# 6.857 — Problem Set 1 — Problem 3

## Detecting Pad Reuse

We are given $f(n) = \mathrm{poly}(n)$ independent ciphertexts $c_1, \ldots, c_{f(n)}$, each consisting of $n$ bits and produced by encrypting random plaintexts with independently and uniformly chosen one-time pads. Each ciphertext therefore behaves as a uniformly random $n$-bit string. Our goal is to show that, with high probability, the longest common substring between any two distinct ciphertexts (under any alignment) has length less than

$$\log_2 n + \log_2 \ln n = \log_2(n \ln n).$$

## Analysis

Consider first two ciphertexts $c_i$ and $c_j$ with $i \neq j$ aligned without offset. For each bit position, the two bits match with probability 1/2, since the ciphertext bits are independent and uniformly random. We may therefore model a sequence whose $k$th value is 1 if the bits match and 0 otherwise as a sequence of independent Bernoulli(1/2) trials. A run of consecutive matching bits corresponds to a run of heads in this sequence.

By the fact provided in the problem statement concerning the longest run $R_{1/2}(n)$ of heads in $n$ coin flips, we know that with high probability

$$R_{1/2}(n) < \log_2 n + \log_2 \ln n.$$

Thus, the longest matching substring between $c_i$ and $c_j$ under zero offset is bounded above by $\log_2(n \ln n)$ with high probability.

Next, consider the same pair of ciphertexts with a shift (offset) of $k$ bits, where $0 \leq k < n$. The overlap region contains $n - k \leq n$ bits, and by the same reasoning, the longest matching run satisfies

$$R_{1/2}(n-k) \leq R_{1/2}(n) < \log_2(n \ln n)$$

with high probability. Since there are $2n - 1 = O(n)$ possible offsets, the bound applies to every offset individually with high probability.

## Union Bound Over All Ciphertexts

There are $\binom{f(n)}{2}$ pairs of ciphertexts, which is polynomial in $n$ because $f(n)$ is polynomial in $n$. For each pair, there are $O(n)$ offsets. Therefore, the total number of comparisons we must consider is polynomial in $n$.

For any single pair and offset, the probability that a run longer than $\log_2(n \ln n)$ occurs is at most $1/\mathrm{poly}(n)$. Applying a union bound over polynomially many events still results in an overall failure probability of $o(1)$. Thus, with high probability, no pair of ciphertexts shares a longer repeated substring.

## Conclusion

With high probability, among $\text{poly}(n)$ independently generated $n$-bit ciphertexts, the longest matching substring between any two ciphertexts under any alignment has length strictly less than $\log_2(n \ln n)$. Therefore,

$$\boxed{\text{the longest repeated substring is } < \log_2(n \ln n) \text{ with high probability.}}$$

## Part (b)

Assuming English plaintext is encoded in US-ASCII (7 bits per character), an $n$-character passage corresponds to $7n$ bits. Using the upper bound from part (a), the longest repeated substring in random ciphertext (measured in characters) is approximately

$$\frac{\log_2(7n) + \log_2 \ln(7n)}{7} = \frac{\log_2(7n \ln(7n))}{7}.$$

Comparing this bound to the empirical Churchill data (longest aligned character runs in English text), we observe that the empirical plaintext curve eventually exceeds the theoretical ciphertext upper bound for larger $n$. This occurs because natural language exhibits significant redundancy and structural patterns (common digraphs such as “th”, “qu”, and highly frequent letters), whereas independent one-time-pad ciphertext behaves like random noise. Thus, for sufficiently large passages, English plaintext produces longer repeated runs than the random-ciphertext upper bound.

## Part (c)

We are given $f(n) = \text{poly}(n)$ ciphertexts $c_1, \ldots, c_{f(n)}$, each $n$ bits long, with exactly one pair encrypted using the same one-time pad. Let $N = n \cdot f(n)$ be the total ciphertext length. From part (a), any two ciphertexts with independently chosen pads share no common substring longer than

$$\log_2(n \ln n)$$

with high probability. From part (b), two plaintexts that share structure (e.g., English text) exhibit a much longer common substring, and thus if they share a pad, their ciphertexts will contain a correspondingly long common run. Therefore, $\log_2(n \ln n)$ serves as a threshold distinguishing pad reuse from independent pads.

### Algorithm

We find the longest substring shared between any two ciphertexts and compare it against this threshold.

1. Form a single string
$$S = c_1\$_1\, c_2\$_2 \cdots c_{f(n)}\$_{f(n)},$$
using distinct terminators not appearing in the ciphertexts.

2. Construct a generalized suffix tree for $S$ in $O(N \log N)$ time.

3. Traverse the tree to locate the internal node of maximum depth $d$, which corresponds to the longest repeated substring.

4. If $d \leq \log_2(n \ln n)$, output that no pad reuse exists. Otherwise, retrieve the two ciphertext identifiers appearing below that node and output those ciphertexts as the reused-pad pair.

### Correctness

The deepest internal node of the suffix tree represents the longest repeated substring in $S$. If it occurs in two distinct ciphertexts, its depth exceeds $\log_2(n \ln n)$ only if those ciphertexts share a pad; any pair of independent pads cannot exceed this bound by part (a). Repetitions within the same ciphertext correspond to offset alignment and similarly remain below the threshold.

### Running Time

Building the suffix tree dominates the cost, requiring $O(N \log N)$ time. All other steps are linear. Thus the reused-pad pair can be found in

$$\tilde{O}(N) = O(N \log N),$$

as required.