650.445: Practical Cryptographic Systems

Symmetric Cryptography II & Asymmetric Cryptography

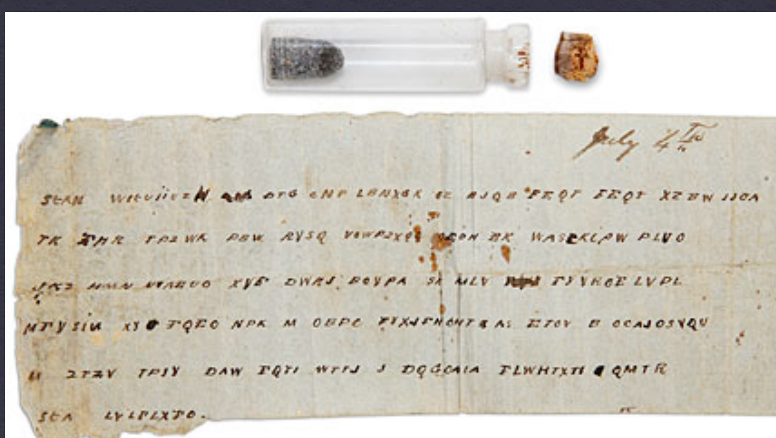
Instructor: Matthew Green

Housekeeping

- Office hours:
  - Thurs: after class & by appointment
  - My email: mgreen@cs.jhu.edu
  - Readings (for Tues):
    Guttman, Lessons Learned
    http://www.cs.auckland.ac.nz/~pgut001/pubs/usenix02.pdf

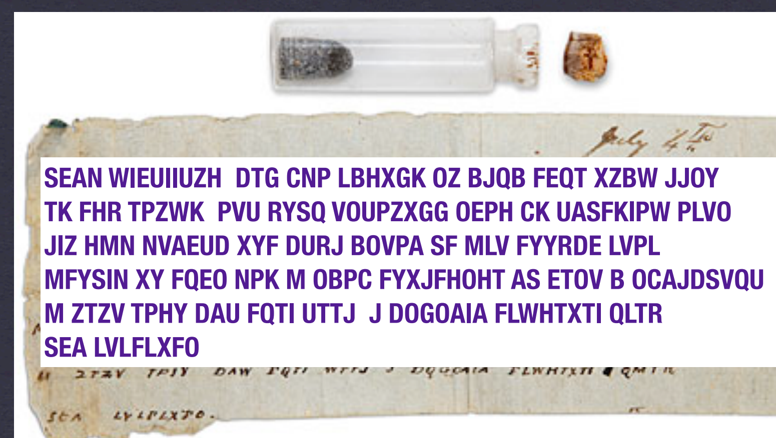
    Dowd (Actionscript):
    http://documents.iss.net/whitepapers/IBM_X-Force_WP_final.pdf

An aside...



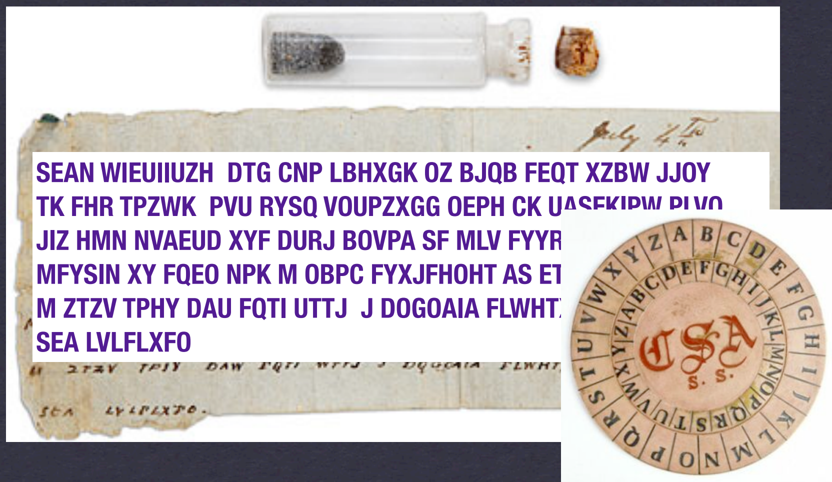
http://www.alertboot.com/blog/blogs/endpoint_security/archive/2010/12/29/data-encryption-bottled-civil-war-message-used-vigenere-cipher.aspx

An aside...



http://www.alertboot.com/blog/blogs/endpoint_security/archive/2010/12/29/data-encryption-bottled-civil-war-message-used-vigenere-cipher.aspx

An aside...

