CIS433/533 - Computer and Network Security
Cryptography

Professor Kevin Butler
Winter 2011
A historical moment ... 

- Mary Queen of Scots is being held by Queen Elizabeth ... 
  - ... and accused of treason. 
  - All communication with co-conspirators encrypted. 
  - Cipher was “unbreakable”.

- Walsingham needs to prove complicity.
Intuition

- Cryptography is the art (and sometimes science) of secret writing
  - Less well known is that it is also used to guarantee other properties, e.g., authenticity and integrity of data
  - This is an mathematically deep and important field
  - However, much of our trust in cryptographic systems is based on faith (particularly in efficient secret key algorithms)
  - … ask Mary Queen of Scots how that worked out.

- This set of lectures will provide the intuition and some specifics of modern cryptography, seek others for additional details (Menezes et. al.).
Cryptography

- Cryptography (cryptographer)
  - Creating ciphers
- Cryptanalysis (cryptanalyst)
  - Breaking ciphers

- The history of cryptography is an arms race between cryptographers and cryptanalysts
Encryption algorithm

- Algorithm used to make content unreadable by all but the intended receivers

\[
\text{Encrypt(plaintext, key)} = \text{ciphertext}
\]
\[
\text{Decrypt(ciphertext, key)} = \text{plaintext}
\]

- Algorithm is public, key is private

- Block vs. Stream Ciphers
  - Block: input is fixed blocks of same length
  - Stream: stream of input
Hardness and security...

- Functions
  - Plaintext $P$
  - Ciphertext $C$
  - Encryption ($E$) key $k_e$
  - Decryption ($D$) key $k_d$

$$D(E(P, k_e), k_d) = P$$

- Computing $P$ from $C$ is hard, computing $P$ from $C$ with $k_d$
  - Is easy for all $P$s (operation true for all inputs)...
  - ... except in some vanishingly small number of cases
Example: Caesar Cipher

- Substitution cipher
- Every character is replaced with the character three slots to the right

Q: What is the key?

SECURITY AND PRIVACY
VHFUXULWBDQGSULYDFB
Cryptanalyze this ....

“BERTBA ARGJBEX FRPHEVGL”
Cryptanalysis of ROTx Ciphers

- Goal: to find plaintext of encoded message
- Given: ciphertext
- How: simply try all possible keys
  - Known as a brute force attack

1  T  F  D  V  S  J  U  Z  B  M  E  Q  S  J  W  B  D  Z
2  U  G  E  W  T  K  V  A  C  N  F  R  T  H  X  C  E  A
3  W  H  F  X  U  L  W  B  D  Q  G  S  U  L  Y  D  F  B

SECURITY AND PRIVACY
Attacking a Cipher

- The attack mounted will depend on what information is available to the adversary
  - *Ciphertext-only attack*: adversary only has the ciphertext available and wants to determine the plaintext encrypted
  - *Known-plaintext attack*: adversary learns one or more pairs of ciphertext/plaintext encrypted under the same key, tries to determine plaintext based on a different ciphertext
  - *Chosen-plaintext attack*: adversary can obtain the encryption of any plaintext, tries to determine the plaintext for a different ciphertext
  - *Chosen-ciphertext attack*: adversary can obtain the plaintext of any ciphertext except the one the adversary wants to decrypt
Shared key cryptography

• Traditional use of cryptography
• Symmetric keys, where a single key \((k)\) is used is used for encryption \((E)\) and decryption \((D)\)

\[ D(E(p,k),k) = p \]

• All (intended) receivers have access to key
• Note: Management of keys determines who has access to encrypted data
  ‣ E.g., password encrypted email
• Also known as symmetric key cryptography
Key size and algorithm strength

- Key size is an oft-cited measure of the strength of an algorithm, but is strength strongly correlated (or perfectly correlated with key length)?
  - Say we have two algorithms, A and B with key sizes of 128 and 160 bits (the common measure)
  - Is A less secure than B?
  - What if A=B (for variable key-length algorithms)?
- Terminology: key length is the security parameter.
Is there an unbreakable cipher?

