Quiz 1 Solutions

- **DO NOT OPEN** this quiz until instructed to do so.
- You should not have more than one empty chair between you and the next person. If seating availability permits, do not sit directly next to another person.
- This quiz is **open book**. You may use any of the results presented in class, in the handouts, or in the problem sets.
- There are fifteen (15) problems totaling 100 points. Problems are labelled with their point values.
- Put your name on the top of **every** page – *these pages may be separated for grading.*
- Write your solutions in the space provided. Should you need extra space, write on the back of the sheet containing the question.
- **Be neat and write legibly**. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it.

Problem Q1-1. [4 pts]

Fill in your name and the names of the people sitting next to you. If you are at the end of a row, write $\perp$ in the space provided.

Your name:	Solution
Name of person to your right:	$\perp$
Name of person to your left:	$\perp$

DO NOT WRITE ON THIS PAGE -OK!

Problem	Grade	Points
Q1-1		4
Q1-2		4
Q1-3		4
Q1-4		7
Q1-5		12
Q1-6		14
Q1-7		5
Q1-8		4
Q1-9		5
Q1-10		3
Q1-11		7
Q1-12		13
Q1-13		6
Q1-14		4
Q1-15		8
Total		100

Problem Q1-2. [4 pts]

For a parallel computer (which can do many operations simultaneously) programmed to perform CBC mode encryption (circle the correct answer):

1 Encryption is faster than decryption.

[✓ 2] Decryption is faster than encryption.

3 Encryption and decryption should run in approximately the same time.

Solution Note: For encryption, you can only compute ciphertext block C_i when you have computed ciphertext block C_{i-1}. For decryption, you can parallelize the computations of all plaintext blocks M_i's since you know all ciphertext blocks C_i's.

Problem Q1-3. [4 pts]

Circle true or false for the following statements. If $\mathcal{P} = \mathcal{NP}$, then:

True [✓ **False**] The one-time pad still provides information-theoretically secure message authentication.

Solution Note: The one-time pad provides information-theoretically secure message *encryption.*

True [✓ **False**] Secure encryption becomes impossible.

Solution Note: Secure encryption is *still* possible through one-time pads for instance.

True [✓ **False**] Shamir's secret-sharing technique becomes insecure.

Solution Note: Shamir's secret-sharing technique is not only computationally secure, but *information-theoretically* secure, so remains secure *even if* $\mathcal{P} = \mathcal{NP}$.

[✓ **True**] **False** One-way functions do not exist.

Solution Note: One-way functions are by definition easy to compute and hard to invert. For a one-way function f, even if it is hard to guess an inverse x of a given y (so that $f(x) = y$), it is easy to check whether a given x' is actually an inverse of y (as f is easy to compute). $\mathcal{P} = \mathcal{NP}$ means that any problem which is easy to check would also be easy to guess. Therefore, one-way functions can't exist if $\mathcal{P} = \mathcal{NP}$.

Problem Q1-4. [7 pts]

Circle true or false for the following statements:

[✓ **True**] **False** Alma Whitten's experiments show that PGP 5.0's graphical user interface is not sufficiently effective to provide security for most users.

Solution Note: cf. Handout 2.

True [✓ **False**] The WSJ cookie authentication scheme was insecure because of sequential session IDs.

Solution Note: The WSJ used an insecure message authentication code.

True [✓ **False**] A cryptographically secure hash function $h : \Sigma^* \rightarrow \Sigma^k$ (OW, CR) must be injective.

Solution Note: A function mapping an infinite number of inputs to a finite number of outputs cannot be injective.

True [✓ **False**] Triple-DES uses uses three unique 56-bit DES keys.

Solution Note: We decided to remove this question from the quiz. In class, we explained that Triple-DES uses two keys, but there are some versions that use three keys too.

[✓ **True**] **False** Consecutive Fibonacci numbers are the worst-case input for Euclid's Algorithm.

Solution Note: cf. Lecture 06 or CLR page 859 (Th 31.11).

