2022201041

Eves dropping

Encryption with a pseudorandom generator is considered secure because it produces a stream of bits that is indistinguishable from true random bits. In other words, a pseudorandom generator generates a sequence of bits that appears to be random, but it is actually deterministic and generated by a mathematical algorithm.

The security of encryption with a pseudorandom generator relies on the fact that it is difficult to distinguish the generated stream of bits from a truly random stream of bits. If the generated stream of bits is truly indistinguishable from a random stream of bits, then it should be impossible to determine the original message that was encrypted.

However, it is important to note that the security of encryption with a pseudorandom generator also relies on the strength of the underlying cryptographic algorithm and the key used to generate the stream of bits. If the cryptographic algorithm is weak or the key is compromised, then the security of the encryption can be compromised as well.

Hardcore bit of discrete log [Blum-Micali '81]: Let p be an n-bit prime, $g \in \mathbb{Z}_p^*$. Define $B : \mathbb{Z}_p^* \to \{0, 1\}$ as

$$B(x) = msb(x) = \begin{cases} 0 & \text{if } x < p/2 \\ 1 & \text{if } x > p/2 \end{cases}$$

[1]

$$x_{i+1} = g^{x_i} \mod p.$$

The ith output of the algorithm is 1 if $x_i \leq \frac{p-1}{2}$. Otherwise the output is 0.

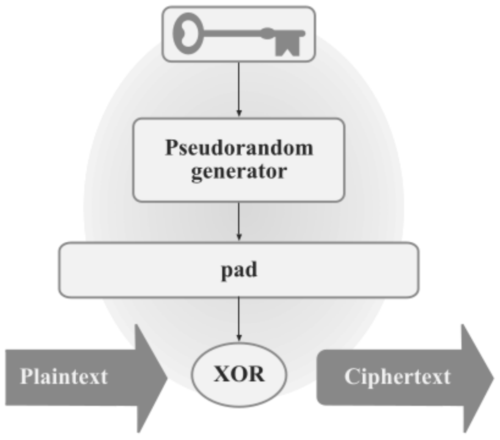

FIGURE 3.2: Encryption with a pseudorandom generator.

The encryption scheme. Fix some message length $\ell(n)$ and let G be a pseudorandom generator with expansion factor $\ell(n)$ (that is, $|G(s)| = \ell(|s|)$). Recall that an encryption scheme is defined by three algorithms: a key-generation algorithm Gen, an encryption algorithm Enc, and a decryption algorithm Dec. The key-generation algorithm is the trivial one: $\mathsf{Gen}(1^n)$ simply outputs a uniform key $k \in \{0,1\}^n$. Encryption works by applying G to the key (which serves as a seed) in order to obtain a pad that is then XORed with the plaintext. Decryption applies G to the key and XORs the resulting pad with the ciphertext to recover the message. The scheme is described formally in Construction 3.17. In Section 3.6.2, we describe how stream ciphers are used to implement a variant of this scheme in practice.

CONSTRUCTION 3.17

Let G be a pseudorandom generator with expansion factor $\ell(n)$. Define a fixed-length private-key encryption scheme for messages of length $\ell(n)$ as follows:

- Gen: on input 1^n, choose uniform $k \in \{0,1\}^n$ and output it as the key.
- Enc: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^{\ell(n)}$, output the ciphertext
$$c := G(k) \oplus m.$$
- Dec: on input a key $k \in \{0,1\}^n$ and a ciphertext $c \in \{0,1\}^{\ell(n)}$, output the message
$$m := G(k) \oplus c.$$