hence it is a [illegible]

– Ameya Prabhu, 201402004

1) (a) 1) Key generation algorithm → It chooses $k \in K$ uniformly at random given the length as input.

2) Encryption Algorithm → Map the message text $m \in M$ to a ciphertext $c \in C$, either deterministically or non-deterministically.

3) Decryption Algorithm → Map the ciphertext $c \in C$ which is a valid ciphertext to messagetext m deterministically.

4) Message → A message given to us from a predefined message space M.

This is the 4-tuple $< KeyGen, Enc, Dec, M >$ given.

KeyGen → Non-d (random)
Enc → Non-d/d
Dec → strictly d
M → Given.

(b) 1) Shift Cipher → KeyGen → Choose a Key from 0-25 randomly

$Enc = (m \oplus K)$

$Dec = (c \oplus K)$

m → given plaintext.

2) Vigenere Cipher → Key → Choose a string of length ℓ randomly from the set of all strings of

length l.

Enc → Divide the messagetext m into blocks of length l.

For each block b,

$$c(b) = m(b) \oplus K.$$

and concatenate all the blocks obtained to get the ciphertext.

Dec → Divide the ciphertext c into blocks of length l,

For each block b,

$$m(b) = c(b) \oplus K$$

and concatenate all the blocks obtained to get the plaintext back.

2) a) They are explained in brief above.

Say: $M =$ 'attackatdawn'

$K =$ 'lemon'

$m(0) =$ attac	$m(1) =$ katda	$m(2) =$ wn
$K =$ lemon	$K =$ lemon	$K =$ le
lxfop	vefrn	hv

$C =$ lxfopvefrnhv

Yes. It is possible to obtain perfect secrecy using vigenere cipher.

We just have to choose a key as long as the message. Then it is equivalent to a vernam cipher, which is proven to be perfectly secret.

Breaking → Shift cipher → Brute force (check all 26 keys) and then do a frequency attack, and choose the closest ciphertext.

1) Vigenere Cipher → Brute force over all lengths and for each length ℓ, divide the message m by a round-robin fashion into ℓ boxes and then perform a frequency attack to check whether it is a meaningful plaintext

The case where Vigenere Cipher is perfectly secret is when ℓ == len(m) as then every box contains only 1 alphabet & shift cipher is perfectly secure for length 1, hence vigenere cipher becomes perfectly secure.

(b) $(26)^8 + (26)^9 + (26)^{10} + (26)^{11} + (26)^{12}$ is the size of the key space $\approx (26)^{12}$.

(c) If we use a mono-alphabetic substitution cipher instead of a shift-cipher attack; then we use the exact same attack as on the traditional vigenere cipher, the only difference being there is not a shift in the histogram, but a permutation, hence the histogram needs to be rearrange.

Hence, it is also completely secure in the exact same condition as the previous process, i.e. key has same length as the message.

3) 1) Shift Cipher → Compare the C & M and from the first character, calculate the shift s and we can decrypt any new message.

2) Substitution Cipher → Compare the known-plaintext & ciphertext, and note the mappings in between the plaintext & ciphertext. Use them to crack

any new given plaintext.

(c) XOR the m & c, to get the key and identify the smallest repeating substring which makes the string. Hence, we have the key, we can decrypt any new C is the vigenere ciphertext.

4) Perfect Secrecy:

a) $K \in K$.

Encryption scheme = $E_K = M$ (K is not used at all)

If such silly encryption schemes are possible then there is no perfect secrecy.

Encryption scheme $C = M + K$ (Addition simple) is not perfectly secure as 52 strictly implies that the plaintext & key were both 2.

(b) True.

Proof :- Let the K be the key

$$C = m \oplus K.$$

$K_1, K \in K$

$$Pr(K_1 = K) = \frac{1}{26} \quad \text{(Assuming random K)}$$

$$Pr(C_1 = C) = Pr(m \oplus K) = Pr(K) = \frac{1}{26}.$$

Hence, it is a perfectly secure scheme.

c) True.

Proof :- Since, the encryption scheme is perfectly secure,

$$Pr[M = m \mid C = C] = Pr[C = C]$$

[Shannon's Defn #2 of Perfect Secrecy]

$$\therefore Pr[M = m_0 \mid C = c] = Pr[M = m_1 \mid C = c] = Pr[C = C]$$

d) True.

For a Vernam Cipher,

Input $\rightarrow \{0,1\}^l$ Output $\rightarrow \{0,1\}^l$

Key $\rightarrow \{0,1\}^l$.

