Cryptography and Security — Final Exam

Serge Vaudenay

12.1.2011

- duration: 3h
- no documents is allowed
- a pocket calculator is allowed
- communication devices are not allowed
- answers to every exercise must be provided on separate sheets
- readability and style of writing will be part of the grade
- do not forget to put your name on every sheet!

Family Name: ....

Given Name: ....

Section: ....

SCIPER: ....

1 3-Collisions

Let f be a random-looking function from a set $\mathcal{X}$ to a set $\mathcal{Y}$. Let N denote the cardinality of $\mathcal{Y}$. We call an r-collision a set $\{x_1, \ldots, x_r\}$ of r elements of $\mathcal{X}$ such that $f(x_i) = f(x_j)$ for every i and j.

Q.1 Recall what preimage resistance and collision resistance mean.

Q.2 Name the ideas behind two collision finding algorithms from the course and give their time and memory complexity.

Q.3 Let $y \in \mathcal{Y}$ be a target value. Provide an algorithm $\mathcal{A}_1$ such that upon input y it returns $x \in \mathcal{X}$ such that $f(x) = y$ with average complexity N in terms of f evaluations.
Make the complexity analysis.

Q.4 By using $\mathcal{A}_1$ as a subroutine, provide an algorithm $\mathcal{A}_2$ producing r-collisions with complexity rN in terms of f evaluations.
Make the complexity analysis.

Q.5 We consider an algorithm $\mathcal{A}_3$ for making r-collisions, defined by two parameters α and β. The algorithm works in two phases. In the first phase, it picks N^α random $x \in \mathcal{X}$ and stores $(f(x), L_{f(x)})$ in a hash table, where $L_{f(x)}$ is a list initialized to the single element x. In the second phase, it iteratively picks N^β random $x \in \mathcal{X}$. For each of these x's, it looks whether $y = f(x)$ has an entry in the hash table. If it does, and if x is not already in the list L_y, x is inserted into the list L_y. If L_y has r elements, the algorithm output L_y. We assume that $\mathcal{A}_3$ never picks the same x twice.

for $i = 1$ to N^α **do**
    pick a new x at random
    set $y = f(x)$ and store $(y, (x))$ at place $h(y)$
end for
for $i = 1$ to N^β **do**
    pick a new x at random
    if there is an entry (y, L_y) at place $h(f(x))$ such that $y = f(x)$ **then**
        insert x in list L_y
        if L_y has size r **then**
            yield L_y and stop
        end if
    end if
end for
algorithm failed

Q.5a Show that $\mathcal{A}_3$ either generates r-collisions or fails.

Q.5b Show that the memory complexity is $M = O(N^{\alpha} r \log N)$ and that the time complexity in terms of f evaluations is $T = N^{\alpha} + N^{\beta}$.

In what follows we will approximate $T \approx \max(N^{\alpha}, N^{\beta})$ and $M \approx N^{\alpha}$.

Q.5c For $r = 2$, which inequality shall α and β satisfy to reach a constant probability of success?
For $r = 3$, show that this inequality becomes $\alpha + 2\beta \geq 2$.
Hint: apply the birthday paradox in Phase 2.

Q.5d Show that for parameters for $r = 3$ reaching a constant probability of success, $\log T$ is a function in terms of $\log M$.
Plot its curve.

Q.6 We consider another algorithm $\mathcal{A}_4$ for making 3-collisions, defined by parameters α and β. Now, $\mathcal{A}_4$ runs N^α times a collision-finding algorithm and stores the N^α obtained collisions in the same form (y, L_y) with $L_y = (x_1, x_2)$ as before. In a second phase, $\mathcal{A}_4$ picks N^β random x and check if $f(x)$ hits one of the y in the hash table. If it is the case, a 3-collision is found. (We assume that no x is picked several times.)

for $i = 1$ to N^α **do**
  run a collision-finding algorithm and get x_1 and x_2
  set $y = f(x_1)$ and store $(y, (x_1, x_2))$ at place $h(y)$
end for
for $i = 1$ to N^β **do**
  pick a new x at random
  if there is an entry (y, L_y) at place $h(f(x))$ such that $y = f(x)$ **then**
    insert x in list L_y
    yield L_y and stop
  end if
end for
algorithm failed

Q.6a Show that the memory complexity is $M \approx N^\alpha$ and that the time complexity in terms of f evaluations is $T \approx \max(N^{\alpha+\frac{1}{2}}, N^\beta)$.

Q.6b Show that for $\alpha + \beta \geq 1$ we obtain a constant probability of success.
Plot the curve of minimal $\log T$ in terms of $\log M$ to reach a constant probability of success.
Compare with $\mathcal{A}_3$.
When is it better?

