Performing the simplest version of Haståd's Broadcast Attack, and reversing OAEP

August 28, 2023

1 Introduction

This challenge is doubly educational: we have to apply the simplest version of Haståd's broadcast attack AND reverse padding. Firstly though, some suitable cipher-key-pairs had to be harvested from the server (i.e., I queried the server until I got three key-cipher-pairs where the public encryption exponent was equal to 3). Apparently I was lazy and did this by hand. In the follow-up challenge (noble collector), I automated this querying procedure.

2 Håstad's Broadcast - special case with identical messages

In this version of the Håstad Broadcast attack, the same message is encrypted multiple times using the same public exponent e. In the standard case, $e = 3$ and the same message has been sent thrice, e.g., the same invite has been sent in identical form (without personal addressing - how impolite!) to three different people. While the public exponent used for RSA encryption was the same for all three of them, the respective moduli n_1, n_2, n_3 were different.

The attack is possible thanks to the following *Lemma*: "If a message is encrypted with the same exponent e but e different moduli n_i, we can recover the message." - Here's the proof given in the lecture slides:

Firstly, among the three moduli there might be a pair that is not co-prime. Then we are done immediately: we have found a prime factor. Thus, in the following we will without loss of generality assume that n_1, n_2, n_3 are pairwise co-prime and we have a system of congruences of the form $c_i \equiv m_i^e \mod n_i$, $i \in \{1, ..., e\}$.

Now, put $x = m^e$ and solve via CRT. This gives us a unique solution $m^e \mod \prod n_i$. Now we observe: m must be smaller than each of the n_i. From this follows that $m^e < \prod n_i \implies m^e \mod \prod n_i = m^e$. Hence, we can just compute the third root, like we did in previous challenges and retrieve m.

3 OAEP - Optimal Asymmetric Encryption Padding

To make the challenge, well, a bit more challenging we now also have to reverse the padding that was applied to m before encryption. This padding is OAEP, the current (gold?) standard of padding when it comes to asymmetric encryption. Reversing it requires either building from the code provided with the challenge or following the instructions in the Wikipedia article. Also this was the first of the challenges I completed as part of this course and thus I was still commenting my code really neatly, sigh!