- As it turns out, yes ....
  - (Claude Shannon proved it)
The one-time pad (OTP)

- Assume you have a secret bit string $s$ of length $n$ known only to two parties, Alice and Bob
  - Alice sends a message $m$ of length of $n$ to Bob
  - Alice uses the following encryption function to generate ciphertext bits: 
    \[ \sum_{i=0}^{n} c_i = m_i \oplus k_i \]
    - E.g., XOR the data with the secret bit string
  - An adversary Mallory cannot retrieve any part of the data

- Simple version of the proof of security:
  - Assume for simplicity that value of each bit in $m$ is equally likely, then you have no information to work with.
Data Encryption Standard (DES)

- Introduced by the US NBS (now NIST) in 1972
- Signaled the beginning of the modern area of cryptography
- Block cipher
  ‣ Fixed sized input
- 8-byte input and a 8-byte key (56-bits + 8 parity bits)
Breaking Ciphers

- **Brute force cryptanalysis**
  - Just keep trying different keys and check result (early breaks)

- **Linear cryptanalysis**
  - Construct linear equations relating plaintext, ciphertext and key bits that have a high bias; that is, whose probabilities of holding (over the space of all possible values of their variables) are as close as possible to 0 or 1
  - Use these linear equations in conjunction with known plaintext-ciphertext pairs to derive key bits.

- **Differential cryptanalysis**
  - study of how differences in an input can affect the resultant difference at the output (showing non-random behavior)
  - Use *chosen plaintext* to uncover key bits
Substitution Box (S-box)

- A substitution box (or S-box) is used to obscure the relationship between the plaintext and the ciphertext
  
  ‣ Shannon's property of **confusion**: the relationship between key and ciphertext is as complex as possible.  
  
  ‣ In DES S-boxes are carefully chosen to resist cryptanalysis.  
  
  ‣ Thus, that is where the security comes from.

Example: Given a 6-bit input, the 4-bit output is found by selecting the row using the outer two bits, and the column using the inner four bits. For example, an input "011011" has outer bits "01" and inner bits "1101"; the corresponding output would be "1001".
Cryptanalysis of DES

- DES has an effective 56-bit key length
- Wiener: $1,000,000 - 3.5$ hours (never built)
- July 17, 1998, the EFF DES Cracker, which was built for less than $250,000 < 3$ days
- January 19, 1999, Distributed.Net (w/EFF), 22 hours and 15 minutes (over many machines)
- We all assume that NSA and agencies like it around the world can crack (recover key) DES in milliseconds

What now? Give up on DES?
Variants of DES

- DESX (XOR with separate keys $\sim 60$-bits)
  - Linear cryptanalysis

- Triple DES (three keys $\sim 112$-bits)
  - Keys $k_1, k_2, k_3$

$$C = E(D(E(p, k_1), k_2, k_3))$$

Diagram:

```
  k1  k2  k3
  |   |   |
  v   v   v
  p   E   D   E  c
```
Advanced Encryption Standard (AES)

- International NIST bakeoff between cryptographers
  - Rijndael (pronounced “Rhine-dall”)

- Replacement for DES/accepted symmetric key cipher
  - Substitution-permutation network, not a Feistel network
  - Variable key lengths
  - Fast implementation in hardware and software
  - Small code and memory footprint
Public Key Cryptography

- Public Key cryptography
  - Each key pair consists of a public and private component: $k^+$ (public key), $k^-$ (private key)
    \[
    D(E(p, k^+), k^-) = p
    \]
    \[
    D(E(p, k^-), k^+) = p
    \]
- Public keys are distributed (typically) through public key certificates
  - Anyone can communicate secretly with you if they have your certificate
  - E.g., SSL-based web commerce
Hash Algorithms

- Hash algorithm
  - Compression of data into a hash value
  - E.g., \( h(d) = \text{parity}(d) \)
  - Such algorithms are generally useful in algorithms (speed/space optimization)

- … as used in cryptosystems
  - One-way - (computationally) hard to invert \( h() \), i.e., compute \( h^{-1}(y) \), where \( y = h(d) \)
  - Collision resistant hard to find two data \( x_1 \) and \( x_2 \) such that \( h(x_1) = h(x_2) \)

- Q: What can you do with these constructs?
Hash Functions

- Design a “strong cryptographic hash function”
- No formal basis
  - Concern is backdoors
- MD2
  - Substitution based on pi
- MD4, MD5
  - Similar, but complex functions in multiple passes
- SHA-1
  - 160-bit hash
  - “Complicated function”
Message Authentication Code

- **MAC**
  - Used in protocols to authenticate content, authenticates integrity for data $d$
  - To simplify, hash function $h()$, key $k$, data $d$
    \[
    MAC(k, d) = h(k \oplus d)
    \]
    - E.g., XOR the key with the data and hash the result