True [✓ **False**] The El Gamal encryption scheme is plaintext-aware.

Solution Note: cf. Lecture 11.

True [✓ **False**] To make a deterministic public-key encryption scheme secure against an adaptive chosen ciphertext attack, it suffices to pad the given plaintext with some random bits before encryption (such random bits being discarded upon decryption).

Solution Note: Suppose for instance that you use RSA and pad the message in the following manner: $\boxed{\quad m \quad}\boxed{\quad r \quad}$.
Then the plaintext you actually encrypt is of the form $M = (2^N m + r) \bmod n$, and the target ciphertext you want to decrypt would be $M^e \bmod n = (2^N m + r)^e \bmod n$.
By querying $2^e(2^N m + r)^e \bmod n$ to the decryption oracle (remember e is public), you get $M' = 2.(2^N m + r) \bmod n$.
Now you can retrieve $(2^N m + r) \bmod n$ by multiplying M' by the modular inverse of 2 modulo n, namely $\frac{n+1}{2}$.
Finally, you can then recover the target message m by discarding r.

Problem Q1-5. [12 pts]

Consider the following generalization of Lamport's one-time signature scheme, for signing a value m, where m is drawn from a finite set $\{1, 2, \ldots, t\}$ for some $t > 2$.

The use-once portion of the key used to sign m consists of two values x_0 and y_0. Here x_0 and y_0 are the roots of hash chains of length $t + 1$. That is, $x_i = h(x_{i+1})$ for $0 \leq i < t$ and $y_i = h(y_{i+1})$ for $0 \leq i < t$, where h is a one-way hash function.

To sign m, where $1 \leq m \leq t$, the signer reveals both $X = x_m$ and $Y = y_{t-m}$. The signature can be verified by checking that $h^m(X) = x_0$ and $h^{t-m}(Y) = y_0$.

$$\begin{tikzcd}
x_0 & x_1 \arrow[l, "h"'] & \cdots \arrow[l, "h"'] & |[draw, circle]| x_m \arrow[l, "h"'] & \cdots \arrow[l, "h"'] & x_{t-m} \arrow[l, "h"'] & \cdots \arrow[l, "h"'] & x_{t-1} \arrow[l, "h"'] & x_t \arrow[l, "h"'] \\
y_0 & y_1 \arrow[l, "h"'] & \cdots \arrow[l, "h"'] & y_m \arrow[l, "h"'] & \cdots \arrow[l, "h"'] & |[draw, circle]| y_{t-m} \arrow[l, "h"'] & \cdots \arrow[l, "h"'] & y_{t-1} \arrow[l, "h"'] & y_t \arrow[l, "h"']
\end{tikzcd}$$

(a) **[6 pts]** Why are two chains used per value signed? (Why not eliminate the y chain?)

Solution: Without the y chain, when the signer signs $m > 1$ by publishing x_m, then an adversary can forge the signature for $m - 1$: $x_{m-1} = h(x_m)$.

(b) **[6 pts]** Argue briefly that this scheme is secure, if h is indeed one-way. (Why can't a forger produce a signature for a different value m', after having seen the signature for m?)

Solution: Suppose this scheme were not secure.
This would mean that after having the signature (x_m, y_{t-m}) for m, an adversary could produce a signature $(x_{m'}, y_{t-m'})$ for $m' \neq m$. Note that the adversary cannot know more than one signature with the same public key (x_0, y_0), since this is a one-time signature scheme.

1) If $m' > m$, then this means you are able to compute among other values, an h-inverse x' for x_m: $x' = x_{m+1} = h^{m'-m-1}(x_{m'})$.
2) Else $m' < m$, ane this means you are able to compute among other values, an h-inverse y' for y_{t-m}: $y' = y_{t-m+1} = h^{m-m'-1}(y_{t-m'})$.

In any case, this contradicts the fact that h is one-way.

