

- Number theory review
- RSA Example
- Applications of Public Key Encryption:
  - Key Distribution and Management
  - Certificates and User Authentication

*CSCI283 Fall 2003 Lecture 4*  
*GWU*

# Announcements

# NSF Fellowship applications

- NSF fellowships for full-time grad students who are US citizens or permanent residents

<http://www.gwu.edu/~fellows/nsf.html>

Poorvi out of town 25<sup>th</sup> to 1<sup>st</sup>

No office hours

Send questions by email.

Try to give you a response in 24 hours

# Number theory review

(from Memon's notes, Brooklyn Poly)

Modular arithmetic

# Mathematical Operations

Calculate the following:

- $(5 - 7) \bmod 9$

$$-2 \bmod 9 = 7 \bmod 9$$

- $\sqrt{5} \bmod 11$

$$2^2 = 4; 3^2 = 9; 4^2 = 16 = 5; 7^2 = 49 = 5;$$

4 and 7 are square roots of 5 mod 11

# Efficient exponentiation (from Memon notes)

Compute  $x^c \bmod n$

Represent  $c$  as bit string  $b_{k-1} \dots b_0$  and use the following algorithm:

$z = 1$

*For  $i = k-1$  downto  $0$  do*

*$z = z^2 \bmod n$*

*if  $b_i = 1$  then  $z = z x \bmod n$*

How many multiplications?  $k = 2\lceil \log_2 c \rceil$

$$3^{11} \bmod 35$$

$i$	$b_i$	$z$	
3	1	1	3
2	0	9	9
1	1	$81 \bmod 35 = 11$	$33 = -2$
0	1	4	12

# RSA Again

# RSA Example: keys

- Large primes  $p$  and  $q$

3, 5

- Their product  $n$

15

- A number  $e$  relatively prime to  $(p-1)(q-1)$

7

- A number  $d$  using number theory

7

# RSA Example encryption

Encryption with  $e$

$$c = m^e$$

$$m = 3 \text{ mod } 15$$

$$c = 3^7 \text{ mod } 15$$

$i$	$b_i$	$z$	
2	1	1	3
1	1	9	$27=12=-3$
0	1	9	$27=12$

# RSA Example decryption

Decryption

$$m = c^d;$$

$$m = 12^7 \text{ mod } 15$$

i	$b_i$	z	
2	1	1	$12 \equiv -3$
1	1	9	$108 \equiv 3$
0	1	9	3

# Applications of public key encryption

- ✓ Digital signatures
- > Key management and exchange
- Certificates and user authentication

Applications of Public Key Encryption:  
Digital Signatures  
Do problem in handout

# Digital Signatures – signing and verification

## Digital Signatures – Signing.

- Alice signs  $m$  to get

$$S_{\text{private}(A)}(m) = E_{\text{private}(A)}(h(m))$$

- She then encrypts with Bob's public key to get

$$E_{\text{public}(B)}[m \parallel S_{\text{private}(A)}(m)].$$

# Signature Verification

- Bob decrypts with private key to get

$$D_{\text{private}(B)} E_{\text{public}(B)}[m \parallel a] = m \parallel a$$

- Bob then verifies Alice's signature with her public key to get

$$D_{\text{public}(A)}[a] ? h(m)$$

- It should match, as it would if  $a = S_{\text{private}(A)}(m)$

# Applications of Public Key Encryption: Key Exchange and Management (from Memon notes)

# Protocol III

## Session Key Exchange With Public Keys

- Alice gets Bob's public key from KDC.
- Alice generates a random session key, encrypts with Bob's public key and sends to Bob.
- Bob decrypts using his private key to get session key.
- Alice and Bob exchange a challenge-response.

Above is still susceptible to *man-in-the-middle* attack.

# Protocol III

## Man-in-the-middle Attack

- Alice send request to KDC. Mallory intercepts and sends his own public key.
- Alice generates random session key and encrypts using Mallory's (she thinks Bob's) public key and sends to Bob.
- Mallory intercepts session key, decrypts, then encrypts with Bob's public key and sends to Bob.
- Bob decrypts session key.
- Protection: Alice and Bob use a session key to communicate that Bob knows!

