McGill

DECEMBER 2007
Final Examination

FINAL EXAMINATION

Computer Science COMP-547A
Cryptography and Data Security

10 DECEMBER 2007, 14h00

Examiner:	Prof. Claude Crépeau	Assoc Examiner:	Prof. David Avis

INSTRUCTIONS:

• This examination is worth 50% of your final grade.

• The total of all questions is 100 points.

• Each question heading contains (in parenthesis) a list of values for each sub-questions.

• This is an **open book** exam. **All documentation is permitted.**

• Faculty standard calculator permitted only.

• The exam consists of 5 questions on 3 pages, title page included.

Suggestion:

read all the questions and their values before you start.

Question 1. Entropy (5+5+5 points)

Consider a random variable X with 4 possible outcomes: "0" with probability ¼, "1" with probability ¼, "2" with probability 1/8 and "3" with probability 3/8.

• Compute H(X), the entropy of X. (you may express your answer in terms of $\tau = \log_2 3$)

• Give another distribution Y on {0,1,2,3} such that H(Y)=H(X).

• Compute H(X mod 2) and H(Y mod 2).

Question 2. Short and *Sweet* (5+5+5+5+5 points)

(justify briefly your answers)

(a)
Explain the relevance of large prime numbers to public-key cryptography.

(b)
Given an **RSA** public-key (n,e), is the problem of finding d such that $e \times d \bmod \phi(n) = 1$ equivalent to the problem of factoring n ?

(c)
Name a crypto-system in which the following operation is relevant:
(multiplicative) inversion of an element in the field of 256 elements.

(d)
Identify the 13 finite fields with a number of elements between 100 and 150.

(e)
What is the advantage of combining a cryptographic hash function (message digest) together with a digital signature scheme ?

Question 3. AES PRBG (8+5 points)

Explain two ways of constructing pseudo-random bit generators from AES:

• In a first construction favor efficiency making sure the AES function is used only t times to produce $t \times 128$ pseudo-random bits. Discuss the impact of the AES key size on efficiency and security.

• In a second construction, favor security by making sure your PRBG is as secure as the AES function. (Assuming AES is a one-way permutation)

Question 4. ElGammal (10+5+6+6 points)

(A) Double ElGammal signature

Let (p,α,β,β') be a set of **ElGammal** public-keys. Let (a,a') be a pair of **ElGammal** private keys such that $\beta=\alpha^a \bmod p$ and $\beta'=\alpha^{a'} \bmod p$. Consider the **DEG (double-ElGammal)** signature scheme of a message m to be $\mathbf{DEG}(m) := [\ (\gamma,\delta), (\gamma',\delta')\ ]$ where everything is computed the standard way but for both sets of parameters.

• Analyze the impact of this improved way of signing messages on the (2) known existential-forgery attacks on **ElGammal** signatures.

(B) ElGammal PKC is multiplicative

Let (p,α,β,a) be a set of **ElGammal** public/private-keys. Let (y_1,y_2) be the **ElGammal** encryption of an unknown message x. Let (y'_1,y'_2) be the **ElGammal** encryption of another message z.

• Show how a valid encryption of the message $xz \bmod p$ can be obtained from the encryptions of x and z. Explain how this is similar to the multiplicative property of RSA and its significance.

• Argue that the $lsb(x)$ <u>cannot be easy</u> to compute from an ElGammal encryption of x when the Computational Diffie-Hellman problem is <u>hard</u> to solve.

• Consider a variation on this encryption scheme where the encryption of x is performed as $\gamma=x+\beta^k \bmod p$ instead of $\gamma=x\times\beta^k \bmod p$. Can this change the security of the system ? Is it now possible that the $lsb(x)$ <u>be easy</u> to compute from such an encryption of x ?

Question 5.  MACs (8+6+6 points)

<u>**NOTE**: all the questions below are NOT about the inner structure of SHA-1.</u>

• Explain the design principles leading to HMAC. In particular, clarify why *ipad* and *opad* must be distinct constants.

• The search for collisions in SHA-1 is very active and it seems very likely that existential collisions on SHA-1 will be found in the near future (if not already!). Explain why such collisions have very little impact on the security of HMAC.

• Consider a notion of *public-key* MAC: for an arbitrary message m, and a public-key encryption system (e_{pk}, d_{pk}), let $(\ m, \mathrm{HMAC}_k(m), e_{pk}(k)\ )$ be a public-key MAC of m using a random key k. Upon reception of $(\ a, b, c\ )$ the validity of the message is checked by computing $k':=d_{pk}(c)$, and verifying $\mathrm{HMAC}_{k'}(a) = b$. A public-key MAC should be tamper resistant. What is wrong with the proposed implementation ?