Public Key Cryptography Exercises

Cryptology
FME (UPC) 2024
Jorge L. Villar

Last updated: Aug 8 20:10:16 2024

Public Key Encryption Schemes

1. Describe a variant of ElGamal encryption in which the encryption function is $\mathrm{Enc}(pk, m) = (g^r m, y^r)$ instead of the original one $\mathrm{Enc}(pk, m) = (g^r, y^r m)$. Show how can it be used to encrypt a message once for two different recipients as $\mathrm{Enc}_2(pk_A, pk_B, m)$.

2. The so-called **linear** encryption scheme is a variant of ElGamal encryption scheme with a "double" secret key (x_1, x_2) and public key $(\mathrm{param}, y_1 = g^{x_1}, y_2 = g^{x_2})$ such that $\mathrm{Enc}(pk, m) = (g^{r_1+r_2} m, y_1^{r_1}, y_2^{r_2})$.

   Describe the decryption function.

   Show that given only one of the secret values (say x_2) one can transform the ciphertext into a modified ElGamal ciphertext (described in the previous exercise) for the public key y_1 and the same message m.

   Observe that, in this way, linear encryption is a two-party ElGamal encryption: A ciphertext can only be decrypted with the collaboration of two parties A and B, holding each one one of the two secret values.

3. Using plain RSA to send exactly the same message to several users can be completely insecure. Show how a message $m < 2^\lambda$ can be efficiently recovered from three independent encryptions with public exponent $e = 3$ using public keys n_1, n_2, n_3.

   Hint: Use the Chinese Remainder Theorem with the product $n = n_1 n_2 n_3$.

4. Why cannot you encrypt very short messages with plain RSA and small public exponent?

Homomorphic Encryption

5. Show that if c is an encryption of $m \in Z_n$ using Paillier encryption scheme with $g = n + 1$, then for any $u \in Z_n$, $c^u \bmod n^2$ is an encryption of $mu \bmod n$.

   Use this to find a way to compute the encryption of $P(u) \bmod n$, given $u \in Z_n$ and the ciphertexts $c_0, ..., c_d$ encrypting the coefficients of the polynomial P of degree at most d.

6. Show how to compute the tally in a electronic voting scheme in which every voter generates its encrypted ballot as an ElGamal encryption of the generator g (for "yes"), or 1 (for "no").

   Hint: If the number of voters n is limited, the discrete logarithm of g^x for $0 \le x \le n$ can be efficiently computed with a precomputed table.

Digital Signatures

7. Discuss why the random number k generated in the DSA signing algorithm is critical for the security of the scheme. In particular, show that if k is leaked for a particular message / signature pair, then the attacker learns the secret signing key.

   Similarly, the same value of k cannot be used for more than one signature. Show that from two message / signature pairs generated with the same k, an attacker learns k and the secret signing key. Show that the same attack still works if the value of k comes from a counter (and then $k_2 = k_1 + 1$ for two consecutive DSA signatures).

   Hint: In the second part, consider the linear system of equations obtained from $kt - xr = H(m)$, for a message m, the corresponding DSA signature (r, t), and the secret key x.

(Generated by lineprocx v2.97)

8. Consider the following 3-moves identification protocol based on RSA: A prover P proves its knowledge of x such that $x^e = y \bmod n$, where the public RSA key (n, e) and y are common inputs given to both parties. First, P sends the commitment $a = r^e \bmod n$ to V, for a random r. Next, V sends a random challenge c to P. Then, P answers the challenge by sending $t = rx^c \bmod n$. Finally, V verifies whether $t^e = ay^c \bmod n$.

   Describe a Schnorr-like signature scheme based on the previous identification scheme.

Public Key Infrastructures

9. Describe the information needed by a user A to send a message to another user B with confidentiality guarantees, assuming that A and B uses two different certification authorities CA_A and CA_B, and these authorities depend on a common root certification authority $\mathrm{CA}_{\mathrm{root}}$. Specify all the public keys and the necessary fields in every certificate, and indicate all the verification operations performed by A before sending an encrypted message to B.