# Some more number theory

## primitive roots

- A primitive root in  $Z_n$  is a number ( $\neq 0$ ) whose powers generate all other numbers in  $Z_n$

- Example:  $n = 7$

$$1^2 = 1 \text{ not } 1$$

$$2^2 = 4 \quad 2^3 = 8 = 1 \text{ no more, not } 2$$

$$3^2 = 9 = 2 \quad 3^3 = 6 \quad 3^4 = 18 = 4 \quad 3^5 = 12 = 5 \quad 3^6 = 15 = 1$$

$$(1, 3, 2, 6, 4, 5)$$

$$\log_3 6 \text{ mod } 7 = ?$$

# The Discrete Log Problem

- Given  $y$  and  $a$  in  $Z_p$  where  $p$  is prime, find the unique  $x$  in  $Z_p$ , such that  $y = a^x \pmod p$ , i.e. find  $\log_a$
- Example: given 949 and 2 in  $Z_{2579}$ , find the unique  $x$  such that  $2^x = 949 \pmod{2579}$ !!  
 $2^{765} = 949 \pmod{2579}$ . Check.
- No efficient algorithm known

# Protocol IV: Diffie-Hellman Key Exchange

- Protocol for exchanging secret key over public channel.
- Select global parameters  $n$  and  $g$ .  $n$  is prime and  $g$  is a *primitive root* in  $Z_n$ . These parameters are public and known to all.

# Protocol IV – contd.

## Diffie-Hellman Key Exchange

- Alice privately selects random  $a$  and sends to Bob  $g^a \bmod n$ .
- Bob privately selects random  $b$  and sends to Alice  $g^b \bmod n$ .
- Alice and Bob privately compute  $g^{ab}$  which is their shared secret.
- An observer Oscar can only compute  $g^{a \bmod n} * g^b \bmod n = g^{a+b} \bmod n$ . To compute  $g^{ab}$  he needs to know either  $a$  or  $b$  or solve the discrete log problem.
- This is a *key agreement* protocol.

# Public Key Management

- Diffie-Hellman is susceptible to man-in-the-middle attack.
  - Mallory captures  $a$  and  $b$  in transmission and replaces with own  $a'$  and  $b'$ .
  - Essentially runs two Diffie-Hellman's. One with Alice and one with Bob.
- How do you trust a public key?
  1. Public announcement of keys.
  2. Publicly available directory.
  3. Public Key Authority.
  4. Web of Trust (PGP).

# Method V: Public Key Management

## Public Key Certificates.

- Certificate consisting of user's ID and public key.
  - Signed by a trusted third party – A Certificate Authority (CA).
  - There could be many CA's.
  - There could be a hierarchy of CA's.
- 
- Which approach is best?
    - 1 and 2 are really no solution.
    - 3 does not scale well.
    - 4 is interesting but has not succeeded in practice as expected.
    - 5 is the solution embraced by industry. This is what we study in more detail.

