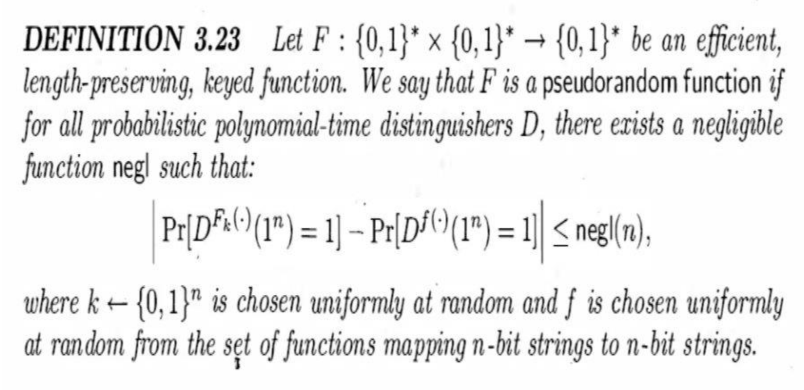
Soumodipta Bose| 2021201086

Build a Provably Secure PRF(Theory)

A pseudorandom function is a deterministic function of a key and an input that is indistinguishable from a truly random function of the input. More precisely, let s be a security parameter, let K be a key of length s bits, and let f(K,x) be a function on keys K and inputs x.



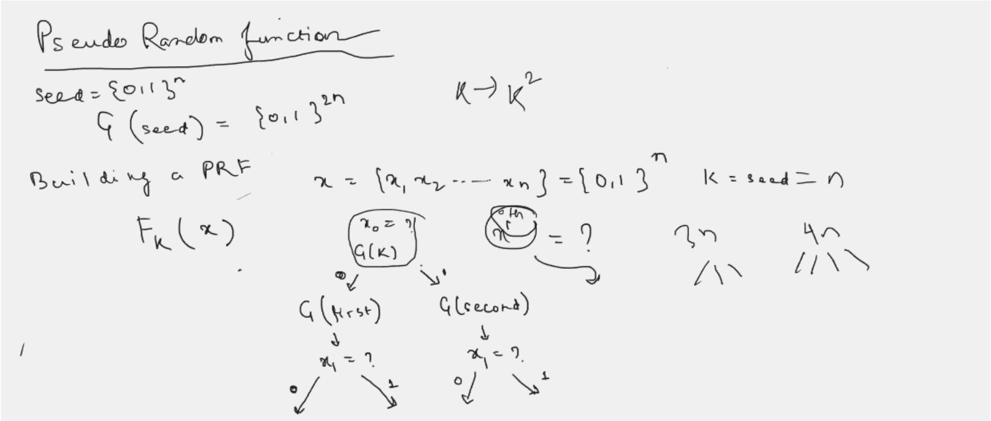
DEFINITION 3.23 *Let $F : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ be an efficient, length-preserving, keyed function. We say that F is a* pseudorandom function *if for all probabilistic polynomial-time distinguishers D, there exists a negligible function* negl *such that:*

$$\left|\Pr[D^{F_k(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot)}(1^n) = 1]\right| \leq \mathsf{negl}(n),$$

where $k \leftarrow \{0,1\}^n$ is chosen uniformly at random and f is chosen uniformly at random from the set of functions mapping n-bit strings to n-bit strings.

Assumption: The input key should have even number of bits

Sketches/Designs:



Construction:

We will be using a pseudo random generator such that it extends the bits by a factor of 2, i.e. for n bit input it would produce 2n bit output.

We start with two inputs to the PRF one is our key and the other is message/string input x. Both are in binary. The key has n bits and the string input has m bits.

Now we iterate through every bit of x.

1. For each iteration we fragment the key into two equal parts called first and last.

2. If the i^{th} bit of x is 0 then we apply pseudo random generator on the first part, else we apply pseudo random generator on the last part.
3. We would then use the newly generated random bits from the generator, and use it in the next iteration to fragment again based on the bits in x.

This way we would be able to obtain more efficient, truly random bits from a pseudo random generator.