& we assume the Generator Algorithm uniformly chooses a key at random

$$\therefore \Pr[K = K] = \frac{1}{2^\ell}$$

$$\text{Enc} = m \oplus K$$

$$\text{Dec} = c \oplus K$$

$$\text{Now, } \Pr[C = c \mid M = m_0] = \Pr[M \oplus K = C \mid M = m_0]$$

$$= \Pr[m_0 \oplus K = c]$$

$$= \Pr[m_0 \oplus c = K] = \frac{1}{2^\ell}$$

Hence, it doesn't matter what m_0 is, the probability is $\frac{1}{2^\ell}$

(e) $M \Rightarrow$ Set of all alphabets is the biggest space for which perfect secrecy is defined.

For 2 letter words, if $C = ab$, we know that in the original message those 2 letters were different and if $C = aa$, then both letters were same.

5) a) $$P(a|b) = \frac{P(a \cap b)}{P(b)}$$

$$\therefore \Pr[M = \text{'aa'} \mid C = \text{'bb'}] = \frac{P(M = \text{'aa'} \cap C = \text{'bb'})}{P(C = \text{'bb'})}$$

Since $C = \text{'bb'}$, the only possible keys are $\{\text{'a'}, \text{'b'}, \text{'aa'}, \text{'ab'}, \text{'ba'}, \text{'bb'}\}$.

Out of this, 'a' is not possible due to the restricted message space. Also, 'aa', 'ab' are not possible.

$\therefore$ 'b' $\xrightarrow{K}$ with 'aa' — (1)
'ba' with 'ab' — (2)
'bb' with 'aa' — (3)

} give. $C = \text{'bb'}$

$Pr[M = \text{'aa'}]$ in this $= Pr[M = \text{'aa'}] \cdot Pr[\ell(K) = 1] \cdot Pr[K = \text{'b'}] + Pr[M = \text{'aa'}] \cdot Pr[\ell(K) = 2] \cdot Pr[K = \text{'bb'}]$.

$Pr[M = \text{'aa'} \mid K = \text{'b'}] + Pr[M = \text{'ab'} \mid K = \text{'ba'}] + Pr[M = \text{'bb'} \mid K = \text{'aa'}]$

$= 0.4 \times 0.5 \times \left(\frac{1}{26}\right) + 0.4 \times \frac{1}{2} \times \frac{1}{26} \times \frac{1}{26}$

$\left(0.4 \times 0.5 \times \left(\frac{1}{26}\right) + 0.4 \times \frac{1}{2} \times \frac{1}{26} \times \frac{1}{26} + 0.4 \times \frac{1}{2} \times \frac{1}{26} \times \frac{1}{26}\right)$.

$= \frac{0.2}{26} + \frac{0.2}{26 \times 26} = \frac{0.00798816}{0.00828402}$

$\frac{0.2}{26} + \frac{0.4}{26 \times 26} = 0.96428$

≈ 0.9643

Possible keys are 00100 & 11011 each with probability $\frac{1}{2^5}$

$$\therefore Pr[C = 00000] = Pr[M = 00100] \cdot Pr[K = 00100] + Pr[M = 11011] \cdot Pr[K = 11011]$$

$$= 0.1 \times \frac{1}{2^5} + 0.9 \times \frac{1}{2^5} = \frac{1}{2^5}$$

$$= 0.03125$$

6) Perfect Indistinguishability -

A cryptosystem is said to be Indistinguishable, if no adversary A can distinguish a chosen ciphertext C from 2 element message space determined by himself.

In layman terms, the adversary should learn no information from seeing a ciphertext.

The most advanced standard is IND-CCA2, which is indistinguishability given adaptive chosen ciphertext attacks.

If we achieve perfect secrecy, then obviously it is perfectly indistinguishable as perfect secrecy is an higher standard than IND-CCA2 secure.

No. He will not be able to generate any part of the original message from the Ciphertext given as indistinguishability implies that Ciphertext reveals nothing about the plaintext

under certain conditions (CPA , CCA1 , CCA2).

7) a) A function $\mu(n) : \mathbb{N} \to \mathbb{R}$ is negligible, if for every positive polynomial $poly(\cdot)$, there exists an integer $N_{poly} > 0$ such that for all $x > N_{poly}$

$$|\mu(x)| < \frac{1}{poly(x)}.$$

In practice, we use functions that are ~~exp~~ $O(\frac{1}{exp})$, i.e. not polynomially bound or $Pr < \frac{1}{2^{128}}$.