# Public Directory or Authority Solutions.

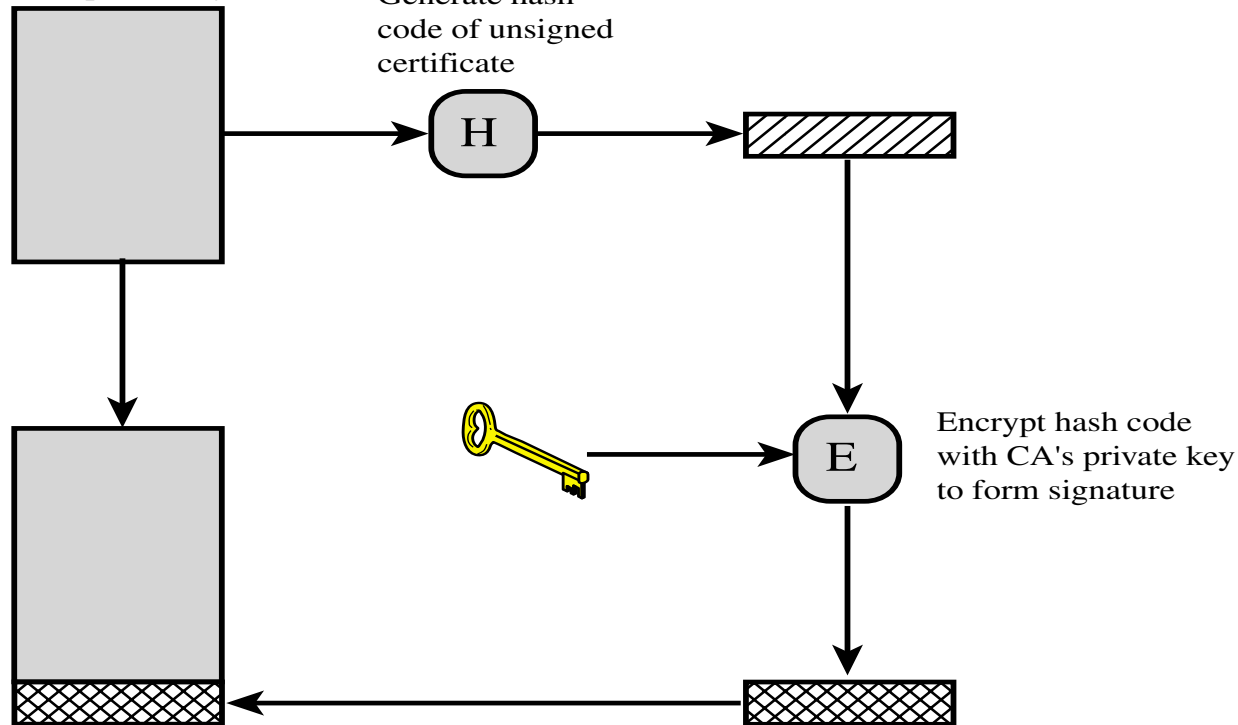
- If Alice and Bob want to talk they both get each others public key from central directory or authority.
- Disadvantages
  - Does not scale well.
  - KDC has to store every user's public key.
  - KDC provides single point of failure.
  - Performance bottleneck.
- Multiple directories or authorities do not solve the above problems.
- Instead – Digital Certificates.

# Public Key Certificate

- *Public Key Certificate* – Signed messages specifying a name (identity) and the corresponding public key.
- Signed by whom – *Certification Authority (CA)*, an organization that issues public key certificates.
- We assume that everyone is in possession of a trusted copy of the CA's public key.
- CA could be
  - Internal CA.
  - Outsourced CA.
  - Trusted Third-Party CA.

# Public Key Certificate

Unsigned certificate:  
contains user ID,  
user's public key



Signed certificate:  
Recipient can verify  
signature using CA's  
public key

Note: Mechanism of certification and content of certificate, will vary but at the minimum we have email verification and contains ID and Public Key.

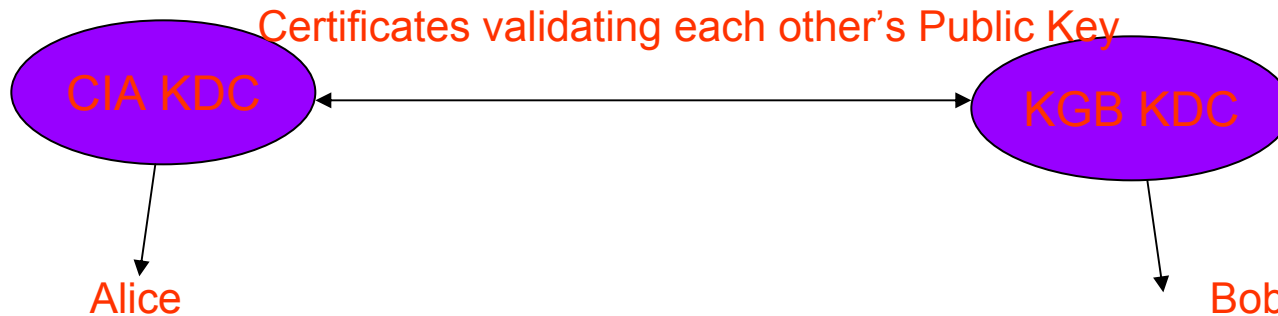
# Certificate Revocation

- Need some mechanism to *revoke* certificates
  - Private key compromised.
  - CA mistake in issuing certificate.
  - Particular service the certificate grants access to may no longer exist.
  - CA compromised.
- Expiration time only partial solution
- Certification Revocation Lists (CRL)
  - a list of every certificate that has been revoked but not expired, quickly grows large!
- CRL's distributed periodically.
  - What about time period between revocation and distribution of CRL?

