Solution

Cryptography – Homework 7

Discussed on Tuesday, 27th January, 2015.

For questions regarding the exercises, please send an email to schlund@in.tum.de or just drop by at room 03.11.055

Exercise 7.1 RSA

For this exercise we will use the notation (for elements of the RSA-PKES and PSS$[k,l]$) introduced in the lecture. We will use msbf-representation throughout the exercise.

(a) We choose two primes $p = 17$ and $q = 19$, select $e = 5$ and use them for instantiating the RSA encryption scheme. Then $N = pq = 323 = (101000011)_2$, i.e., N's bit-length is 9. We apply the OAEP padding scheme to the messages we want to encrypt using the following (very simple and unrealistic) instantiation:

- (Unpadded) messages m are 3-bit strings,
- $k_1 = 2$ and $k_0 = 4$,
- $G(x_1x_2x_3x_4) = x_1x_2x_3x_4x_1$, for $x_i \in \{0,1\}$ for $1 \leq i \leq 4$,
- $H(x_1x_2x_3x_4x_5) = (x_1 \oplus x_2)(x_2 \oplus x_3)(x_3 \oplus x_4)(x_4 \oplus x_5)$, for $x_i \in \{0,1\}$ for $1 \leq i \leq 5$.

Let us assume we receive the ciphertext 116. Compute d and the original (3-bit) message. Apply the Chinese Remainder Theorem in your computation.

(b) Show that decoding/encoding works regardless of the choice for G and H in the OAEP padding scheme.

(c) We assumed for the RSA scheme that the plaintext message m is an element of $\mathbb{Z}_N^*$. Show that encryption and decryption is also possible if $m \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$.

(d) We want to sign a message using the PSS$[k,l]$ scheme. We reuse the values from the previous exercise for N, d and e, and add the following parameters:

- $n = 10$, $k = l = 3$,
- $h(x_1 \dots x_s) := x_1x_2x_3$,
- $g(x_1x_2x_3) := x_1x_2x_3x_1x_2x_3$.

Compute $\mathsf{Sgn}_{(323,d)}(0110)$. Assume hereby that r was chosen as the bitstring 101.

Solution:

(a) By using the extended Euclidean algorithm for the arguments $M = \text{lcm}(16,18) = 144$ and $e = 5$, we obtain $d = 29$, the inverse of e in $\mathbb{Z}_M^*$ (if we had chosen $M = \varphi(N) = 288$) we would have gotten $d = 173$). Now the decryption $\hat{m}$ of the ciphertext 116 can be carried out by computing $116^{29} \bmod N$. We use the chinese remainder theorem and compute

- $116^{29} \bmod p = 116^{29} \bmod 17 = 14^{29} \bmod 17 = 14^{13} \bmod 17 = 5$ and
- $116^{29} \bmod q = 116^{29} \bmod 19 = 2^{29} \bmod 19 = 2^{11} \bmod 19 = 15$.

We apply the inverse morphism $h^{-1}(x,y) = 9 \cdot 17 \cdot y - 8 \cdot 19 \cdot x$ to the pair $(5,15)$ to get back $\hat{m}$

$$h^{-1}(5,15) = \hat{m} = 243.$$

(recall that h^{-1} can be obtained by the extended euclidean algorithm since $\gcd(17,19) = 1$.) Since the OAEP scheme was used, we compute (see next exercise), we write $\hat{m} = 243 = (011110011)_2$ and obtain the strings $X = 01111$ (the first 5 bits) and $Y = 0011$ (the last 4 bits).

- Compute $H(01111) = 1000$.
- Compute $H(01111) \oplus 0011 = 1011$.

- Compute $G(1011) \oplus 01111 = 10111 \oplus 01111 = 11000 (= m0^2)$.

We obain the message $(110)_2 = 6$ (two zeros are padded).

(b) The structure of the OAEP scheme is similar to a two-round Feistel network. Given the two output strings X, Y, we show how to obtain the message m (we use the notation introduced in the lecture):

- Compute $H(X) = H(m0^{k_1} \oplus G(r))$
- Compute $H(X) \oplus Y = r$.
- Compute $G(r) \oplus X = m0^{k_1}$.

(c) We have to show that for every $m \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$, $m^{ed} = m$ holds.
For $m \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$, $p \,|\, m$ or $q \,|\, m$ has to hold, but not both, since in this case $m \geq p \cdot q = N$. Without loss of generality we assume $p \,|\, m$ and write $m = p^l r$, with $r \in \mathbb{Z}_N^*$ and $l > 0$.

Again we use the chinese remainder theorem. We know that $ed = 1 \bmod (p-1)(q-1)$ holds. That means there exists a $a \in \mathbb{Z}$ such that $ed + a(p-1)(q-1) = 1$. From this it follows that $ed = 1 \bmod (q-1)$, which we will use in the following:

- $m^{ed} \bmod p = p^{led} r^{ed} \bmod p = 0 = m \bmod p$,
- $m^{ed} \bmod q = p^{led} r^{ed} \bmod q = p^l r^1 = m \bmod q$,

Since m^{ed} has the same remainders modulo p and q as m, $m^{ed} = m$ has to hold.

Remark: Decrypting via the Chinese Remainder Theorem is useful in any case — it allows to compute with numbers of half the size (=bitlength) and thus speeds up computations by roughly a factor of 4!