Hence,

(i) $\frac{1}{2^n}$ is $O\left(\frac{1}{2^n}\right)$ ∴ Negligible.

(ii) $(\log n)!$ → Not a very good definition, since factorials are defined only for integers.

If $\Omega((\log n)!)$ is 2^n then yes, it is negligible.

My guess would be it is ~~non~~ negligible as exp grows slower than factorial. If It was $\log(n!) = n \log n = O(n^2)$

Hence, non-negligible.

(iii) $\frac{1}{(\log \log n)!}$

Again. Same explaination. My guess, ~~non~~ non-negligible. as $\log\log$ grows ~~faster than factorial~~ extremely slow as compared to factorial, hence, wouldn't be negligible ever. But, just a guess.

(iv) $\frac{1}{1010}$ non negligible as $> 2^{-128}$.

(v) $n^{1/n}$

$$\lim_{n\to\infty} n^{1/n} = \lim_{n\to\infty} e^{\frac{\log n}{n}} = e^{\lim_{n\to\infty}\frac{\log n}{n}}$$

$$\therefore = 1+\varepsilon, \quad \varepsilon \to 0.$$

Non-negligible as $1 > 2^{-128}$.

(vi) → Non negligible $> 2^{-128}$

(vii) $\frac{n}{2^n}$ negligible as it is o(exp)

(viii) $\frac{1}{n}$ → Non-negligible as it o(poly).

(c) f, g are negligible functions.

(i) $H(n) = f(n) + g(n)$.

$f(n)$ is not o(poly) & neither $g(n)$ is o(poly).

∴ Their sum can never be o(poly) as exp. functions are closed under addition.

(ii) $H(n) = f(n) \times g(n)$

Negligible functions if multiplied become much more smaller. Negligible obviously!

(iii) $H(n) = f(n)/g(n)$ → Not negligible

Counter-example → $f(n) = \frac{n^2}{2^n}$

$$g(n) = \frac{1}{2^n}$$

Both negligible, but $f(n)/g(n) = n^2$ Not negligible!

8) a) Defⁿ → ~~As function~~ $A = \{A : \{0,1\}^n \to \{0,1\}^*\}$ is a class of functions, called as the statistical tests, also called Adversaries. They are PPTM-bounded ~~in~~ in number of checks allowed.

A function $G : \{0,1\}^\ell \to \{0,1\}^n$ with $\ell \leq n$ is a pseudorandom generator against A with a bias ϵ if for every A in A, the value of ϵ is negligible, i.e. the statistical distance between $A(G(U_\ell))$ and $A(U_\ell)$ is atmost ϵ, where U_k is uniform distribution $\{0,1\}^k$.

The expansion factor of an PRG is (n/ℓ).

b) i) True. By definition of a PRG.

(ii) Every element in G should be indistinguishable statistically. If the 0 were to lead to distinguishability in the 2n-bit string space, then NO. As for a given n-bit r, there is exactly 1 2n bit string.

(iii) ~~Yes. As concatenation of an n-bit & a 2n-bit uniform string, in principle forms a 3n bit string~~ concatenation is not a good operation for checking uniformity.

(iv) $G(r) \mid G(r+1)$ ~~Yes~~. Yes. As, $G(r)$ & $G(r+1)$ are computationally indistinguishable 2n-bit strings. Concatenation, in principle forms indistinguishable 4n-bit strings.

(v) ~~Reduce. Since coming back from~~ True. As if it were not one-way, then we could compute from where did the 2n-bit string originate from, rendering it not random.

9) Defⁿ → One-way function is a function that is easy to compute on every input, but hard to invert given the image of a random input. "Easy" and "Hard" are understood in sense of Computational Complexity theory.

(a) $h(x) \stackrel{def}{=} f(x) \oplus g(x)$

True. One-way function.

Random ⊕ Random = Random.

(b) $h(x) \stackrel{def}{=} f(f(x))$

True. One-way function.

$f(x) = j$

$f(j) =$ ans

Since, both steps are irreversible, the entire sequence is irreversible. Hence, one-way.

(c) No. If I know $f(x_1)$ & $f(x_2)$, then I can know $f(x_3)$ $x_3 = x_1 \| x_2$.

∴ Not one way!

(d) Incomplete question.