# Advantages of CA Over KDC

- CA does not need to be on-line!
- CA can be very simple computing device.
- If CA crashes, life goes on (except CRL).
- Unlike keys, certificates can be stored in an insecure manner!!
- Scales well.
- Certificates also used for identification/user authentication

# Multiple CA's



- How does Alice talk to Bob?
- She obtains Bob's certificate signed by KGB-KDC.
- She obtains KGB-KDC's certificate signed by CIA-KDC.
- Concept can be generalized to multiple CA's.
- Helps if they are organized in a hierarchy.

# Applications of Public Key Encryption: Certificates and User Authentication

# Methods of user authentication (Text, sections 2.8 and 4.5)

## Something the user

- has (car keys, college ID, passport)
- knows (password, PIN, answers to questions)
- is (biometrics: retina scans, fingerprinting)

## 1. Regular Password

### Possible Attacks:

Dictionary attack: Try all passwords/brute force; try probable passwords

Wire-tapping/listening in: See password go by  
Search system/back-ups for password list

# Methods of user authentication - contd. (Text, sections 2.8 and 4.5)

## 2. Password with limited physical access

Attacks: As in 1

## 3. Secure hash of password

Store and compare one-way function of password,  $h(\text{password})$

Attacks: If I notice that my hashed password is the same as Ben's, I know our passwords are also the same

# Methods of user authentication – contd. (Text, sections 2.8 and 4.5)

## 4. Adding Salt

12 bit number formed using clock and process ID.

Stored value is  $h(\text{password} + \text{salt}) \parallel \text{salt}$

Can be stored in open

## 5. One-time Passwords/Challenge-response

System provides  $x$ , user responds with  $f(x)$

System provides seed, user generates random number

System provides  $E_{\text{public\_Alice}}(x)$ , Alice responds with

$E_{\text{private\_Alice}}(x \parallel f(x))$

(Proof of knowledge of her private key)

# Methods of user authentication – contd. (Text, sections 2.8 and 4.5)

## Typical attacks:

- Trojan horse that masquerades as system

- Ask system to stop all processes when authenticating

- Ctrl Alt Del

## 6. Public key certificates

- User shows something that links her name to her public key

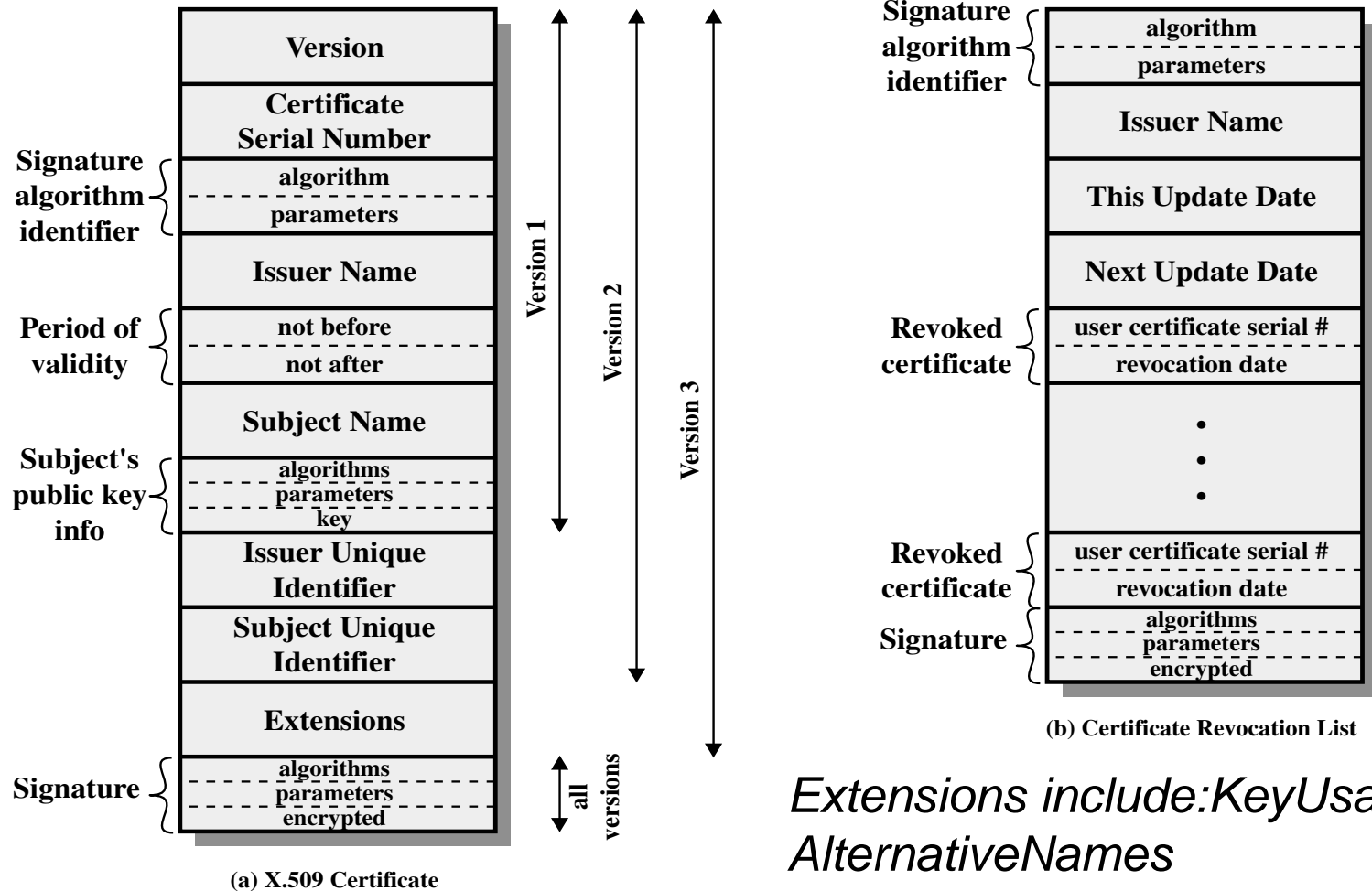
- Then demonstrates POK of her private key

