Internet Security?

- The Internet was *not* designed for security.
- Sending data via the Internet is like sending post cards through the mail ... ...when you don’t trust the Post Office.

A Typical Internet Session

<table>
<thead>
<tr>
<th>You (client)</th>
<th>Merchant (server)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>I want to make a purchase.</td>
<td></td>
</tr>
<tr>
<td>What is your Credit Card Number?</td>
<td></td>
</tr>
<tr>
<td>My Credit Card is 6543 2345 6789 8765.</td>
<td></td>
</tr>
</tbody>
</table>

Basic Encryption

Can we at least protect the credit card number so that it won’t be revealed to anyone except the intended merchant?

Kerckhoffs’s Principle (1883)

The security of a cryptosystem should depend only on the key.

You should assume that attackers know everything about your system *except* the key.

PINs, Passwords, & Keys

Informally ...
- A PIN is a 4-6 digit speed bump.
- A password is a short, user-chosen, usually guessable selection from a small dictionary.
- A key is an unguessable, randomly chosen string – usually at least 128 bits.
Off-Line Attacks
- Don't even think about using user-chosen passwords as encryption keys.
- Don't even think about using keys derived deterministically from user-chosen passwords.
- Given the ciphertext, an attacker can do a (guided) exhaustive search through the space to find the password.

Symmetric Encryption
- If the client has a pre-existing relationship with the merchant, the two parties may have a shared secret key K – known only to these two.
- User encrypts private data with key K.
- Merchant decrypts data with key K.

Symmetric Ciphers
Private-key (symmetric) ciphers are usually divided into two classes.
- Stream ciphers
- Block ciphers

Symmetric Ciphers
Private-key (symmetric) ciphers are usually divided into two classes.
- Stream ciphers
- Block ciphers

Stream Ciphers
RC4, A5/1, SEAL, etc.
- Use the key as a seed to a pseudo-random number-generator.
- Take the stream of output bits from the PRNG and XOR it with the plaintext to form the ciphertext.

Stream Cipher Encryption
- Plaintext: 
- PRNG(seed): 
- Ciphertext: 

Stream Cipher Decryption

Ciphertext: urrection
PRNG(seed): urrection
Plaintext: urrection

A PRNG: Alleged RC4

Initialization
S[0..255] = 0,1,…,255; j=0
K[0..255] = Key,Key,Key,…
for i = 0 to 255
  j = (j + S[i] + K[i]) mod 256
  swap S[i] and S[j]

A PRNG: Alleged RC4

Iteration
i = (i + 1) mod 256
j = (j + S[i]) mod 256
swap S[i] and S[j]
t = (S[i] + S[j]) mod 256
Output S[t]

Some Good Properties

• Stream ciphers are typically very fast.
• Stream ciphers can be very simple.
• The same function is used for encryption and decryption.

Stream Cipher Security

If two plaintexts are ever encrypted with the same stream cipher and key

\[ C_1 = K \oplus P_1, \]
\[ C_2 = K \oplus P_2, \]
an attacker can easily compute

\[ C_1 \oplus C_2 = P_1 \oplus P_2, \]
from which \( P_1 \) and \( P_2 \) can usually be teased apart easily.

Stream Cipher Encryption

Plaintext: urrection
PRNG(seed): urrection
Ciphertext: urrection
Stream Cipher Integrity

- It is easy for an adversary (even one who can’t decrypt the ciphertext) to alter the plaintext in a known way.

Bob to Bob’s Bank:
Please transfer $1,000,002.00 to the account of my good friend Alice.

Symmetric Ciphers

Private-key (symmetric) ciphers are usually divided into two classes.

- Stream ciphers
- Block ciphers

Block Ciphers

AES, DES, 3DES, Twofish, etc.

Plaintext Data → Block Cipher → Ciphertext

Usually 8 or 16 bytes

How to Build a Block Cipher

Plaintext → Key → Block Cipher → Ciphertext

Usually 16 or more bytes
Feistel Ciphers

Typically, Feistel ciphers are iterated for about 10-16 rounds.

Different “sub-keys” are used for each round.

Even a weak round function can yield a strong Feistel cipher if iterated sufficiently.
Feistel Ciphers

- Typically, Feistel ciphers are iterated for about 10-16 rounds.
- Different “sub-keys” are used for each round.
- Even a weak round function can yield a strong Feistel cipher if iterated sufficiently.

Transfer of Confidential Data

You (client) Merchant (server)

- I want to make a purchase.
- Please encrypt your credit number with our shared secret key.
- What is your Credit Card Number?
- My Credit Card is 6543 2345 6789 8765.

Asymmetric Encryption

- What if the user and merchant have no prior relationship?
- Asymmetric encryption allows someone to encrypt a message for a recipient without knowledge of the recipient’s decryption key.
The Fundamental Equation

\[ Z = Y^X \mod N \]

March 5, 2013

When \( Z \) is unknown, it can be efficiently computed.

