Asymmetric crypto
Symmetric

Before cryptography: exchanging keys

- **Secret key**
  - Only Alice and Bob know the secret key

- **Private key**
  - Only Alice's knows Alice's private key (Bob doesn't know and never finds out)
  - Only Bob knows Bob's private key (Alice doesn't know and never finds out)
In the food coloring demo, we assumed that mixing colors is cheap, but *un-mixing* them is prohibitively expensive.
Modular arithmetic

$5 + 7 = 2 \pmod{10}$

$7^2 = 9 \pmod{10}$

$8 + 8 = 6 \pmod{10}$
Modular arithmetic

$8 + 9 = ? \pmod{10}$
$4^3 = ? \pmod{10}$
$1 + 1 = ? \pmod{10}$
Modular arithmetic

\[ 8 + 9 = 7 \pmod{10} \]
\[ 4^3 = 4 \pmod{10} \]
\[ 1 + 1 = 2 \pmod{10} \]
Diffie-Hellman (1976)

\[ s = (A)^b \pmod{p} \]
\[ s = (B)^a \pmod{p} \]
<table>
<thead>
<tr>
<th>Alice</th>
<th>Known</th>
<th>Unknown</th>
<th>Bob</th>
<th>Known</th>
<th>Unknown</th>
<th>Eve</th>
<th>Known</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$p = 23$</td>
<td></td>
<td></td>
<td>$p = 23$</td>
<td></td>
<td></td>
<td>$p = 23$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$g = 5$</td>
<td></td>
<td></td>
<td>$g = 5$</td>
<td></td>
<td></td>
<td>$g = 5$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$a = 6$</td>
<td>$b$</td>
<td></td>
<td>$b = 15$</td>
<td>$a$</td>
<td></td>
<td>$a, b$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$A = 5^a \mod 23$</td>
<td></td>
<td></td>
<td>$B = 5^b \mod 23$</td>
<td></td>
<td></td>
<td>$A = 8, B = 19$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$A = 5^6 \mod 23 = 8$</td>
<td></td>
<td></td>
<td>$B = 5^{15} \mod 23 = 19$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$B = 19$</td>
<td></td>
<td></td>
<td>$A = 8$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$s = B^a \mod 23$</td>
<td></td>
<td></td>
<td>$s = A^b \mod 23$</td>
<td></td>
<td></td>
<td></td>
<td>$s$</td>
</tr>
<tr>
<td></td>
<td>$s = 19^6 \mod 23 = 2$</td>
<td></td>
<td></td>
<td>$s = 8^{15} \mod 23 = 2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Stolen from Wikipedia
Diffie-Hellman (1976)

- Security is based on the hardness of the discrete logarithm problem
- Can be used for key exchange
  - Not encryption/decryption
  - Not signatures, i.e., nonreputability
- Susceptible to trivial man-in-the-middle attacks if you don't independently verify the session key
- More commonly done with elliptic curves these days
Rivest-Shamir-Adleman (1977)

Mark Zuckerberg, President
RSA

Encryption:
\[ c \equiv m^e \mod n \]

Decryption:
\[ c^d \equiv (m^e)^d \mod n \]

RSA provides encryption, authentication, and non-repudiation.
RSA

- Security is based on the hardness of integer factorization
\( n = pq \)

- \( p \) and \( q \) are primes, suppose \( p = 61, \ q = 53 \)
- \( n = 3233 \)
- Euler's totient counts the positive integers up to \( n \) that are relatively prime to \( n \)
- \( \text{totient}(n) = (p - 1)(q - 1) = 780 \)
- Choose \( 1 < e < 780 \) coprime to 780, e.g., \( e = 17 \)
- \( d \) is the modular multiplicative inverse of \( e \), \( d = 413 \)
- \( 413 \times 17 \mod 780 = 1 \)
Public/private key pair

- Public key is \((n = 3233, e = 17)\)
- Private key is \((n = 3233, d = 413)\)
- Encryption: \(c(m = 65) = 65^{17} \mod 3233 = 2790\)
- Decryption: \(m = 2790^{413} \mod 3233 = 65\)
- Signature: \(s = 100^{413} \mod 3233 = 1391\)
- Verification: \(100 = 1391^{17} \mod 3233\)
- Fast modular exponentiation is the trick
- Using RSA for key exchange or encryption is often a red flag, more commonly used for signatures
QQ Browser Fail #1