- **Q:** Why does this provide integrity?
  - Cannot produce $mac(k,d)$ unless you know $k$ and $d$
  - If you could, then can *invert* $h()$
HMAC

- MAC that meets the following properties
  - Collision-resistant
  - Attacker cannot compute proper digest without knowing K
    - Even if attacker can see an arbitrary number of digests H(k+x)
- Simple MAC has a flaw
  - Block hash algorithms mean that new content can be added
  - Turn H(K+m) to H(K+m+m’) where m’ is controlled by an attacker
- HMAC(K, d) = H(K + H(K + d))
  - Attacker cannot extend MAC as above
  - Prove it to yourself
Birthday Attack

- A birthday attack is a name used to refer to a class of brute-force attacks.
  - birthday paradox: the probability that two or more people in a group of 23 share the same birthday is greater than 50%.

- General formulation
  - function f() whose output is uniformly distributed
  - On repeated random inputs \( n = \{ n_1, n_2, \ldots, n_k \} \)
    - \( \Pr(n_i = n_j) = 1.2k^{1/2} \), for some \( 1 \leq i,j \leq k, 1 \leq j < k, i \neq j \)
    - E.g., \( 1.2(365^{1/2}) \approx 23 \)

- Q: Why is resilience to birthday attacks important?
Using hashes as authenticators

- Consider the following scenario
  - Prof. Alice has not decided if she will cancel the next lecture.
  - When she does decide, she communicates to Bob the student through Mallory, her evil TA.
  - She does not care if Bob shows up to a cancelled class.
  - Alice does not trust Mallory to deliver the message.

- She and Bob use the following protocol:
  1. Alice invents a secret t
  2. Alice gives Bob $h(t)$, where $h()$ is a crypto hash function
  3. If she cancels class, she gives $t$ to Mallory to give to Bob
    - If does not cancel class, she does nothing
    - If Bob receives the token $t$, he knows that Alice sent it
Hash Authenticators

- Why is this protocol secure?
  - \( t \) acts as an authenticated value (authenticator) because Mallory could not have produced \( t \) without inverting \( h() \)
  - **Note**: Mallory can convince Bob that class is occurring when it is not by simply not delivering \( h(t) \) (but we assume Bob is smart enough to come to that conclusion when the room is empty)

- What is important here is that hash preimages are good as (single bit) authenticators.

- Note that it is important that Bob got the original value \( h(t) \) from Alice directly (was provably authentic)
Hash chain

- Now, consider the case where Alice wants to do the same protocol, only for all 26 classes (the semester)

- Alice and Bob use the following protocol:

  1. Alice invents a secret $t$
  2. Alice gives Bob $H^{26}(t)$, where $H^{26}()$ is 26 repeated uses of $H()$.
  3. If she cancels class on day $d$, she gives $H^{(26-D)}(t)$ to Mallory, e.g.,
     - If cancels on day 1, she gives Mallory $H^{25}(t)$
     - If cancels on day 2, she gives Mallory $H^{24}(t)$
     .......
     - If cancels on day 25, she gives Mallory $H^{1}(t)$
     - If cancels on day 26, she gives Mallory $t$
  4. If does not cancel class, she does nothing
   - If Bob receives the token $t$, he knows that Alice sent it
Why is this protocol secure?

- On day $d$, $H^{(26-d)}(t)$ acts as an authenticated value (authenticator) because Mallory could not create $t$ without inverting $H()$ because for any $H^k(t)$ she has $k > (26-d)$

- That is, Mallory potentially has access to the hash values for all days prior to today, but that provides no information on today’s value, as they are all post-images of today’s value

- Note: Mallory can again convince Bob that class is occurring by not delivering $H^{(26-d)}(t)$

- Chain of hash values are ordered authenticators

- Important that Bob got the original value $H^{26}(t)$ from Alice directly (was provably authentic)
Basic truths of cryptography …

- Cryptography is not frequently the source of security problems
  - Algorithms are well known and widely studied
    - Use of crypto commonly is … (e.g., WEP)
  - Vetted through crypto community
  - Avoid any “proprietary” encryption
  - Claims of “new technology” or “perfect security” are almost assuredly snake oil
Common issues that lead to pitfalls

- Generating randomness
- Storage of secret keys
- Virtual memory (pages secrets onto disk)
- Protocol interactions
- Poor user interface
- Poor choice of key length, prime length, using parameters from one algorithm in another