Mathematical Cryptology 2013

Dept. of EIT
Lund University
Box 118, 221 00 Lund, Sweden

WEEK 4,

Reading instructions:

- Chapter 14, 15
- Chapter 16 (read through)

There will be 4 hours of lectures covering the above.

Hand-in exercises:

1) Review exercises: 14.1-14.6, 15.1-15.6

2) Standard exercises: 14.7, 15.7-15.10

3) Programming exercise: (from before: 2.12 (maximum 5 hours))

Hand in no later than **March 1, 2013.**

Ch 14: Review exercises:
14.1 Explain the man in the middle attack on the Diffie-Hellman key exchange algorithm.
14.2 Why do digital signature algorithms solve the problem raised by the man in the middle attack?
14.3 Why does a MAC not provide non-repudiation?
14.4 Describe the DSA signature algorithm, and describe where the security comes from.
14.5 What are the advantages of Schnorr signatures over DSA?
14.6 Explsin the similarities and differences between the Diffie-Hellman protocol and the MQV protocol.

Ch 14: Standard exercises:
14.7 Show that if a user uses the same ephemeral key k to sign two different messages in the DSA algorithm then an attacker can recover their long term private key.

Ch 15: Review exercises:
15.1 What is the maximum and average number of multiplications in the basic binary exponentiation algorithm?
15.2 Why is the smallest possible Hamming weight of an RSA public exponent equal to two?
15.3 What stops one using signed exponentiation techniquest in an RSA implementation?
15.4 Why is Karatsuba multiplication rarely used in an RSA implementation?
15.5 What is meant by Montgomery representation and why does it offer advantages in systems implementing algorithms such as RSA?
15.6 In algorithms for finite fields of characteristic two, why does one use an analogue of Karastsuba multiplication and not Montgomery multiplication?

Ch 15: Standard exercises:
15.7 Assume modular multiplication of a k-bit number requires k^2 operations. How much faster is two 512 512-bit modular exponentiations by 512-bit exponents, compared to a single 1024-bit modular exponentiation by a 1024-bit exponent? Conclude that the CRT method for RSA decryption and signing is more efficient.

15.8 Suppose a device uses CRT to speed up RSA decryption or signatures, i.e. it computes

$$m_p = c^{d \pmod{p-1}} \pmod{p},$$

$$m_q = c^{d \pmod{q-1}} \pmod{q},$$

and then computes m from m_p and m_q cia CRT. However, an attacker manages to get the device to compute m_p incorrectly. Show that if m_q is still computed correctly then the attacker can use this broken device to recover the private key.
15.9 Show that the RSA encryption and decryption algorithms can be implemented in $O(n^3)$-bit operations, where n is the bit length of the modulus N.
15.10 Show that ElGamal encryption requires about $2 \log p$ multiplications modulo p. Hence deduce its bit complexity is $O((\log p)^3)$.