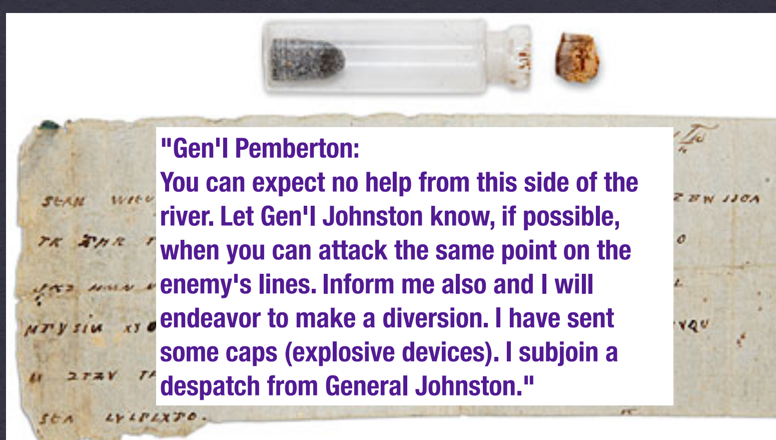


http://www.alertboot.com/blog/blogs/endpoint_security/archive/2010/12/29/data-encryption-bottled-civil-war-message-used-vigenere-cipher.aspx

An aside...

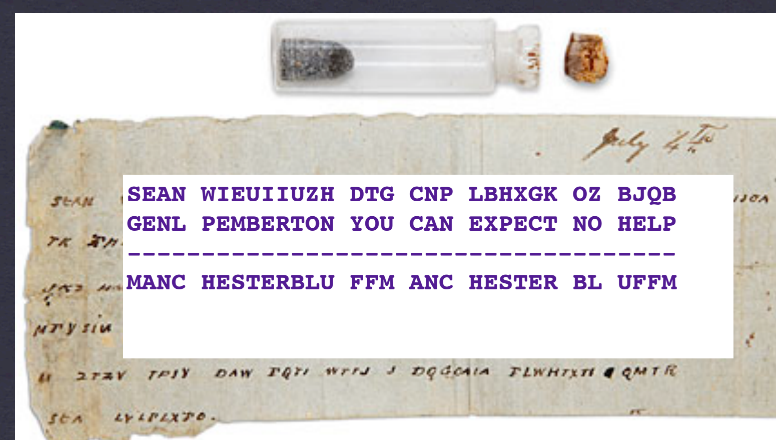
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Frequency of letters in ciphertext (%)	4	3	2	3	4	6	2	4	4	4	3	4	2	3	6	5	3	2	4	6	4	5	2	4	4	4
Frequency of letters in English (%)	8	2	3	4	13	2	2	6	7	0	1	4	2	7	8	2	0	6	6	9	3	1	2	0	2	0

An aside...



http://www.alertboot.com/blog/blogs/endpoint_security/archive/2010/12/29/data-encryption-bottled-civil-war-message-used-vigenere-cipher.aspx

An aside...



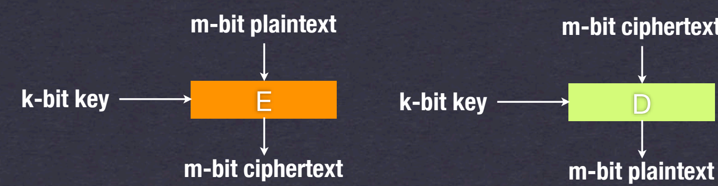
http://www.alertboot.com/blog/blogs/endpoint_security/archive/2010/12/29/data-encryption-bottled-civil-war-message-used-vigenere-cipher.aspx

Review

- Last time:
  - Historical tour of Cryptography
  - Classical Crypto, Mechanical ciphers
  - One Time Pads
  - <u>Block ciphers</u>
    - Modes of Operation (CBC & CTR)