□

Problem Q1-6. [14 pts]

(a) [4 pts]

Recall that the WSJ used crypt() in its MAC, $\text{MAC}_k = \text{crypt(username}||\text{secret)}$ where $||$ denotes concatenation. Assume that the secret can be any sequence of 8-bit (not necessarily printable) characters. Give the maximum number of Web queries an interrogative adversary must make to achieve a total break (universal forgery).

Solution: We expected the answer 2048 based on the attack presented in lecture and in a handout. We accepted the answer of 1024 when justified that crypt does not really take 8-bit inputs, it takes 7-bit inputs. A couple students received full credit for describing a new attack that takes just 8 queries. Each character can be guessed offline. One makes 8 fake accounts, one for each username length 7 down to 0. Starting with the 7-character username, you can brute force the first character of the pad offline (256 offline guesses) with the same attack code. One continues which each username length.

(b) [4 pts]

The Backstreet Journal, a new branch of the WSJ catering to aging pop-star financial news, decided to use a cryptographically secure (OW, CR) hash function $h : \{0, \ldots, 255\}^k \rightarrow \{0, \ldots, 255\}^{20}$ instead of crypt() in MAC_k. Similar to crypt(), the h function truncates input after the kth byte. Give the maximum number of Web queries an interrogative adversary must make to achieve a total break (universal forgery) if the secret is any sequence of 8-bit (not necessarily printable) characters. You can assume that usernames can be any length.

Solution: This is a generalized version of part (a). The solution is $256 * k$, but we also accepted k when justified with the offline attack described in the solution to part (a).

(c) [6 pts]

If the WSJ had used SHA-1 instead of crypt() in its MAC, would you expect the scheme to be stronger? Why or why not?

Solution: We expect that the new scheme would be no weaker than the current scheme, and perhaps even stronger. One reason to suspect that the scheme is stronger is that the old dynamic programming attack no longer works. The best attack we know of is to guess the entire secret at once, rather than character by character. This is exponential, rather than linear, in the size of the secret. Because SHA-1 does not truncate input, our old attack does not work.

Problem Q1-7. [5 pts]

For each of the following applications, list the necessary hash function properties (OW, CR, WCR):

PGP fingerprints	CR/WCR
Unix password files	OW
Secure URLs	CR/WCR
Hash cash	OW
One-time passwords	OW

> *Solution Note:* PGP fingerprints and secure URLs are very similar. Since the public key or web page being hashed is public, one-wayness is not necessary. We accepted both CR and WCR because it depends on the adversarial model. For the remaining applications, we did not accept CR or WCR. It's OK to have multiple passwords hashing to the same value as long as it's one-way. Hash cash and one-time passwords rely on the difficulty of inverting.

Problem Q1-8. [4 pts]

Ben Bitdiddle upgraded his plaintext telnet server to a telnet server with one-time passwords based on the Lamport password authentication scheme. Which of the following attacks is Ben's new system no longer or less susceptible to (circle all that apply):

[✓ **1**] Replay attack

2 Session hijacking

[✓ **3**] Dictionary attack on stolen database

4 Keystroke logging

> *Solution Note:* We accepted the replay attack because simple replays of one-time passwords will not work. We can still hijack a session because there is no session authentication. A dictionary attack should become more difficult because the one-time passwords stored in the database should be uniformly random. Keystroke logging can still capture the initial secret used to generate one-time passwords. One can still conduct surveillance by capturing all keystrokes.

Problem Q1-9. [5 pts]

In the list below, circle the symmetric block ciphers:

[✓ AES]	[✓ DES]	DSA/DSS	El Gamal	RC4
[✓ RC5]	RC Cola$^{\text{TM}}$	[✓ Rijndael]	RSA	Triple-CBC

Solution Note: Triple-DES is a symmetric block cipher, but Triple-CBC does *not* exist. DSA/DSS, El Gamal, and RSA are asymmetric algorithms. RC4 is a stream cipher. RC Cola$^{\text{TM}}$ is one of the TA's favorite soft drinks.

