CHALMERS UNIVERSITY OF TECHNOLOGY
Dept of Computer Sciernce and Engineering
Björn von Sydow DIT250/TDA352

Solutions to exam in Cryptography, Friday January 14, 2022.

1. (a) A hash function H is collision resistant if it is infeasible to find x_1 and x_2 such that $H(x_1) = H(x_2)$.

   (b) We have a document D that we want signed. The signing operation is not applied to D itself but to the hash value $H(D)$. This makes the signing operation faster and can also prohibit some attacks based on the mathematical structure of signing, such as the homomorphic property of RSA signatures. Finally, it makes it unnecessary to define the signing operation for long inputs.

   (c) All hash functions are vulnerable to the birthday attack on collision resistance. A hash function with $2n$ bit values gives only n bits of security against this attack, so this choice matches the security level of the system.

2. (a) The first step is to choose (randomly) two primes p and q of approximately equal size such that $N = p \cdot q$ has the desired bitlength. Then a public key e is chosen, typically a small positive integer, with $\gcd(e, \Phi(N)) = 1$, where $\Phi(N) = (p-1)(q-1)$. Finally, the private key $d = e^{-1} \in \mathbb{Z}_{\Phi(N)}$ is computed. We do an example with $p = 5$, $q = 11$. Then $N = 55$ and $\Phi(N) = 40$. We choose $e = 3$ and compute d using the extended Euclidean algorithm with inputs 40 and 3, at the same time verifying that $\gcd(e, \Phi(N)) = 1$:

r_i	q_i	s_i	t_i
40		1	0
3		0	1
1	13	1	-13
0			

   We get $\text{gcdE}(40, 3) = (1, 1, -13) = (1, 1, 27)$, i.e. $d = 27$.

   (b) $65537 = 2^{16} + 1$, so this requires fewer multiplications than other numbers of similar size.

   (c) The public key is chosen small, but the private key will typically be n bits long. Thus decryption will require between n and $2n$ modular multiplications, which is in the order of 100 times that of encryption for typical key sizes.

   (d) Key recommendations try to match keys for different primitives so that the best attacks against each primitive require roughly the same amount of work. The work for a brute force search on AES with 128 bit keys is roughly the same as for factoring a 3072 bit integer.

3. (a) The verify algorithm takes a key, a message and a tag and returns accept/reject. (Alternatively, if non-deterministic MAC's are not considered, verification can be done by redoing signing and comparing the result with the given tag.)

   (b) A MAC is secure if an attacker, who is allowed to choose a number of messages and get them signed, still cannot produce a valid, fresh (message,tag) pair, except with negligible probability.

   (c) The proposed MAC is insensitive to the order of blocks, so an attacker could choose $M_1 \neq M_2$ and ask for a MAC on M_1M_2. If he gets the tag T, he can then produce (M_2M_1, T), which will be valid.

4. OTP is $c = k \oplus m$, where k is a random bitstring. A brute force attack on a cipher is an attack that tries all possible keys in order to achieve its goal (e.g. to retrieve the plaintext).

   OTP has perfect secrecy. Thus given c, every plaintext of the same length has the same probability (over choice of key), so all six-digit plaintexts are possible and are equally likely, given the encrypted string. There is thus no way for Eve to prefer one plaintext before another.

5. Authentication means that both parties should be convinced of the identity of the party at the other end. Perfect forward secrecy means that, even if the subsequent session is recorded and later the long-term keys of the parties are compromised, the content of the session is not leaked.

   To achieve this, both long-term (for authentication) and short-term (for forward secrecy) keys should be involved in the key derivation. One possibility is that Alice computes $k = H(B^a||Y^x)$ and Bob computes $k = H(A^b||X^y)$. (Other combinations are possible). Both parties should also verify the CA signature on the certificate received.

6. (a) $m' = D^{\text{CBC}}(k, c)$; parse m' as $t||m$. If $t \neq h(m)$ return **reject**, else return m.

   (b) After step 2, the Challenger chooses b and k at random and computes $t = h(h(m_0)||m_b)$. Then $t||h(m_0)||m_b$ will be encrypted in CBC mode, giving the four block ciphertext $c = C_0C_1C_2C_3$ (here we for simplicity assume that m_0 and m_1 are one block messages). In step 4, the attacker removes C_0; the remaining ciphertext is a valid CBC encryption of $h(m_0)||m_b$. So the decryption result r will be m_0 if $b = 0$ and **reject** if $b = 1$. The winning algorithm for the attacker is therefore:
   if $r =$ **reject** return 1, else return 0 (other formulations possible).

   (c) A well-established strategy to get CCA security is to encrypt using a CPA secure cipher, and then MAC the ciphertext using a secure MAC (encrypt-then-MAC):

$$\mathbb{E}((k_e, k_a), m) = (c, \text{MAC}(k_a, c)) \text{ where } c = E^{\text{CBC}}(k_e, m).$$

7. (a) The receiver computes $k = H(Y^x)$ and then $m = D(k, c)$. This will give correct decryption since $Y^x = X^y$, so sender and receiver will get the same k.

   (b) Let (i_j, Z_j) for $j = 1, \ldots t$ be the received replies. S first computes the t Lagrange polynomials $\{\delta_j\}_{j=1}^t$ based on $i_1, i_2, \ldots i_t$. These have the property that $f(x) = x_1\delta_1(x) + x_2\delta_2(x) + \ldots + x_t\delta_t(x)$, where f is the polynomial used to compute the x_i, so $x = f(0) = x_1\delta_1(0) + x_2\delta_2(0) + \ldots + x_t\delta_t(0)$.
   S can then compute

$$H(Z_1^{\delta_1(0)} \cdot Z_2^{\delta_2(0)} \cdot \ldots \cdot Z_t^{\delta_t(0)}) = H(Y^{x_1\delta_1(0)+x_2\delta_2(0)+\ldots+x_t\delta_t(0)}) = H(Y^x) = k,$$

   and proceed as in (a).
   Remark: The explicit formula for δ_j is

$$\delta_j(x) = \prod_{\substack{\nu = 1, \ldots t, \\ \nu \neq j}} \frac{x - i_\nu}{i_j - i_\nu}.$$

   but this is of minor importance. Note also that the secret key x is never reconstructed during decryption.