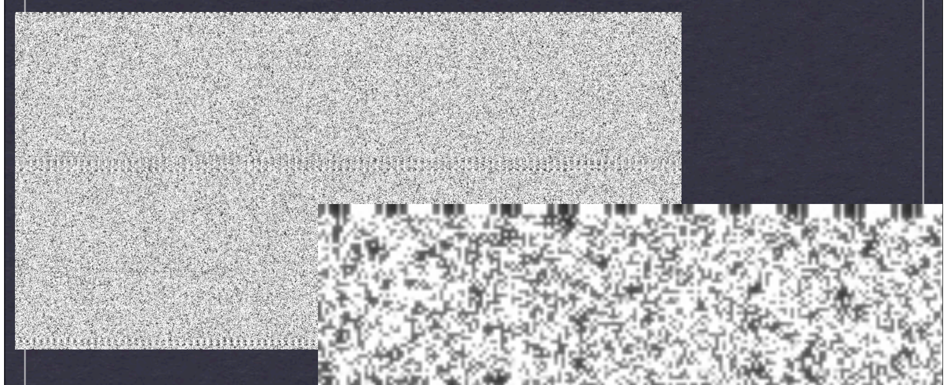
Review

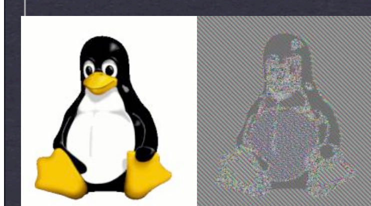
- <u>Review: Block Ciphers</u>
  - Keyed primitive
  - Operate on fixed-size input blocks
  - Invertible (pseudo-random permutation)
  - Examples: DES, AES...



Defining Security

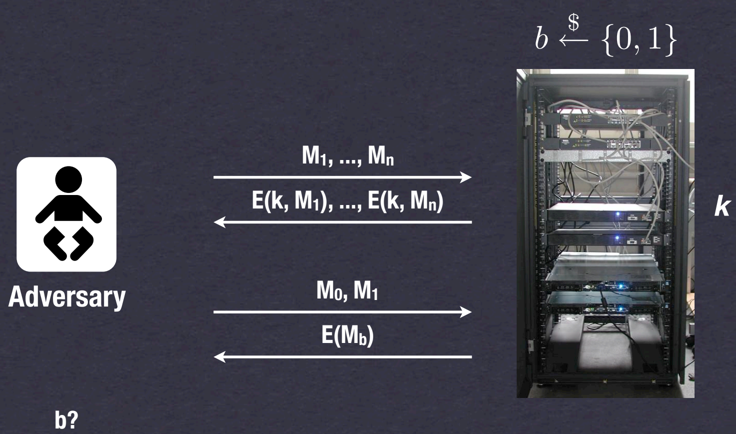
- What we'd like from our encryption scheme?
  - Ciphertext doesn't leak <u>*any*</u> info about plaintext
  - Even if Adversary knows a lot about the plaintext distribution
  - Even if Adversary can <u>choose</u> the plaintext distribution
  - Even if Adversary can obtain encryptions of chosen plaintexts





Review

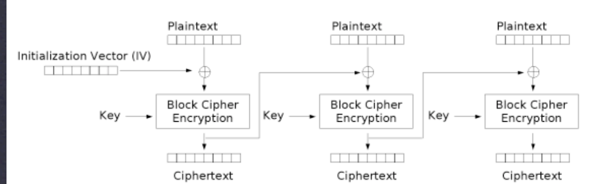
- Semantic Security (IND-CPA)



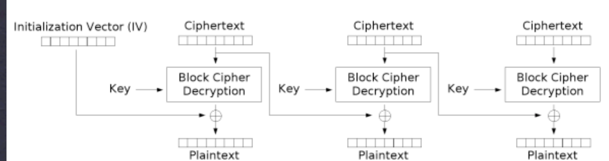
Using Block Ciphers

- ECB is not semantically secure, hence we use a "mode of operation"
  - e.g., CBC, CTR, CFB, OFB (and others)
- These provide:
  - Security for multi-block messages
  - Randomization (through an Initialization Vector)

CBC Mode



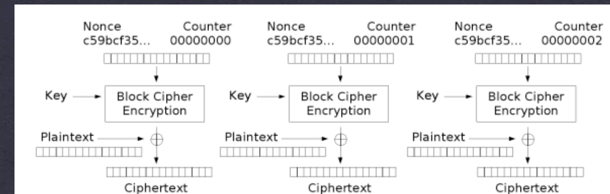
Cipher Block Chaining (CBC) mode encryption



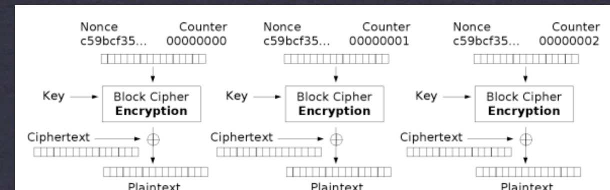
Cipher Block Chaining (CBC) mode decryption

Public domain image courtesy Wikipedia.

