Cryptography I, homework sheet 4

Due: 2 October 2014, 10:45

Please submit your homework electronically. If you use scans, please bundle them into one pdf file. Team up in groups of two or three to hand in your homework. We do not have capacity to correct all homeworks individually. To submit your homework, email it to `crypto14@tue.nl`.

1. The PGP cryptosystem uses RSA as the public-key system. To enable humans to check that they downloaded the correct key, short fingerprints of the keys are provided which are easier to compare to one provided e.g. on a business card. In version 3 of PGP a key could be referenced by its key ID, its shortened key ID, or its fingerprint. The 128-bit fingerprint was obtained as the MD5-hash of the user's public key (apart from some minor details the (n, e) used in RSA); the 64-bit key ID was obtained as the bottom 64 bits of the public modulus, and the shortened key ID was the bottom 32 bits of the key ID. The most commonly used way of checking for correctness of the public key was to use the `pgp -kv` command, which outputs the shortened key ID.

   In the following you should ignore the problem that the public keys are RSA keys and just assume that any string would be accepted.

   Your task is to fool a user into accepting your key as the key of Deaddrop, an email account for leaking documents to. Deaddrop's key is widely available on the internet in the following representations:

   (a) the full 128-bit fingerprint;

   (b) the 64-bit key ID; or

   (c) the 32-bit shortened key ID.

   For each of these three situations give an estimate of how many operations it takes to generate a matching fake key with high probability.

   How do your three attacks compare to the rogue-CA attack in terms of complexity? For this question, ignore that the rogue-CA exploited weaknesses of MD5 beyond its short output length of 128 bits.

2. 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 k-target preimage attack if the attacker is given the outputs $h(m_1), h(m_2), \ldots, h(m_k)$ but not the inputs $m_1, m_2, \ldots, m_k$ of a hash function $h : \{0,1\}^* \to \{0,1\}^n$ and has the goal of finding some (i, x) so that $h(x) = h(m_i)$.

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

   (b) Show that a k-target preimage attack 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/k.