- Keys should be 2048 or 4096 bits, at least
- 128 bits is pathetic
- \[245406417573740884710047745869965023463 = 14119218591450688427 \times 17381019776996486069\]
- [https://citizenlab.org/2016/03/privacy-security-issues-qq-browser/](https://citizenlab.org/2016/03/privacy-security-issues-qq-browser/)
QQ Browser Fail #2

- AES session key generation
  
  srand(currenttimeinmilliseconds)
  
  key = rand()
QQ Browser Fail #3

• RSA encrypt AES key (using public key of QQ's server) the AES session key and send it
  – Using textbook RSA encryption

• Textbook RSA is malleable...
Let $C$ be the RSA encryption of 128-bit AES key $k$ with RSA public key $(n, e)$. Thus, we have

$$C \equiv k^e \pmod{n}$$

Now let $C_b$ be the RSA encryption of the AES key

$$k_b = 2^b k$$

i.e., $k$ bitshifted to the left by $b$ bits. Thus, we have

$$C_b \equiv k_b^e \pmod{n}$$

We can compute $C_b$ from only $C$ and the public key, as

$$C_b \equiv C(2^{be} \pmod{n}) \pmod{n}$$

$$\equiv (k^e \pmod{n})(2^{be} \pmod{n}) \pmod{n}$$

$$\equiv k^e 2^{be} \pmod{n}$$

$$\equiv (2^b k)^e \pmod{n}$$

$$\equiv k_b^e \pmod{n}$$
Server chops off all but the lowest 128 bits
1. Record a session
2. Connect to the server with key shifted left 127 bits
3. Can you decrypt with 1000000.... or 0000000...?

(Just learned one bit of the key, repeat for left shift of 126 bits, 125 bits, etc. until you learn the key of the recorded session and can decrypt it)

This is a chosen ciphertext attack, and a padding oracle attack, but involves RSA padding rather than AES-CBC padding
Semantic security

- Basic problem: we don't know the format of the plaintext
- Desirable properties
  - Indistinguishability under Chosen Plaintext Attack (IND-CPA)
  - Indistinguishability under Chosen Ciphertext Attack (IND-CCA)
  - Indistinguishability under Adaptive Chosen Ciphertext Attack (IND-CCA2)
Forward secrecy

- Forward secrecy
  - Compromise of long-term keys does not compromise past session keys [Wikipedia]
- Need to generate an ephemeral key and then throw it out after the message is sent/received
Man-in-the-middle attacks
Fun with asymmetric (or other) crypto

• Ring signatures (don't know which group member signed)
• Threshold cryptography
• Identity-Based Encryption
• Secret sharing
• Homomorphic encryption
• Secure multi-party computation
Cryptovirology (1996)

- [Cryptovirology] by Young and Yung
- Ransomware (not counting AIDS trojan in 1989, started in 2005)
- Cryptocounters
- Cryptocurrency (Bitcoin in 2008)
- Mix networks (Tor paper presented in 2004)
- Private Information Retrieval (Chor et al., 1995)
- Subliminal Channels (Gustavus Simmons in 1984)
- Salami slicing (Superman III in 1983, Office Space in 1999)
- RNG biasing
The future?

Image taken from http://filipchsqroom.blogspot.com/
Deutsch-Jozsa algorithm

\[ |0\rangle \xrightarrow{H^\otimes n} |1\rangle \xrightarrow{H} |0\rangle \xrightarrow{U_f} |1\rangle \xrightarrow{H^\otimes n} \]

By Skippydo - Own work, Public Domain,
https://commons.wikimedia.org/w/index.php?curid=2547135
Quantum computing example

\[ f(x) = 10^x \pmod{12} \]

Shor's integer factorization algorithm involves a quantum Fourier transform.
Asymmetric crypto is under threat

- Some newer algorithms can't be broken by quantum computers
  - RSA, Diffie-Hellman, elliptic curves, *etc.* all can

- Symmetric crypto is okay
  - Grover's algorithm finds the input corresponding to an output in $O(\sqrt{N})$ time where $N$ is the size of the function's domain
    - $O(\sqrt{2^{128}}) = O(2^{64})$
    - $O(\sqrt{2^{256}}) = O(2^{128})$
References


- Lots of images and info plagiarized from Wikipedia