CTR Mode



Counter (CTR) mode encryption



Counter (CTR) mode decryption

Public domain image courtesy Wikipedia.

CTR Mode

- Intuition:
  - We're using the block cipher to generate a stream of "randomish" bits
  - XOR that stream of bits with the message
  - This should look familiar

CTR Mode

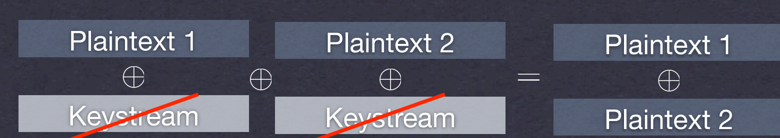
- CTR vs CBC
  - Unlike CBC, CTR can be parallelized (each block of the message encrypted separately by a different processor)
  - CTR keystream can be pre-computed
  - IV is much simpler

Security of CTR

- IND-CPA assuming secure block cipher (PRP)

Security of CTR

- IND-CPA assuming secure block cipher (PRP)
- However, counter range must never be re-used



- Similar example: MS Word 2003
  - (they used RC4, but same problem)

“Two-time pad”

- Re-using CTR IV range gives “two time pad”

= Plaintext 1 ⊕ Plaintext 2

- But can we tear these messages apart?
  - Answer: depends on the plaintexts
  - Venona project (hand-decode)
  - Modern NLP techniques (auto-decode)

Malleability

- The ability to modify a ciphertext
  - Such that the plaintext is meaningfully altered
  - CTR Mode (bad)
  - CBC Mode (somewhat bad)
- The solution:
  - Authenticated Encryption

Authenticated Encryption

MACs

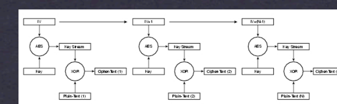
- Symmetric-key primitive
  - Given a key and a message, compute a “tag”
  - Tag can be verified using the same key
  - Any changes to the message detectable
- To prevent malleability:
  - Encrypt then MAC
  - Under separate keys

MACs

- Definitions of Security
  - Existential Unforgeability under Chosen Message Attack (EU-CMA)
- Examples:
  - HMAC (based on hash functions)
  - CMAC/CBC-MAC (block ciphers)

Authenticated Encryption

- Two ways to get there:
  - Generic composition
    Encrypt (e.g., CBC mode) then MAC
    - two different keys, multiple primitives
  - Authenticated mode of operation
    - Integrates both encryption & authentication
    - Single key, typically uses only one primitive (e.g., block cipher)
    - Ex: CCM, OCB, GCM modes

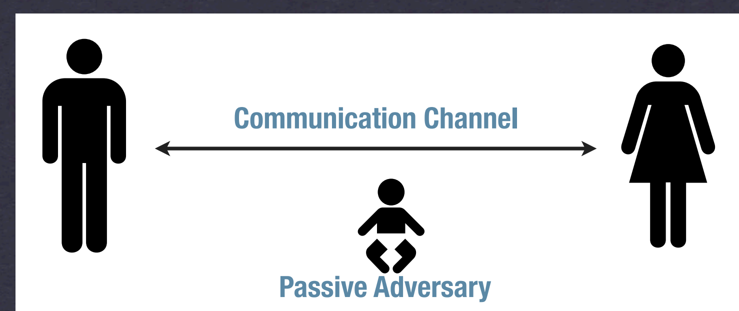


Asymmetric Crypto

- So far we've discussed symmetric crypto
  - Requires both parties to share a key
  - Key distribution is a hard problem!

Key Agreement

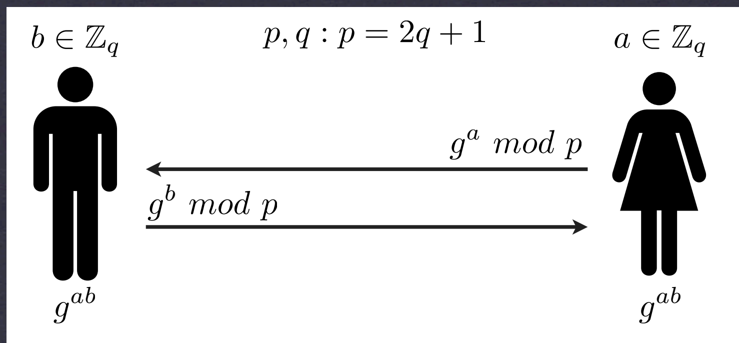
- Establish a shared key in the presence of a passive adversary