Problem Q1-10. [3 pts]

Name one cipher from previous question that is a Feistel cipher:

Solution: DES is the only Feistel cipher among the above.

Problem Q1-11. [7 pts]

In the Digitarian World, people don't have names, but numbers to identity themselves. A group of four students (12, 25, 20, 5) attending the university 13-9-20 is taking 6.857. They are having some issues trying to do problem set 3 problem 1: they just can't find a large prime p such that all their numbers are generators of $\mathcal{Z}_p^*$.

Explain briefly why they could not succeed.

Solution: We are proving in the following that (12, 25, 20, 5) cannot be generators of $\mathcal{Z}_p^*$ at the same time for *any* prime p.

1) If $p = 2$, then both 12 and 20 would be 0 mod 2, and would not be in $\mathcal{Z}_2^*$.
2) Else, p is odd, and $(p-1)$ is even.
   If 5 generates $\mathcal{Z}_p^*$, then its order is $(p-1)$. Then the order of $25 = 5^2$ is $(p-1)/2$ (as $p-1$ is even), and thus 25 cannot be a generator of $\mathcal{Z}_p^*$.

□

We gave partial credit to students who only dealt with the case of Sophie Germain primes.

Problem Q1-12. [13 pts]

Let p be a prime, and $g \in \mathcal{Z}_p^*$ be an element of order q, where q is a prime ≥ 3 (note that g is *not* a generator of $\mathcal{Z}_p^*$).

(a) [**5 pts**] What are valid formulas for the inverse of g modulo p? Circle all correct answer(s).

[✓ $g^{q-1} \bmod p$] $g^q \bmod p$ [✓ $g^{p-2} \bmod p$] $g^{p-1} \bmod p$ $g^p \bmod p$

$g^{q-1} \bmod q$ $g^q \bmod q$ $g^{p-2} \bmod q$ $g^{p-1} \bmod q$ $g^p \bmod q$

Solution Note: We have $g^q \bmod p = g^{p-1} \bmod p = 1$.
So that $g^p \bmod p = g^{p-1}.g \bmod p = g \bmod p$.
The two circled formulas are correct, since: $\begin{cases} g^{q-1}.g \bmod p = g^q \bmod p = 1 \\ g^{p-2}.g \bmod p = g^{p-1} \bmod p = 1 \end{cases}$
The formulas modulo q do not make sense, since we are working in $\mathcal{Z}_p$.

(b) [**4 pts**] Give a formula for the square root of g modulo p.

Solution: $\boxed{g^{\frac{q+1}{2}} \bmod p}$

Indeed,
$$\begin{aligned} (g^{\frac{q+1}{2}})^2 \bmod p &= g^{\frac{q+1}{2}.2} \bmod p \\ &= g^{q+1} \bmod p \\ &= g^q.g \bmod p \\ &= g \bmod p \end{aligned}$$

□

(c) [**4 pts**] For an integer $e \geq 3$ such that $gcd(e,q) = 1$, explain briefly how to compute the e^{th} root of g modulo p, *i.e.* find an h such that $h^e = g \pmod{p}$.

Hint: You may find some inspiration by looking at the RSA encryption/decryption process.

Solution: Using Euclid extended algorithm, we find the modular inverse d of e modulo q[1]. Then $\exists\lambda$, $e.d = \lambda.q + 1$.
The e^{th} root or g modulo p is $\boxed{h = g^d \bmod q}$.

Indeed,
$$\begin{aligned} h^e \bmod p &= (g^d \bmod p)^e \bmod p &&= (g^d)^e \bmod p &&= g^{d.e} \bmod p \\ &= g^{\lambda.q+1} \bmod p &&= (g^q)^\lambda.g \bmod p &&= 1^\lambda.g \bmod p \\ &= g \bmod p \end{aligned}$$

□

[1] Finding the modular inverse modulo $p-1$ works too, since $g^{p-1} \bmod p = 1$.