2 Attack on some Implementations of PKCS#1v1.5 Signature with $e = 3$

Family Name: ....
Given Name: ....
Section: ....
SCIPER: ....

In this exercise we represent bitstrings in hexadecimal by grouping bits into packets of 4, each packet (nibble) being denoted in hexadecimal with a figure between 0 and F. For instance, 2B represents the bitstring 0010 1011. Given a bitstring x, we denote by $\bar{x}$ the integer such that x is a binary expansion of $\bar{x}$. For instance, $\overline{00\,\mathsf{FF}} = 255$.

We call a *cube* an integer whose cubic root is an integer.

Given a message m and an integer ℓ_N, we define the bitstring of length ℓ_N

$$\mathsf{format}_{\ell_N}(m) = 00\,01\,\mathsf{FF}\cdots\mathsf{FF}\,00 \| D(m)$$

where $D(m)$ represents the identifier of the hash function H together with $H(m)$ following the ASN.1 syntax. As an example, in the SHA-1 case, we have

$$D(m) = 30\,21\,30\,09\,06\,05\,2\mathsf{B}\,0\mathsf{E}\,03\,02\,1\mathsf{A}\,05\,00\,04\,14 \| \text{SHA-1}(m)$$

We denote by ℓ_D the bitlength of $D(m)$.

We recall that the PKCS#1v1.5 signature for a message m and a public key (e,N) is an integer s such that $0 \leq s < N$ and $s^e \bmod N$ can be parsed following the format $\mathsf{format}_{\ell_N}(m)$, where ℓ_N is the minimal bitlength of N. It is required that the padding field consisting of FF bytes is at least of 8 bytes.

Throughout this exercise we assume that $e = 3$.

Q.1 What is a signature scheme? Describe its components, its functionality, and give an intuition on its security.

Q.2 What is a valid signature for a message m in PKCS#1v1.5? Detail the verification algorithm.

Q.3 Let $u = \mathsf{format}_{\ell_N}(m)$.

Q.3a If $\overline{u}$ is a cube, show that we can easily forge a signature for m without any secret information.

Q.3b We assume that $\overline{u}$ looks like a random number less than $a = 2^{\ell_N - 15}$. How many cubes are less than a?

What is the probability for $\overline{u}$ to be a cube?

Q.3c Deduce an algorithm to forge a signature for m which works with a success probability $2^{-\frac{2}{3}\ell_N + 10}$.

It this practical?

Q.4 Bleichenbacher observed that some parsers just scan the bytes from the formatting rule but do not check that the string terminates after the digest. That is, these implementations accept the following format

$$\mathsf{00\,01\,FF\cdots FF\,00}\|D(m)\|g$$

where g is any garbage string, provided that the padding field has at least 8 bytes and that the total length (including the garbage) is ℓ_N.

In this question we assume $\ell_N = 3\ell$. We further assume that $\ell_N \geq 84 + 6\ell_D$.

Q.4a Let $P = \mathsf{FF}\cdots\mathsf{FF}$ be a string of FF bytes with bitlength ℓ_P. Show that the ℓ_N-bit string $u = \mathsf{00\,01}\|P\|\mathsf{00}\|D(m)\|\mathsf{00}\cdots\mathsf{00}$ is such that $\overline{u} = 2^{3\alpha} - x2^{\gamma}$ for some integer x, where $\alpha = \ell - 5$ and $\gamma = \ell_N - 24 - \ell_D - \ell_P$.

Q.4b By using the assumption $\ell_N \geq 84 + 6\ell_D$, show that we can select ℓ_P such that $\gamma \geq 2\alpha$ and $x \leq 2^{\frac{1}{2}(3\alpha-\gamma)}$.

Q.4c We assume that $x \bmod 3 = 0$. Let $y = \frac{1}{3}x2^{\gamma-2\alpha}$ and $s = 2^{\alpha} - y$. Show that $\overline{u} \leq s^3 < \overline{u} + 2^{\gamma}$.

Q.4d Deduce an algorithm to forge signatures on a random message m with success probability $\frac{1}{3}$ based on Bleichenbacher's observation when 3 divides ℓ_N and $\ell_N \geq 84 + 6\ell_D$.

Q.4e Finally, apply the attack to $\ell_N = 3072$ with SHA-1. Show that the attack applies and that

$$s = 2^{1019} - \frac{1}{3}(2^{288} - \overline{D(m)})2^{34}$$

is a valid signature with probability $\frac{1}{3}$ over the random selection of the message.