D-H Protocol

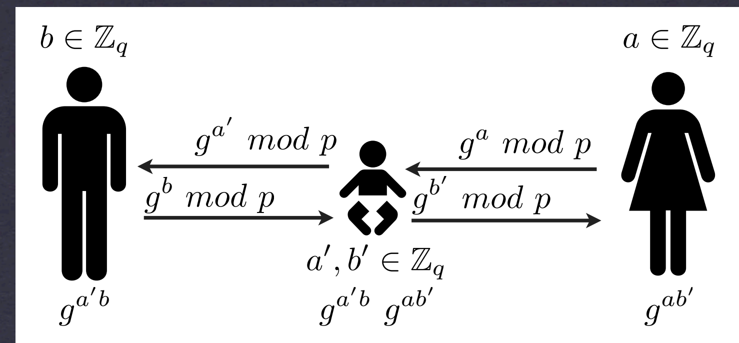
Malcolm Williamson in 72

Diffie-Hellman in 76



Man in the Middle

- Assume an <u>active</u> adversary:

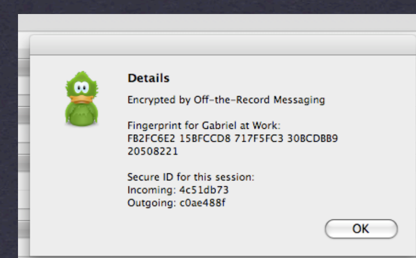


Man in the Middle

- Caused by lack of <u>authentication</u>
  - D-H lets us establish a shared key with anyone... but that's the problem...
- Solution: Authenticate the remote party

Preventing MITM

- Verify key via separate channel
- Password-based authentication
- Authentication via PKI

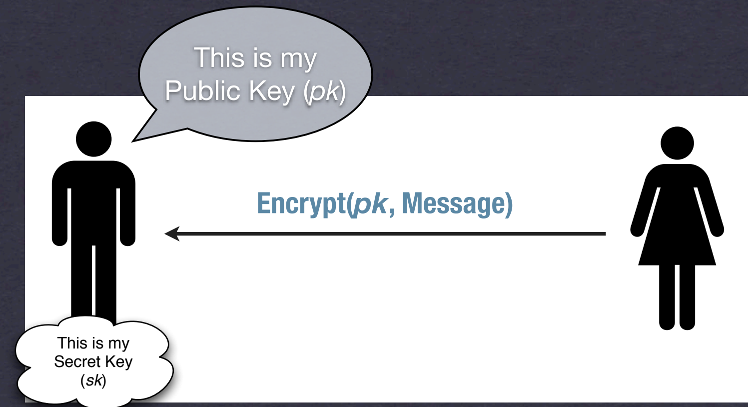


Public Key Encryption

Ellis in 72, Cocks a few months later

- What if our recipient is <u>offline?</u>
  - Key agreement protocols are interactiv
  - e.g., want to send an email

Public Key Encryption



RSA Cryptosystem

<u>Key Generation</u>

Choose large primes: p, q

$N = p \cdot q$

$\phi(N) = (p-1)(q-1)$

Choose:

$e : gcd(e, \phi(N)) = 1$

$d : ed \; mod \; \phi(N) = 1$

Output:

$pk = (e, N)$

$sk = d$

<u>Encryption</u>

$c = m^e \; mod \; N$

<u>Decryption</u>

$m = c^d \; mod \; N$

"Textbook RSA"

- In practice, we don't use Textbook RSA
  - Fully deterministic (not semantically secure)
  - <u>Malleable</u>

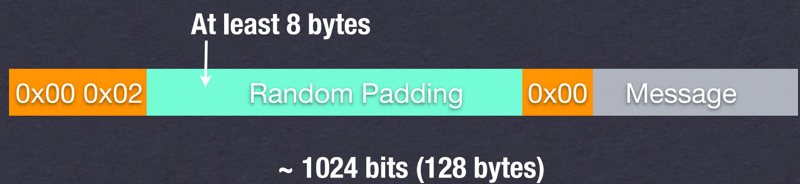
$$c' = c \cdot x^e \; mod \; N$$

$$c'^d = (m^e \cdot x^e)^d = m \cdot x \; mod \; N$$

  - Might be <u>partially</u> invertible
    - Coppersmith's attack: recover part of plaintext (when *m* and *e* are small)