Problem Q1-13. [6 pts]

In the RSA scheme, the modulus $n = pq$ is chosen as a product of two large primes $p < q$. To make factoring n as hard as possible, Ben Bitdiddle decides to make the smaller prime p as large as possible, and thus chooses p and q as consecutive primes.

Explain briefly why Ben's approach is flawed. You can assume that p and q are reasonably close to each other.

Solution: p and q being consecutive primes, and reasonably close, they are $\approx \sqrt{n}$. More precisely, p is the largest prime $< \sqrt{n}$ and q is the smallest prime $> \sqrt{n}$. So, an adversay just needs, for instance, to try all odd integers t from $\sqrt{n}$ down, and test whether $n \bmod t = 0$. He will stop at the the first prime he hits and get p.

Problem Q1-14. [4 pts]

Ben Bitdiddle is using Shamir's (k, n) threshold secret sharing scheme, where n persons want to share a secret of N-bits, so that the shares of k persons are needed to reveal the secret. Ben chooses the prime p to be $(N+1)$-bits.

What is the approximate size (in bits) of each person's share? Circle the correct answer:

$N\frac{1}{k}$ $N\frac{1}{n}$ $N\frac{k}{n}$ $N\frac{n}{k}$ [✓ N]

Solution: All shares are of the form $q(i)$ where q is a polynomial (of degree $k-1$) with coefficients in $\mathcal{Z}_p$. All shares are therefore elements of $\mathcal{Z}_p$. p being of size $N+1$, all shares are of size $N+1$.

Problem Q1-15. [8 pts]

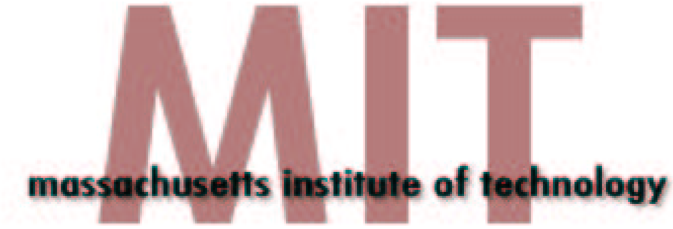

Figure 1: Plaintext picture.

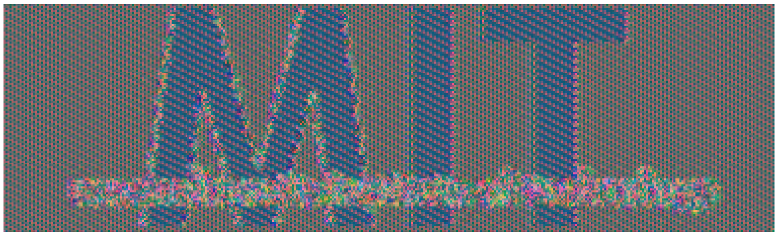

Figure 2: Encrypted picture.

Figure 2 is an encrypted version of Figure 1. The picture was encrypted with DES. The graphic format is very simple. It consists of a sequence of RGB values (ranging from 0 to 255). Each pixel takes three bytes (one for each color). The dimensions of the graphic is known a priori (390 × 115 pixels). In the binary file, the $(3(x + 390y))$th byte denotes the red color of the pixel at location (x, y). A similar formula describes the location of the green and blue colors of pixels. What block cipher block mode did we use to encrypt this graphic? Explain your reasoning.

> *Solution:* The solution is Electronic Codebook (ECB) mode because ECB preserves in the ciphertext the patterns of the plaintext. Note that DES takes 8-byte blocks as plaintext input and each pixel takes 3 bytes of storage. Therefore patterns will develop when 8 pixels in a row (24 bytes or 3 DES blocks) are repeated in the plaintext on 24-byte boundaries. The letters in "massachusetts institute of technology" are blurred because there is not a noticeable repeated pattern of 8 pixels on 24-byte boundaries. Had we used AES, the block size would have been twice as large and the picture would have likely been even more blurred. CBC and counter modes do not preserve plaintext patterns.