March 5, 2013

The Fundamental Equation

\[ Z = Y^X \mod N \]

March 5, 2013

When \( X \) is unknown, the problem is known as the discrete logarithm and is generally believed to be hard to solve.

March 5, 2013

The Fundamental Equation

\[ Z = Y^X \mod N \]

March 5, 2013

The problem is not well-studied for the case when \( N \) is unknown.

March 5, 2013

How to compute \( Y^X \mod N \)

- Compute \( Y^X \) and then reduce \( \mod N \).
- If \( X \), \( Y \), and \( N \) each are 1,000-bit integers, \( Y^X \) consists of \( \sim 2^{1018} \) bits.
- Since there are roughly \( 2^{258} \) particles in the universe, storage is a problem.
How to compute $Y^X \mod N$

- Repeatedly multiplying by $Y$ by itself $X$ times (with a modulo $N$ reduction after each multiplication) solves the storage problem.

- However, we would need to perform $2^{908}$ 64-bit multiplications per second to complete the computation before the sun burns out.

How to compute $Y^X \mod N$

Multiplication by Repeated Doubling
To compute $X = Y$,
compute $Y, 2Y, 4Y, 8Y, 16Y, ...$
and sum up those values dictated by the binary representation of $X$.

Example: $26Y = 2Y + 8Y + 16Y$.

How to compute $Y^X \mod N$

Exponentiation by Repeated Squaring
To compute $Y^2$,
compute $Y, Y^2, Y^4, Y^8, Y^{16}, ...$
and multiply those values dictated by the binary representation of $X$.

Example: $Y^{26} = Y^2 = Y^8 = Y^{16}$.

How to compute $Y^X \mod N$

- We can now perform a 1,000-bit modular exponentiation using $\sim$1,000 1,000-bit modular multiplications.

- 1,000 squarings: $Y, Y^2, Y^4, ..., Y^{2^{1000}}$

- $\sim$500 “ordinary” multiplications

The Fundamental Equation

$Z = Y^X \mod N$

When $Y$ is unknown, the problem is known as discrete root finding and is generally believed to be hard to solve ... without the factorization of $N$.

RSA Encryption/Decryption

- Select two large primes $p$ and $q$.
- Publish the product $N = pq$.
- The exponent $X$ is typically fixed at 65537.
- Encrypt message $Y$ as $E(Y) = Y^X \mod N$.
- Decrypt ciphertext $Z$ as $D(Z) = Z^{1/X} \mod N$.
- Note $D(E(Y)) = (Y^X)^{1/X} \mod N = Y$. 

March 5, 2013
RSA Signatures and Verification

- Not only is \( D(B(Y)) = (x^2)^{1/2} \mod N = Y \), but also \( E(D(Y)) = (Y^{1/2})^2 \mod N = Y \).
- To form a signature of message \( Y \), create \( S = D(Y) = Y^{1/2} \mod N \).
- To verify the signature, check that \( E(S) = S^2 \mod N \) matches \( Y \).

Transfer of Confidential Data

You (client)  Merchant (server)

- I want to make a purchase.
- Here is my RSA public key \( E \).
- My Credit Card is \( E(6543234567898765) \).

Intermediary Attack

You (client)  Intermediary  Merchant (server)

- I want to make a purchase.
- My public key is \( E \).
- My public key is \( E(CCA) \).
- \( E(CCA) \) wants to make a purchase.
- My public key is \( E \).
- My public key is \( E(CCA) \).

Digital Certificates

“Alice’s public modulus is \( N_A = 331490324840 \) ..” -- signed ... someone you trust.

Transfer of Confidential Data

You (client)  Merchant (server)

- I want to make a purchase.
- Here is my RSA public key \( E \) and a cert.
- My Credit Card is \( E(6543234567898765) \).
Replay Attack

You (client)  Merchant (server)
I want to make a purchase.
Here is my RSA public key E and a cert.
My Credit Card is $(6543\ 2345\ 6789\ 8765)$.

Eavesdropper  Later ...  Merchant (server)
I want to make a different purchase.
Here is my RSA public key E and a cert.
My Credit Card is $(6543\ 2345\ 6789\ 8765)$.

Transfer of Confidential Data

You (client)  Merchant (server)
I want to make a purchase.
Here is my RSA public key E and a cert.
My Credit Card is $(6543\ 2345\ 6789\ 8765)$.

Later ...
I want to make a different purchase.
Here is my RSA public key E and a cert and a "nonce".
My Credit Card and your nonce are $E(6543\ 2345\ 6789\ 8765)$, nonce.

SSL/PCT/TLS History
- 1994: Secure Sockets Layer (SSL) V2.0
- 1995: Private Communication Technology (PCT) V1.0
- 1996: Secure Sockets Layer (SSL) V3.0
- 1997: Private Communication Technology (PCT) V4.0
- 1999: Transport Layer Security (TLS) V1.0

SSL/PCT/TLS Handshake

You (client)  Merchant (server)
Let's talk securely.
Here are the protocols and ciphers I understand.
I choose this protocol and these ciphers.
Here is my public key, a cert, a nonce, etc.
Using your public key, I've encrypted a random symmetric key (and your nonce).