10) (a) It is secure only if every message is encrypted using a different key so that chosen plaintext attack is not possible. It is not CPA-secure though! We can distinguish messages from the ciphertext given the same key.

(b) No, it is not secure as the eavesdropper can tell if the messages are repeated or not given multiple encryption to the minimum.

DATE: / /
PAGE NO.:

11) A → Try outputting $2|s|+1$ number of different values
If any value repeats, it is $G(s)$, else r.

Yes. As if the PPTM-bound was not present on the Adversaries, then they could have cracked PRG's trivially, by brute force. It is not possible to replicate 2^{2n} space by 2^n dimensions.

12) (c). Trivial! $\frac{3}{4}$ % times $\approx (50+\varepsilon)$ %.

13). (a) It is not a PRG as the concatenation of 2 exact same set of bits. Hence, it is not a PRG as the adversary can output 1 if the half parts of both keys are same.

(b) True. It is a PRG assuming concatenation as when we concatenate, we can just have n bit random string with n bit random string, effective rendering a 2n bit random string.

(c) True. It is just selecting from the first mapping from a family of PRF's, hence it is a PRG.

(d) True - It is selecting 2 in a given family of PRF chosen randomly and concatenating them, hence it is a PRG.

- Ameya Prabhu, 201402004

15) 1) ~~False. It applies only to CCA-secure schemes~~ True. (IND-CPA secure) is the minimal standard for IND.

2) False. Perfectly secure schemes have must stronger requirements than CCA-secure schemes as in perfectly secure schemes, even if adversary is not PPTM-bounded, it is impossible to crack the scheme!!

3) True. As assumptions of CPA-security $\subseteq$ Assumptions for CCA-security.

19) a) One-way permutation: It is a One-way function (defined in (9)) which is injective & surjective, i.e. bijective in nature.

b) PRG & One way function → Defined previously.

c) PRF → They are a collection of efficiently computable functions which can emulate a random oracle in the following way: No efficient Algorithm can distinguish ~~between a~~ (with significant advantage) between a function chosen randomly from a PRF family & a random oracle (whose outputs are completely random)

d) Invertible PRG → If a PRG is bijective & it is efficiently computable backwards (maybe via a trapdoor) then it is an invertible PRG. If it is always easy going backwards, then it loses its property of being pseudo-random.

(a) Given a One-way permutation, if input is $x \in$ Input (OWP), we output its ~~output~~ image in the permuted domain; thus obtaining a PRG.

(b) X = PRG Y = ~~OWF~~ OWF

Let the inputs of the function be the inputs of the PRG and output be the value generated in the PRG given the same input. It is one-way because the output cannot be traced back to the input as it is indistinguishable from any other output, all appear random.

(c) Design a PRG taking 2 inputs $PRG(I_1, I_2)$
$I_1 \equiv$ Key for the PRF and the $I_2 \equiv$ Family of the PRF.
Hence, for each I_2, we will get a whole set

of outputs for inputs I_1, thus generating a function. For random values of I_2, we get a PRF.

(d). X = PRF Y = IPRF

Cannot be obtained always. It is a special case of a PRF. where there are special properties in the PRF by which if a trapdoor is found, it is easy to invert the function; or non-cryptographically secure PRF's can be generated such as xorshift.

19) Probabilistic encryption is required as otherwise a CPA is possible. Say we have a deterministic algorithm to ~~predict~~ encrypt. The adversary has to distinguish b/w M_0 & M_1 given C. He will encrypt M_1 (Adaptive CPA / CPA2) and see if they are same. If yes, then C is not M_0, if no, then C is M_0. Hence, he cracked it for M_0 message.

For a private key encryption to be indistinguishable on CPA, it has to be ~~deter~~ probabilistic in nature, as given M_1, it should not always generate C, hence, not giving the adversary any advantage in the (CPA2) test.

Yes, as for the message M_1 encrypted multiple times or for any amount of chosen plaintexts, the ciphertext must be not reveal anything about the plaintext.

(b) Formal definition of PRF :-

A family of functions

$f_s : \{0,1\}^{\lambda(|s|)} \longrightarrow \{0,1\}^{\lambda(|s|)}$, where $s \in \{0,1\}^*$

and $\lambda : \mathbb{N} \to \mathbb{N}$ is a PRF iff :-

1) $\forall s, x$, s.t $|x| = \lambda(|s|)$, computing $f_s(x)$ is efficient.
2) If F_n be a uniform distribution of functions over