- Standard way of doing this?

# X.509

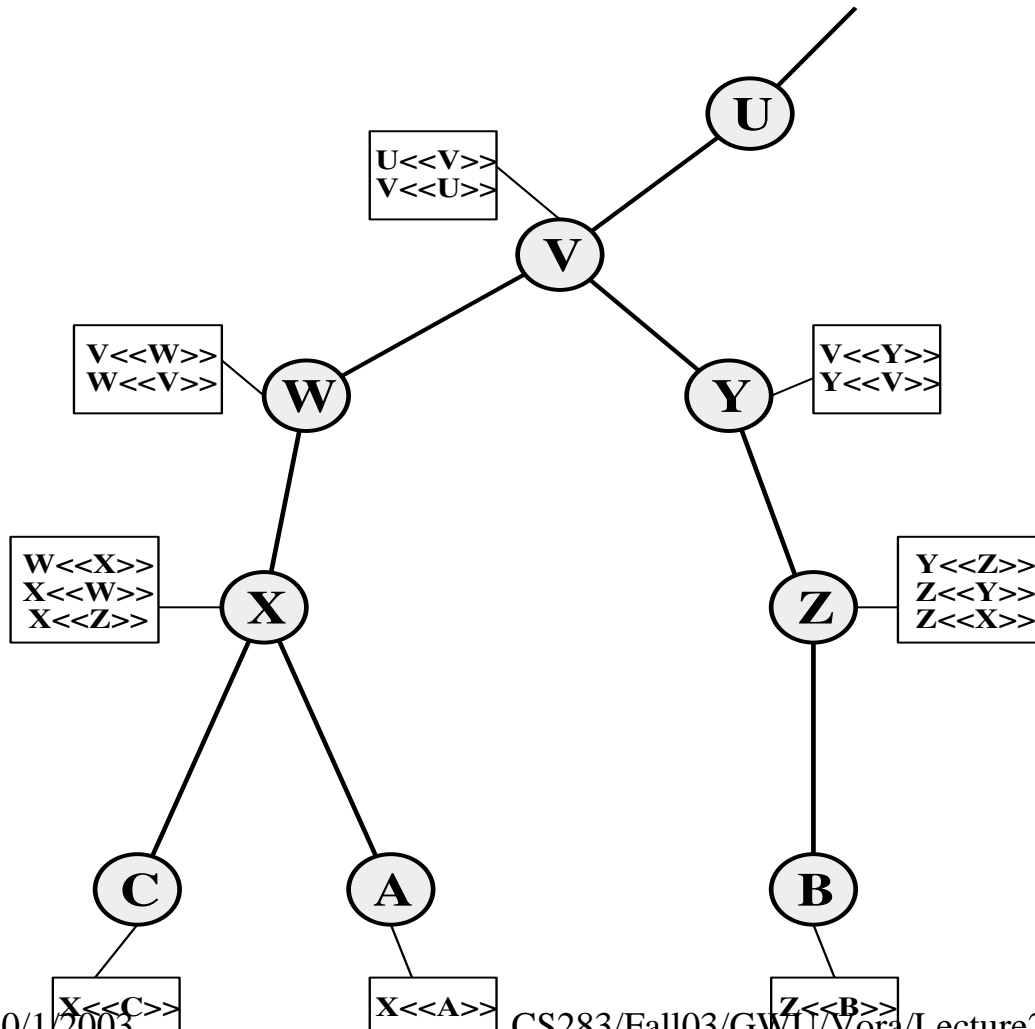
- Internet Engineering Task Force (IETF)'s X.509.
- Originally 1988, revised 93 and 95.
- X.509 is part of X.500 series that defines a directory service.
- Defines a framework for authentication services by X.500 directory to its users.
- Used in many other standards
- Does not dictate use of specific algorithm (recommends RSA).

