14. Anonymous Encryption and Robustness

The primary goal of an encryption scheme is data confidentiality: one cannot learn anything about the plaintext.

This security notion is modeled by the indistinguishability game: given an encryption scheme $S = (\text{Setup}, \mathcal{KG}, \mathcal{E}, \mathcal{D})$, the challenger runs the setup algorithm $\text{Setup}$ and the key generation algorithm $\mathcal{KG}$ to generate the encryption key $ek$ and the associated decryption key $dk$. It provides the adversary $A$ with the encryption key, and $A$ outputs two messages $(m_0, m_1)$. The challenger flips a bit $b \overset{R}{\leftarrow} \{0, 1\}$ and provides $A$ with an encryption $c^* = \mathcal{E}_{ek}(m_b)$ under $ek$. Eventually, $A$ has to guess $b$. To this aim, it outputs $b'$. The quality of the adversary $A$ is measured by its advantage

$$\text{Adv}_{S,A}^{\text{ind}}(A) = \Pr[1 \leftarrow A | b = 1] - \Pr[1 \leftarrow A | b = 0] = \Pr[\text{Exp}_{S,A}^{\text{ind}}(1) = 1] - \Pr[\text{Exp}_{S,A}^{\text{ind}}(0) = 1].$$

The security of the encryption scheme $S$ is measured by the advantage of the best adversary within time $t$:

$$\text{Adv}_{S}^{\text{ind}}(t) = \max_{A \leq t} \{\text{Adv}_{S,A}^{\text{ind}}(A)\}.$$

Another security goal is anonymity: one cannot learn anything about the recipient. More precisely, between two possible public keys, no adversary should be able to guess which one has been used to generate the ciphertext.

Q-1. Let us consider that every users are associated with pairs $(ek, dk)$.

How would you formalize anonymity, with a security game as above?

Note that this notion has been termed: key privacy

Let us consider the following RSA encryption variant:

- $\text{Setup}(k)$: $\text{params} = (k, \ell, \mathcal{H})$, where $\mathcal{H}$ is a random hash function onto $\{0, 1\}^\ell$;
- $\mathcal{KG}(\text{params})$: Choose two random $k$-bit prime integers $p$ and $q$, set $n = pq$ and $e = 2^{16} + 1$, and define $ek = (\mathcal{H}, n, e)$, $dk = (\mathcal{H}, n, d = e^{-1} \mod \varphi(n))$;
- $\mathcal{E}_{ek}(m)$, for $m \in \{0, 1\}^\ell$: choose $r \overset{R}{\leftarrow} \mathbb{Z}_n^*$, and set $c = (c_1 = r^e \mod n, c_2 = \mathcal{H}(r) \oplus m)$;
- $\mathcal{D}_{dk}(c)$, for $c = (c_1, c_2) \in \mathbb{Z}_n^* \times \{0, 1\}^\ell$: $m = c_2 \oplus \mathcal{H}(c_1^d \mod n)$.

Q-2. Show that this RSA encryption scheme provides data confidentiality, in the random oracle model.

Detail a proof by successive alterations of the security game, and precise the security level under the intractability of the RSA problem.

Q-3. Does this RSA encryption scheme provide key privacy?

If yes, prove it; if not, exhibit a distinguisher, and evaluate its quality, with any reasonable assumption on the distribution of the integers you would need.

Q-4. How one can improve the key privacy property for this scheme?
Let us consider the ElGamal encryption:

- **Setup**(*k*): params = (𝑝, 𝑔, ℋ), where 𝑔 is a generator of the cyclic group ℋ of prime order 𝑝;
- **KG**(params): Choose a random 𝑥 ℛ ℤ𝑝 and define ek = (ℋ, 𝑝, 𝑔, 𝑦 = 𝑔𝑥), dk = (ℋ, 𝑝, 𝑔, 𝑥);
- **E**ek(*m*), for *m* ∈ ℋ: choose 𝑟 ℛ ℤ𝑝, and set 𝑐 = (𝑐1 = 𝑔𝑟, 𝑐2 = 𝑦𝑟 × *m*);
- **D**dk(𝑐), for 𝑐 = (𝑐1, 𝑐2) ∈ ℋ2: *m* = 𝑐2/𝑐1.

**Q-5.** Show that the ElGamal encryption scheme provides both data confidentiality and key privacy.
Detail the proofs by successive alterations of the security games, and precise the security levels under the intractability of the DDH problem.

The robustness property is an additional property that allows any recipient to easily check whether the ciphertext is targeted to him. More precisely, an encryption scheme is (weakly) robust if no adversary can find a message *M* and two users (or equivalently two public keys) ek₀ and ek₁ such that an honestly generated ciphertext 𝑐 of *M* under ek₀ can be decrypted successfully under the decryption key associated to ek₁.

**Q-6.** Is the ElGamal encryption scheme (weakly) robust?

A natural solution is to add redundancy:

- **Setup**(*k*): params = (𝑝, 𝑔, ℋ), where ℋ is a random hash function onto {0, 1}𝓁;
- **KG**(params): Choose a random 𝑥 ℛ ℤ𝑝 and define ek = (ℋ, 𝑝, 𝑔, ℋ, 𝑦 = 𝑔𝑥), dk = (ℋ, 𝑝, 𝑔, ℋ, 𝑥);
- **E**ek(*m*), for *m* ∈ ℋ: choose 𝑟 ℛ ℤ𝑝, and set 𝑐 = (𝑐1 = 𝑔𝑟, 𝑐2 = 𝑦𝑟 × *m*, 𝑐3 = ℋ(𝑚));
- **D**dk(𝑐), for 𝑐 = (𝑐1, 𝑐2, 𝑐3) ∈ ℋ3 × {0, 1}𝓁: *m* = 𝑐2/𝑐1, check whether 𝑐3 = ℋ(𝑚) before outputting 𝑚 or “invalid”.

**Q-7.** Does this ElGamal variant provide data confidentiality, key privacy and (weak) robustness?

Another variant with redundancy:

- **Setup**(*k*): params = (𝑝, 𝑔, ℋ, 𝑦), where 𝑔 and ℎ are independent generators of ℋ;
- **KG**(params): Choose a random 𝑥 ℛ ℤ𝑝 and define ek = (ℋ, 𝑝, 𝑔, ℎ, 𝑦 = 𝑔𝑥, ℎ = ℎ𝑥), dk = (ℋ, 𝑝, 𝑔, ℎ, 𝑥);
- **E**ek(*m*), for *m* ∈ ℋ: choose 𝑟 ℛ ℤ𝑝, and set 𝑐 = (𝑐1 = 𝑔𝑟, 𝑐2 = ℎ𝑟 × *m*, 𝑐3 = ℎ𝑟);
- **D**dk(𝑐), for 𝑐 = (𝑐1, 𝑐2, 𝑐3) ∈ ℋ3: *m* = 𝑐2/𝑐1, check whether 𝑐3 = ℎ𝑟 before outputting 𝑚 or “invalid”.

**Q-8.** Does this ElGamal variant provide data confidentiality, key privacy and (weak) robustness?

**Q-9.** Discuss about generic techniques to make an anonymous encryption scheme robust.