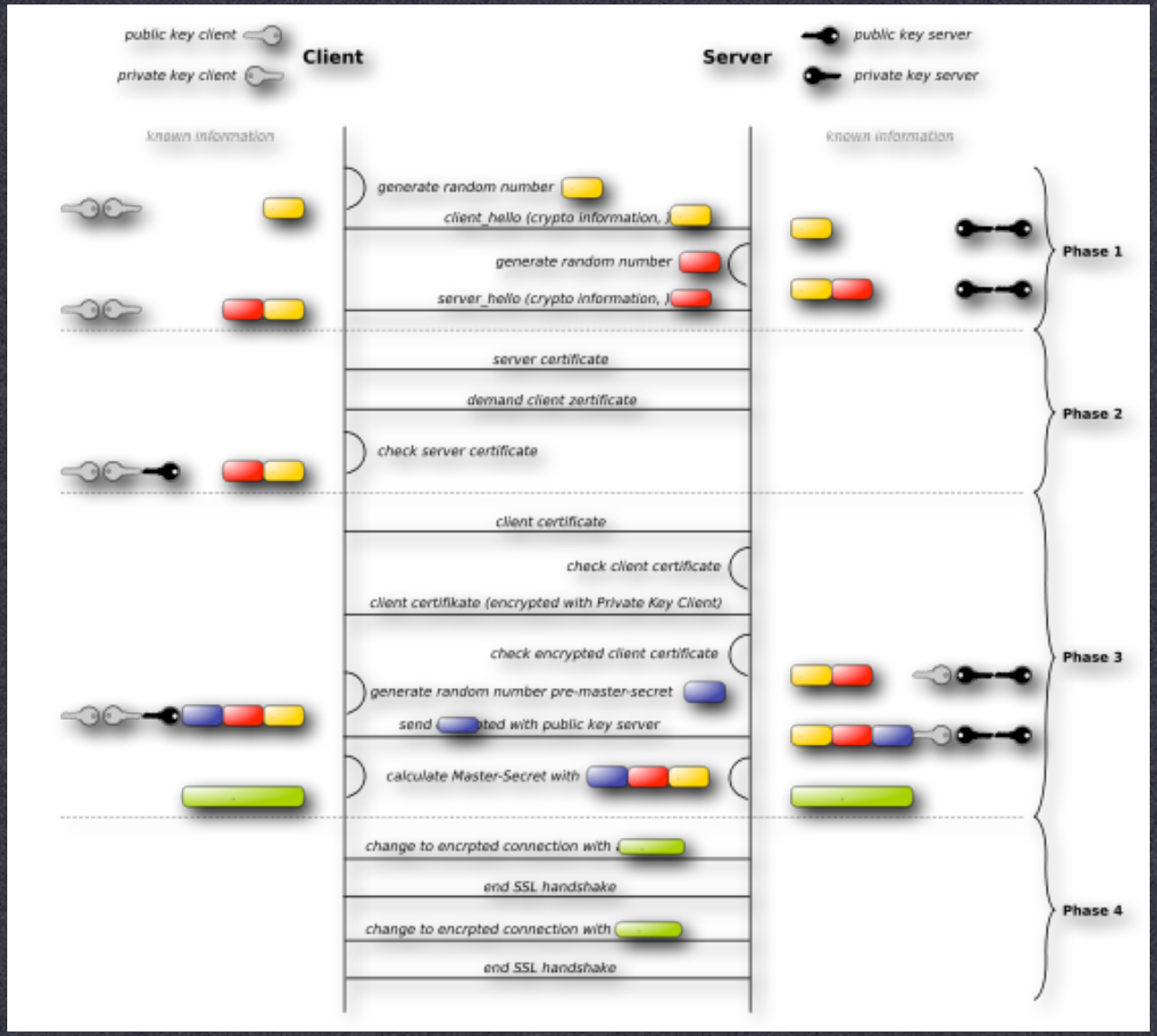


SSL3

- 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



Ciphersuite 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

Ciphersuite 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

Ciphersuite 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

Key Exchange - DH

Bank of America

Bank of Opportunity™

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 Rollback

Bank of America

Bank of Opportunity™

I'd like to use RSA

I'd like to use DH

DH Parameters (p, g, g^a)

RSA Encrypt: k^g mod 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

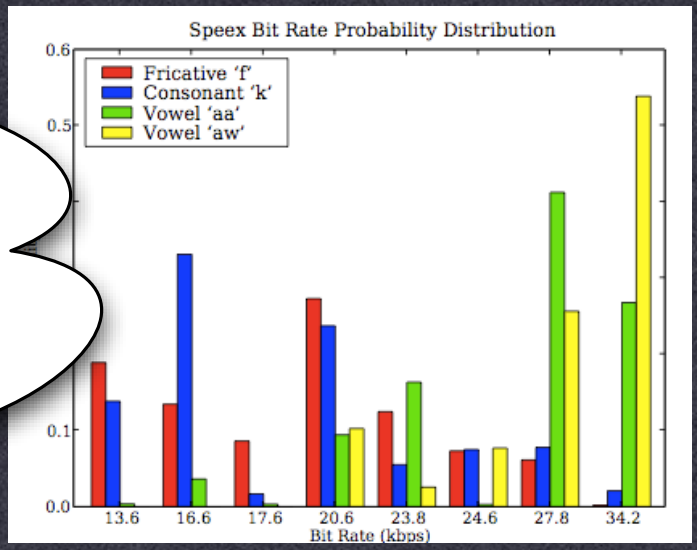


Image from Wright, Ballard, Coull, Monrose, Masson: Spot me if you Can, Uncovering spoken phrases in VoIP conversations