# X.509 Certificate



*Extensions include: KeyUsage, AlternativeNames*

# X.509 CA Hierarchy – Example.



$Y\langle\langle X \rangle\rangle$  means the certificate of user X issued by CA Y.

To talk to B, A obtains the following chain

$X\langle\langle W \rangle\rangle$   
 $W\langle\langle V \rangle\rangle$   
 $V\langle\langle Y \rangle\rangle$   
 $Y\langle\langle Z \rangle\rangle$   
 $Z\langle\langle B \rangle\rangle$

Simpler if X has  $X\langle\langle Z \rangle\rangle$

# Authentication using PK

Message 1  $A \rightarrow B : A, \{T_A, N_A, B, X_A, \{\{Y_A\}_{K_{A-}}\}_{K_B}\}_{K_{A-}}$   
Message 2  $B \rightarrow A : B, \{T_B, N_B, A, X_B, \{Y_B\}_{K_A}\}_{K_{B-}}$   
Message 3  $A \rightarrow B : A, \{B, N_A\}_{K_{A-}}$

From <http://maga.di.unito.it/security/tutorial/node10.html>

Notice: both user (Alice) and system (Bob) can be authenticated to each other this way

# Public-key Infrastructure (PKI)

- Combination of digital certificates, public-key cryptography, and certificate authorities.
- A typical enterprise's PKI encompasses
  - issuance of digital certificates to users and servers
  - end-user enrollment software
  - integration with corporate certificate directories
  - tools for managing, renewing, and revoking certificates; and related services and support
- Verisign and Entrust – PKI providers.
- Your own PKI using Netscape/Microsoft certificate servers

# Ten Risks of PKI – Ellison and Schneier

- Who do we trust, and for what?
- Who is using my key?
- How secure is the verifying computer?
- Which John Robinson is he?
- Is the CA an authority?
- Is the user part of the security design?
- Was it one CA or a CA plus a Registration Authority?
- How did the CA identify the certificate holder?
- How secure are the certificate practices?
- Why are we using the CA process, anyway?

## Further Reading, PKI

- X.509 page <http://www.ietf.org/html.charters/pkix-charter.html>
- Ten Risks of PKI - <http://www.counterpane.com/pki-risks.html>

# Summary

## Crypto and Cryptographic Protocols

- Classical ciphers
  - Shift
  - Substitution
  - One-time Pad
  - Scrambling/Permutation
  - Cryptanalysis of all except one-time pad
- Private key encryption
  - DES
  - AES

# Summary – contd.

- Public Key Encryption
  - Uses
    - Authentication of sender
    - Encryption to prevent message from being read by another (privacy)
  - Methods
    - RSA
- One-way functions
  - Collision Resistance
  - Birthday attack

# Summary – contd.

- Applications of crypto
  - Digital Signatures
  - Key Exchange and Agreement (Diffie-Hellman, Discrete Log)
  - Public Key Authentication (X.509) (and other methods of authentication)

# References

- Bruce Schneier, *Applied Cryptography*
- Douglas Stinson, *Cryptography Theory and Practice*
- Dominic Welsh, *Cryptography and Codes*
- RSA FAQ <http://www.rsasecurity.com/rsalabs/faq/>