

University of Waterloo
Department of C&O

PhD Comprehensive Examination in Cryptography
Summer 2016
Examiners: D. Jao and A. Menezes

June 13, 2016
1:30 pm — 4:30 pm
MC 6486

Instructions

- Answer as many questions as you can.
- You are *not* expected to answer all 8 questions.
- Complete answers are preferred over fragmented ones.
- Some questions may require additional assumptions, such as complexity-theoretic assumptions. State any additional assumptions that you require.
- Justify all answers.

Questions

1. Hash functions

In the triangle of the three properties of a hash function:

- collision resistant
- preimage resistant
- second-preimage resistant

enter six symbols $\in \{\implies, \not\implies\}$ to indicate which property implies the other and which does not.

Prove **three** out of the six directions.

2. Elementary number theory

Suppose that $p = 2^{2^k} + 1$ is prime, where $k \geq 1$.

- Prove that any quadratic nonresidue modulo p is a generator of \mathbb{F}_p^* .
- Hence show that 7 is a generator of \mathbb{F}_p^* .

3. Elementary number theory

Let n be an RSA modulus. Does n always, sometimes, or never have a primitive root? (Recall that a primitive root modulo n is an element of order $\phi(n)$ in the multiplicative group of units \mathbb{Z}_n^* .)

4. RSA

Suppose that textbook RSA is used to encrypt a random 56-bit DES key k without padding; that is, the value of k as an integer is used as an RSA plaintext. Given the corresponding RSA ciphertext, give a (classical) algorithm that, with high probability, recovers the key k in substantially fewer than 2^{56} operations.

Hint: Use the fact that a random integer 56-bit integer factors into a product of two integers less than 2^{29} with high probability.

5. Discrete logarithm problem

Let $p = 2^{2^k} + 1$ be a prime number. Describe and analyze a polynomial-time algorithm for solving the discrete logarithm problem in \mathbb{Z}_p^* . (Recall that the DLP in \mathbb{Z}_p^* is the following: given p , a generator g of \mathbb{Z}_p^* , and $h \in \mathbb{Z}_p^*$, find the integer $\ell \in [0, p-2]$ such that $h = g^\ell \pmod{p}$.)

6. Message Authentication Codes

Recall the definition of CBC-MAC:

Algorithm 1 CBC-MAC

Input: An n -block message $x = x_1 || \dots || x_n$ and a secret key k .

- 1: $IV \leftarrow 00 \dots 0$
- 2: $y_0 \leftarrow IV$
- 3: **for** $i \leftarrow 1$ to n **do**
- 4: $y_i \leftarrow \text{Encrypt}(k, y_{i-1} \oplus x_i)$
- 5: **end for**

Output: Tag y_n

- (a) Is CBC-MAC with one-block inputs existentially unforgeable under a chosen-message attack (EUF-CMA)?
- (b) Is CBC-MAC with variable-length inputs existentially unforgeable under a chosen-message attack (EUF-CMA)?

7. Identification schemes

Let G be a cyclic group of prime order p with generator g . Suppose the verifier is given $\beta = g^\alpha$ for some randomly selected $\alpha \in \mathbb{Z}_p$. Consider the zero-knowledge proof of knowledge of α in Figure 1.

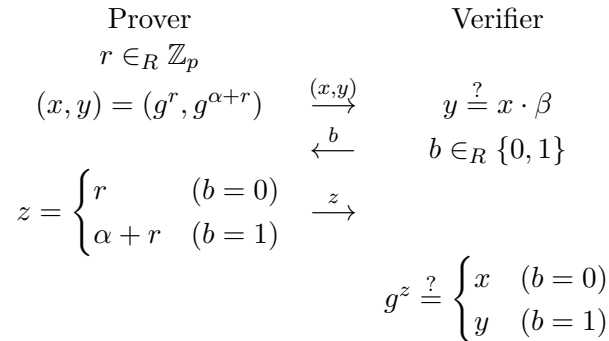


Figure 1: Zero-knowledge proof of knowledge of α

- (a) Show that the proof is zero-knowledge for an honest verifier.
- (b) Show that a cheating prover can succeed with probability $1/2$.
- (c) Describe how to modify the protocol so that the prover's cheating probability is reduced to negligible levels.

8. Provable security

Consider the Zheng-Seberry public-key encryption scheme (1993):

Public parameters: A cyclic group G , a generator g of G , and two random oracles

$$\begin{aligned}H_1: \{0, 1\}^t &\rightarrow \{0, 1\}^n \\H_2: G &\rightarrow \{0, 1\}^{t+n}.\end{aligned}$$

Key generation: Choose a private key $x \in \mathbb{Z}$. The corresponding public key is $h = g^x$.

Encryption: To encrypt $m \in \{0, 1\}^t$, choose $y \in \mathbb{Z}$ and compute

$$\begin{aligned}Y &= g^y \\c &= H_2(h^y) \oplus (m || H_1(m)).\end{aligned}$$

The ciphertext is (Y, c) .

Decryption: Compute $c \oplus H_2(Y^x)$. If the leftmost t bits of the result map to the rightmost n bits under H_1 , then output the leftmost t bits; otherwise output NULL.

Show that the Zheng-Seberry scheme is not **IND-CCA2**.

(IND-CCA2 means “indistinguishable against adaptive chosen-ciphertext attack”. In this attack, the adversary selects two plaintexts m_0, m_1 , is then given the encryption c of m_b (where $b \in_R \{0, 1\}$), and has to determine b with probability significantly greater than $\frac{1}{2}$. The adversary is also given access to a decryption oracle to which it can present any ciphertext for decryption except for the challenge ciphertext c itself.)