Cryptography, homework sheet 3

Due for 2MMC10: 24 September 2020, 10:45 by email to `crypto.course@tue.nl`
and for Mastermath: 15 October 2020, 10:45 by email to `crypto.course@tue.nl`

Team up in groups of two or three to hand in your homework. We do not have capacity to correct all homeworks individually. Do not email Tanja your homework or put homework in mailboxes.
You may use computer algebra systems such as mathematica, gp, or sage or program in C, Java, or Python. Please submit your code (if any) as part of your homework. If you do, make sure that your programs compile and run correctly; the TAs will not debug your programs. The program should also be humanly readable. Code should be sent to `crypto.course@tue.nl` along with the homework.

1. Show how to retrieve the message M in RSA-OAEP from $m_1||m_0$. (Notation as in class on 17 Sep 2019, modulo the G and H confusion that got fixed on the pictures but not the video). This is just considering the encoding and decoding of the message and skips the RSA part. The functions G and H are cryptographic hash functions, so you cannot invert them.

2. You learn that I sent ciphertext
   $c = 221742016667880335235086977604419933217657946219108301$ to a user with RSA public key $(e, n) = (3, 529774210762246675161318616746995617835565246251635147)$ and that this was the result of a form which sends a stereotyped message `myfavoritenumberis____` in base 35, where the empty spaces indicate 6 unknown characters. Use LLL to recover those 6 characters.
   Note that you are not guaranteed to succeed with the first output of LLL. Also note that you can (and should) check your solution.

3. **Combination of hash functions.** Are the following claims true or false? Either present a proof by giving a reduction as in the lecture or a counter example.

   (a) Let $h : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be an efficient keyed permutation. Let $H_k = h_k \circ h_k$ be the permutation resulting from applying h twice, i.e., $H_k(m) = h_k(h_k(m))$.
   Claim: If h is preimage resistant (PRE), H is preimage resistant.

   (b) Let $h^1 : \{0,1\}^n \times \{0,1\}^{\ell(n)} \to \{0,1\}^n$ and $h^2 : \{0,1\}^n \times \{0,1\}^{\ell(n)} \to \{0,1\}^n$ be hash functions.
   Claim: The combined hash function $H : \{0,1\}^{2n} \times \{0,1\}^{\ell(n)} \to \{0,1\}^{2n}$, $(\langle k_1, k_2 \rangle, m) \mapsto h^1_{k_1}(m)||h^2_{k_2}(m)$ is collision resistant if at least one of h^1 and h^2 is collision resistant.

   (c) Let $h^1 : \{0,1\}^{n_1} \times \{0,1\}^{\ell(n_1)} \to \{0,1\}^{n_1}$ and $h^2 : \{0,1\}^{n_2} \times \{0,1\}^{n_1} \to \{0,1\}^{n_2}$ be hash functions.
   Claim: The combined hash function $H : \{0,1\}^{n_1+n_2} \times \{0,1\}^{\ell(n_1)} \mapsto \{0,1\}^{n_2}; (\langle k_1, k_2 \rangle, m) \mapsto h^2_{k_2}(h^1_{k_1}(m))$ is collision resistant if at least one of h^1 and h^2 is collision resistant.

4. **Multi-target attacks.** Sometimes an attacker gets to attack multiple targets at once and is satisfied breaking any *one* of them. For hash functions multi-target preimage attacks are interesting. We speak of a t-target preimage attack if the attacker is given the outputs $h_k(m_1), h_k(m_2), \ldots, h_k(m_t)$ (and k) but not the inputs $m_1, m_2, \ldots, m_t$ of a hash function $h : \{0,1\}^n \times \{0,1\}^{\ell(n)} \to \{0,1\}^n$ and has the goal of finding a pair (i, x) such that $h_k(x) = h_k(m_i)$.

(a) Show that a t-target preimage attack $\mathcal{A}$ succeeding with probability p can be turned into a 1-target preimage attack, i.e., a regular preimage attack, taking the same time as A and succeeding with probability p/t.

(b) The algorithm you just developed is actually also a reduction. What did you prove with that algorithm (In terms of property A implies property B)?

(c) Find an attack that takes time $2^n/t$ to succeed in finding one (i, x) with high probability.