SSL/PCT/TLS Agility

A principal reason for the success of SSL/TLS is its agility.
- The handshake negotiates symmetric and asymmetric ciphers, the hash function, and even the protocol's own version.
- This has allowed the protocol to survive and expand while many underlying primitives have been discredited or lost favor.

SSL/PCT/TLS Secure Channel

Once the negotiation is complete, all subsequent secure messages are sent
- encrypted – using the negotiated session key, and
- integrity checked with a keyed hash.
Hybrid Cryptography

- Asymmetric cryptography has many useful features not available in traditional symmetric cryptography.
- Symmetric cryptography is much more efficient than asymmetric.
- The practical hybrid is formed by using asymmetric cryptography to establish a secure channel and symmetric cryptography within the secure channel.

Application: Verifiable Elections

- Current election technology requires trust in the officials who manage elections, the equipment and its manufacturers, and the processes used in the election.
- Cryptography allows us to eliminate this trust.

A Verifiable Election

<table>
<thead>
<tr>
<th>Voter Name</th>
<th>Vote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice Smith</td>
<td>Jefferson</td>
</tr>
<tr>
<td>Bob Williams</td>
<td>Adams</td>
</tr>
<tr>
<td>Carol James</td>
<td>Adams</td>
</tr>
<tr>
<td>David Fuentes</td>
<td>Jefferson</td>
</tr>
<tr>
<td>Ellen Chu</td>
<td>Jefferson</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Jefferson</td>
<td>3</td>
</tr>
<tr>
<td>Adams</td>
<td>2</td>
</tr>
</tbody>
</table>

A Verifiable Election

<table>
<thead>
<tr>
<th>Voter Name</th>
<th>Vote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice Smith</td>
<td>Jefferson</td>
</tr>
<tr>
<td>Bob Williams</td>
<td>Adams</td>
</tr>
<tr>
<td>Carol James</td>
<td>Adams</td>
</tr>
<tr>
<td>David Fuentes</td>
<td>Jefferson</td>
</tr>
<tr>
<td>Ellen Chu</td>
<td>Jefferson</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Jefferson</td>
<td>3</td>
</tr>
<tr>
<td>Adams</td>
<td>2</td>
</tr>
</tbody>
</table>

A Verifiable Election

<table>
<thead>
<tr>
<th>Voter Name</th>
<th>Vote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice Smith</td>
<td>Jefferson</td>
</tr>
<tr>
<td>Bob Williams</td>
<td>Adams</td>
</tr>
<tr>
<td>Carol James</td>
<td>Adams</td>
</tr>
<tr>
<td>David Fuentes</td>
<td>Jefferson</td>
</tr>
<tr>
<td>Ellen Chu</td>
<td>Jefferson</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Jefferson</td>
<td>3</td>
</tr>
<tr>
<td>Adams</td>
<td>2</td>
</tr>
</tbody>
</table>

A Verifiable Election

<table>
<thead>
<tr>
<th>Voter Name</th>
<th>Vote</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice Smith</td>
<td>Jefferson</td>
</tr>
<tr>
<td>Bob Williams</td>
<td>Adams</td>
</tr>
<tr>
<td>Carol James</td>
<td>Adams</td>
</tr>
<tr>
<td>David Fuentes</td>
<td>Jefferson</td>
</tr>
<tr>
<td>Ellen Chu</td>
<td>Jefferson</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Jefferson</td>
<td>3</td>
</tr>
<tr>
<td>Adams</td>
<td>2</td>
</tr>
</tbody>
</table>
A Verifiable Election

<table>
<thead>
<tr>
<th></th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jefferson</td>
<td>3</td>
</tr>
<tr>
<td>Adams</td>
<td>2</td>
</tr>
</tbody>
</table>

The Voter’s Perspective

Systems that produce verifiable elections can be built to look exactly like current systems ...
- paper-based
- fully-electronic
- in-person
- remote

... with one addition ...

A Verifiable Receipt

The Voter’s Perspective

Voters can ...
- Use receipts to check their results are properly recorded on a public web site.
- Throw their receipts in the trash.
- Write and use their own election verifiers
- Download applications from sources of their choice to verify the mathematical proof of the tally.
- Believe verifications done by their political parties, LWV, ACLU, etc.
- Accept the results without question.

Some systems producing verifiable elections ...
Helios

STAR-Vote
- Voters use electronic ballot marking devices to indicate their preferences.
- When a voter’s selections are completed, the device provides the voter with a paper ballot summary and an encrypted receipt. It also records the encrypted ballot.
- The voter can review the paper ballot summary, and optionally deposit it in a ballot box.
- All encrypted ballots are posted, but the only votes counted are those for which a corresponding paper ballot has been deposited. The remaining ballots are decrypted.

Benefits of E2E-Verifiability
- Strong public assurance of election integrity
- Elimination of trust requirements
- Certification relief

Questions???