(d)
- Compute $w = h(m||r) = h(0110||101) = 011$.
- Compute $r^* = r \oplus g_1(011) = 101 \oplus 011 = 110$.
- Compose $x = 0||w||r^*||g_2(w)$. Interpret x as $(0011110011)_2 = (243)_{10}$.
- Return $243^{173} \bmod N = 3$.

Exercise 7.2 Attacks on Textbook-RSA

In this exercise we will study Textbook-RSA and see why it is insecure and why we have to use a variant with random padding in practice.

(a) Suppose $e = 3$ and Alice sends the same message m encrypted to three (or more) different persons having RSA-keys $(N_1, e), (N_2, e), (N_3, e)$. Show how Eve can compute m having only eavesdropped the three ciphertexts c_1, c_2, c_3.

(b) Suppose we want to use Textbook-RSA in a hybrid encryption choosing large enough primes such that $N > 2^{1024}$ and $e = 7$. We want to encrypt AES-keys, i.e. messages from $\{0,1\}^{128}$. Explain why this is a really bad idea!

Solution:

(a) Assume m is sent to three persons having public keys $(N_1, 3)$,$(N_2, 3)$, and $(N_3, 3)$. Then an eavesdropper sees $c_1 = m^3 \bmod N_1$, $c_2 = m^3 \bmod N_2$, $c_3 = m^3 \bmod N_3$. W.l.o.g. $\gcd(N_i, N_j) = 1$ for $i \neq j$ (otherwise we can factor N_i immediately and can recover the message). By using the Chinese Remainder Theorem we know that there is a (unique) solution $x < N_1 \cdot N_2 \cdot N_3$ to the system of three equations:

$$
\begin{aligned}
x &= c_1 && \bmod N_1 \\
x &= c_2 && \bmod N_2 \\
x &= c_3 && \bmod N_3
\end{aligned}
$$

The unique solution x can be computed using the extended Euclidean algorithm. And because $x = m^3 \bmod N_1 \cdot N_2 \cdot N_3$ and $m < \min(N_1, N_2, N_3)$ we can compute m by taking the cube-root of x in the Reals/Integers! $m = x^{1/3}$ (which takes polynomial time using binary search).

(b) If $\mathcal{M} = \{0,1\}^l$ and $N > 2^{l \cdot e}$ then no reduction modulo N takes place when encrypting $m^e \bmod N = m^e = c$ and thus m can be recovered by simply computing the e-th root of c (over the integers instead of in Z_N!).

Exercise 7.3 Elgamal PKES—why to work in $\mathbb{QR}_p$

Construct a CPA-attack on Elgamal relative to $\mathsf{Gen}\mathbb{Z}^*_{\text{safe}}$, i.e. Gen returns

$$I = (\langle \mathbb{Z}_p^*, 1, \cdot \rangle, q, g, x, h)$$

with p a n-bit prime, $q = p - 1$, and g generates all elements in $\mathbb{Z}_p^*$.)
Hint: Consider the observations made about the DDH problem relative to $\mathsf{Gen}\mathbb{Z}^*_{safe}$ in the lecture.

Solution: Eve's strategy $\mathcal{A}$ is as follows:

- $\mathcal{A}((\mathbb{Z}_p^*, q, g, h))$: returns messages $m_0 = g^1$ and $m_1 = g^2$.
- $\mathcal{A}((\mathbb{Z}_p^*, q, g, h), (c_1, c_2))$: Eve checks whether c_2 is a square. If so, she returns $r = 1$, else she returns $r = 0$.

Bob computes (c_1, c_2) by choosing $b \overset{u}{\in} \{0,1\}, y \overset{u}{\in} \mathbb{Z}_q$ and setting $c_1 = g^y$ and $c_2 = h^y \cdot g^{b+1}$. h^y is a square with probability 3/4. In this case, Eve always guesses the correct value of b. If h^y is not a square, she guesses always the wrong bit. Hence her probability of success in the game is 3/4.

<u>Exercise 7.4</u> Elgamal's DSS—why it fails without hashing!

Elgamal already showed in his paper, how to efficiently forge a valid tag for a new message:

- Let (m, r, s) be a valid message-tag pair.
- Choose $A, B, C \in \mathbb{Z}$ s.t. $\gcd(Ar - Cs, p-1) = 1$.
- Set $r' := r^A \cdot g^B \cdot y^C \bmod p$, $s' := sr'(Ar - Cs)^{-1} \bmod p - 1$, and $m' := r'(Am + Bs)(Ar - Cs)^{-1} \bmod p - 1$.

Show that (r', s') is valid for m'.

Solution:

$$\begin{aligned}
y^{r'} r'^{s'} &= y^{r'} (r^A \cdot g^B \cdot y^C)^{(sr'(Ar-Cs)^{-1})} = \\
&= (y^{r'Ar - r'Cs + r'Cs} r^{Asr'} g^{Bsr'})^{(Ar-Cs)^{-1}} = \\
&= ((y^r r^s)^{Ar'} g^{Bsr'})^{(Ar-Cs)^{-1}} = \\
&= g^{(mAr' + Bsr')(Ar-Cs)^{-1}} = g^{m'}
\end{aligned}$$

Thus verification succeeds—the attack has succeeded in forging a tag/message-pair that is accepted! This illustrates the need for a countermeasure (like cryptographic hash-functions) to "destroy" the algebraic properties of the signature!