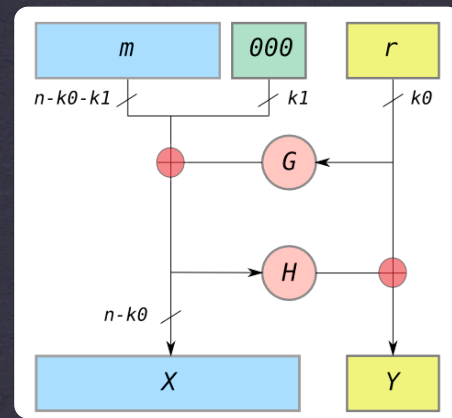
RSA Padding

- Early solution (RSA PKCS #1 v1.5):
  - Add “padding” to the message before encryption
  - Includes randomness
  - Defined structure to mitigate malleability
  - PKCS #1 v1.5 badly broken (Bleichenbacher)



RSA Padding

- Better solution (RSA-OAEP):
  - G and H are hash functions



Efficiency

	Cycles/Byte
AES (128 bit key)	18
DES (56 bit key)	51
RSA (1024 bit key) Encryption	1,016
RSA (1024 bit key) Decryption	21,719

$m^e \ mod \ N$
$e = 65,537$

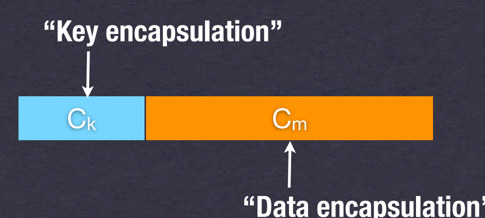
$m^d \ mod \ N$

Benchmarks from: http://www.cryptopp.com/benchmarks.html
Microsoft Visual C++, Windows XP, Intel Core 2 1.83Mhz in 32-bit mode

Hybrid Encryption

- Mixed Approach
  - Use PK encryption to encrypt a symmetric key
  - Use (fast) symmetric encryption on data

$$k \xleftarrow{\$} \{0,1\}^k$$
$$C_k \leftarrow RSA.Encrypt_{pk}(k)$$
$$C_m \leftarrow AES.Encrypt_k(message)$$



Key Strength

Level	Protection	Symmetric	Asymmetric	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash
1	Attacks in "real-time" by individuals *Only acceptable for authentication tag size*	32	-	-	-	-	-
2	Very short-term protection against small organizations *Should not be used for confidentiality in new systems*	64	816	128	816	128	128
3	Short-term protection against medium organizations, medium-term protection against small organizations	72	1008	144	1008	144	144
4	Very short-term protection against agencies, long-term protection against small organizations *Smallest general-purpose level,* *Use of 2-key 3DES restricted to 2^{40} plaintext/ciphertexts,* *protection from 2009 to 2011*	80	1248	160	1248	160	160
5	Legacy standard level *Use of 2-key 3DES restricted to 10^{6} plaintext/ciphertexts,* *protection from 2009 to 2018*	96	1776	192	1776	192	192
6	Medium-term protection *Use of 3-key 3DES,* *protection from 2009 to 2028*	112	2432	224	2432	224	224
7	Long-term protection *Generic application-independent recommendation,* *protection from 2009 to 2038*	128	3248	256	3248	256	256
8	"Foreseeable future" *Good protection against quantum computers*	256	15424	512	15424	512	512

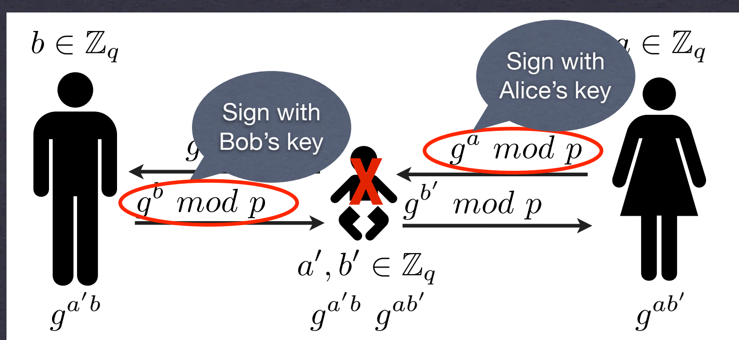
Source: www.keystrength.com (BlueKrypt). Based on ECrypt recommendations.

Digital Signatures

- Similar to MACs, with public keys
  - Secret key used to sign data
  - Public key can verify signature
  - Advantages over MACs?

Preventing MitM

- Assume an active adversary:



PKI & Certificates

- How do I know to trust your public key?
  - Put it into a file with some other info, and get someone else to sign it!

Next Time

- Protocols & Implementation
- Reading!
- A1 coming up in 1 week