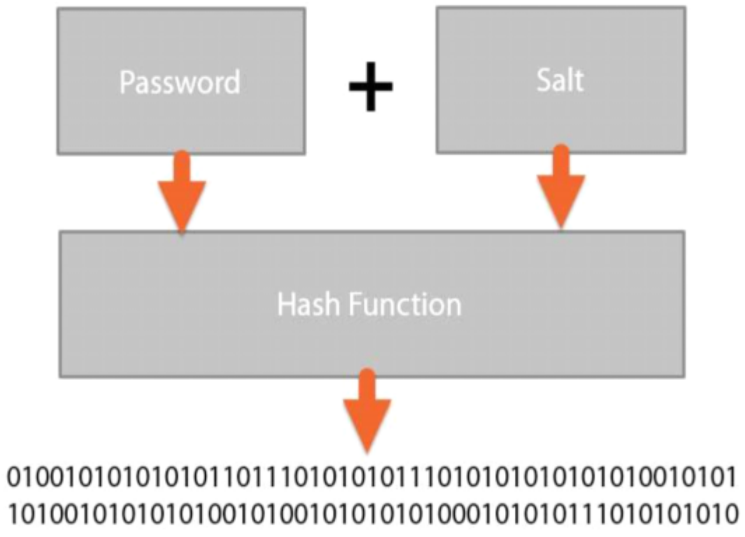

Figure 7: Add a salt to the password before hashing to increase entropy

The salt can be generated as a random number and then appended on the password before hashing. In the following code, we have a hash class containing everything we need to do this. This has the hash function itself, the salt generator, and a method to combine two byte arrays together.

```
public class Hash
{
    public static byte[] HashPasswordWithSalt(byte[] toBeHashed, byte[] salt)
    {
        using (var sha256 = SHA256.Create())
        {
            return sha256.ComputeHash(Combine(toBeHashed, salt));
        }
    }

    private static byte[] Combine(byte[] first, byte[] second)
    {
        var ret = new byte[first.Length + second.Length];

        Buffer.BlockCopy(first, 0, ret, 0, first.Length);
        Buffer.BlockCopy(second, 0, ret, first.Length, second.Length);

        return ret;
    }

    public static byte[] GenerateSalt()
    {
        const int SALT_LENGTH = 32;

        using (var randomNumberGenerator = new RNGCryptoServiceProvider())
        {
            var randomNumber = new byte[SALT_LENGTH];
```