Sketch of Lecture 30

Fri, 4/5/2019

Further comments on RSA and ElGamal

Theorem 171. Determining the secret private key d in RSA is as difficult as factoring N.

Proof. Let us show how to factor $N = pq$ if we know e and d.

- Write $ed - 1 = 2^t m$, where t is chosen as large as possible such that 2^t divides $ed - 1$.
  Since $ed - 1 \equiv 0 \pmod{(p-1)(q-1)}$ and 2^2 divides $(p-1)(q-1)$, we have $t \geqslant 2$.
- Pick a random invertible residue x. Observe that $x^{ed-1} \equiv 1 \pmod{N}$. In other words, $(x^m)^{2^t} \equiv 1$.
  Hence, the multiplicative order of x^m must divide 2^t.
- Suppose that x^m has different order modulo p than modulo q.

  Note. This works for at least half of the (invertible) residues x. If we are unlucky, we just select another x.

  Since both orders must divide 2^t, we may suppose x^m has order 2^s modulo p, and larger order modulo q.
  Then, $x^{2^s m} \equiv 1 \pmod{p}$ but $x^{2^s m} \not\equiv 1 \pmod{q}$.
  Consequently, $\gcd(x^{2^s m} - 1, N) = p$ so that we have found the factor p of N.

  Note. Of course, we don't know s (because we don't know p and q), but we can just go through all $s = 1, 2, \ldots, t - 1$. One of these has to reveal the factor p. $\square$

However. It is not known whether knowing d is actually necessary for Eve to decrypt a given ciphertext c. This remains an important open problem.

Example 172. (homework) Bob's public RSA key is $N = 323$, $e = 101$. Knowing $d = 77$, factor N using the approach of the previous theorem.

Solution. Here, $de - 1 = 7776 = 2^5 \cdot 243$ so that $t = 5$ and $m = 243$.

- Let's pick $a = 2$. $a^m = 2^{243} \equiv 246 \pmod{323}$ must have order dividing 2^5.
  $\gcd(246^2 - 1, 323) = 19$ (so we don't even need to check $\gcd(246^{2^s} - 1, 323)$ for $s = 2, 3, 4$)
  Hence, we have factored $N = 17 \cdot 19$.

Comment. Among the $\phi(323) = 16 \cdot 18 = 288$ invertible residues a, only 36 would not lead to a factorization. The remaining 252 residues all reveal the factor 19.

Another project idea. Run some numerical experiments to get a feeling for the number of residues that result in a factorization.

Armin Straub
straub@southalabama.edu

Definition 173. Bob's public key cryptosystem is **semantically secure** if Eve cannot do better than guessing in the following challenge:

- Bob determines a random public and private key. The public key is given to Eve.
- Eve selects two plaintexts m_1 and m_2.
- Alice flips a fair coin and, accordingly, using the public key encrypts m_1 or m_2 as c.
- Eve now needs to decide whether c is the encryption of m_1 or m_2.

For this definition to make precise mathematical sense, we need to assume that Eve's computing power is somehow limited (typically, she is limited to polynomial-time algorithms).

Comment. Also, many variations exist of what semantic security exactly is. All of these try to capture the idea that an attacker does not learn anything about m from knowing c. The one above is often referred to as IND-CPA (Indistinguishability under Chosen Plaintext Attack).

Important comment. Realize that semantic security is a very strong property to ask for! In particular, this is much stronger than what we usually think about in terms of security: you might call a cipher secure if it is "impossible" for an attacker to get m from c. Semantic security is requiring that an attacker gets so little information from c that she cannot even tell whether it came from (her own choices) m_1 or m_2.

Example 174. Is vanilla RSA semantically secure?

Solution. No. Eve can just encrypt both m_1 and m_2 herself, and compare with c. She then knows for sure which of the two was encrypted.

Comment. As mentioned before, in practice, RSA is never used in its vanilla (or "textbook") version (unless random plaintexts are encrypted). Instead, it is randomized (like ElGamal is by design) by padding the plaintext with random stuff.

Check out OAEP: https://en.wikipedia.org/wiki/Optimal_asymmetric_encryption_padding

The resulting RSA-OAEP has been proven semantically secure (under the "RSA assumption" that finding m from c is hard).

Example 175. Is ElGamal semantically secure?

Solution. Essentially, yes.

Recall that the public key is $(p, g, h) = (p, g, g^x)$.

The ciphertext is $(c_1, c_2) = (g^y, h^y m) = (g^y, g^{xy} m)$. Eve needs to decide whether the m in there is m_1 or m_2.

Equivalently, she needs to decide whether $r = c_2/m_1$ (or $r = c_2/m_2$) equals g^{xy} or not.

This is essentially the DDH problem.

Strictly speaking. Because of the issue with quadratic residues mentioned when we introduced the DDH problem, ElGamal is not semantically secure in the sense we defined things. However, if we wanted (this is more of a theoretical point), this issue could be fixed by not computing with all invertible residues modulo p, but only with quadratic residues. We could further select p to be a **safe prime**, meaning that $(p-1)/2$ is prime again, in which case all quadratic residues (except 1) have order $(p-1)/2$ (so that no similar games can be played using orders of elements).

Practical implications. Indeed, Diffie–Hellman and ElGamal in practice often use safe primes p. In that case, as we observed in Example 169, there is no elements of small order (besides 1 and -1). Since generating such primes can be a bit expensive, it is common to use preselected ones. For instance, RFC 3526 lists six such primes (together with a generator g) with $1536, 2048, \ldots, 8192$ bits.

https://www.ietf.org/rfc/rfc3526.txt

Important. It is perfectly fine that p and g are not random in Diffie–Hellman or ElGamal. However, it is absolutely crucial that x (and y) are random (generated using a cryptographically secure PRG).