Family Name:..........................

First Name:...........................

Section:...............................

Security and Cryptography

Final Exam - Solutions

January 12^{th}, 2009

Duration: 4 hours

This document consists of 16 pages.

Instructions

Electronic comunication devices and documents are *not* allowed.

Other electronic devices are permitted.

Answers must be written on the exercises sheet.

This exam contains 4 *independent* exercises.

Answers can be either in French or English.

Questions of any kind will certainly *not* be answered. Potential errors in these sheets are part of the exam.

You have to put your full name on the first page and have all pages *stapled*.

1 Impact of Moore's Law on Massive Bruteforce Projects

We assume that the number of elementary operations per second of an up to date computer sold at time t is $f_0 e^{\frac{t}{\tau}}$ and that its price is a constant c. A spook agency is hiring some new cryptographer to run a massive bruteforce cryptanalysis project. The project starts at time $t = 0$ and is bound to complete at time $t = t_c$. The agency wants to run an exhaustive key search. We assume that a computer needs r elementary operations to try a single key.

The project consists of making all bought computers participate in the exhaustive search. Let M denote the total budget to be spent. Let K be the total number of keys which can be tried during the entire project.

Let $b(t)$ be the number of machines running at time t. We assume that machines can all run until time t_c whenever they start running, so $b(t)$ is increasing and depends on the *strategy* to buy new machines. We will compare the following strategies:

- The "startup funds strategy" consists of using the budget right away. In terms of b this translate into a constant function $b(t) = \alpha$.
- The "sleeping strategy" consists of waiting without spending until time $t = t_1$ then buying all machines. In terms of b this translate into a locally constant function $b(t) = 0$ if $t < t_1$ and $b(t) = \alpha$ if $t \geq t_1$.
- The "continous budget strategy" consists of regularly buying computers. In terms of b this translate into a linear function $b(t) = \lambda t$. (For this strategy we model $b(t)$ by a continuous function.)

For numerical applications we will take the following constants

$$\begin{array}{rcc} \tau = \frac{18}{\ln 2}\text{months} & f_0 = 3\text{GHz} & c = 5\,000\text{CHF} \\ t_c = 30\text{years} & r = 200 & M = 10\,000\,000\,000\text{CHF} \end{array} \tag{1}$$

1. If we buy a computer at time t and make it run until time t_c, express how many keys K it will be able to try in terms of f_0, τ, t_c, r.

$$K = \frac{(t_c - t) f_0 e^{\frac{t}{\tau}}}{r}.$$

2. For the startup funds strategy, express the optimal parameter α in terms of M and c. Express K in terms of $f_0, \tau, t_c, r, \alpha$. Using (1), do the numerical analysis.

> We want to buy as many machines as possible, thus $\alpha = \frac{M}{c}$.
> Numerically, we obtain $\alpha = 2 \times 10^6$.
> So we can try $\alpha K(t=0)$ keys:
>
> $$\alpha K(t=0) = \alpha \frac{t_c f_0}{r} = 2.84 \times 10^{22}$$

3. For the sleeping strategy, express the optimal parameter α in terms of M and c. Express K in terms of $f_0, \tau, t_c, r, M, t_1$. Find the optimal t_1. Using (1), do the numerical analysis.

> We also want to buy as many machines as possible, thus $\alpha = \frac{M}{c}$.
> Numerically, we obtain $\alpha = 2 \times 10^6$.
> So we can try $\alpha K(t=t_1)$ keys:
>
> $$\alpha K(t=t_1) = \alpha \frac{(t_c - t_1) f_0 e^{\frac{t_1}{\tau}}}{r}$$
>
> the optimal t_1 is given when the derivative is 0:
>
> $$(\alpha K(t=t_1))' = \frac{\alpha f_0}{cr\tau}(t_c - t_1 - \tau) e^{\frac{t_1}{\tau}}$$
>
> and we obtain:
>
> $$(\alpha K(t=t_1))' = 0 \Longrightarrow t_1 = t_c - \tau$$
>
> Doing the numerical analysis, we find:
>
> $$t_1 = 334.031\text{month}, \quad K = 7.9 \times 10^{26}$$

4. Assuming that $b(t)$ is now a derivable function such that $b(0) = 0$, show that

$$K = \frac{f_0}{r} \int_0^{t_c} \frac{t - (t_c - \tau)}{\tau} e^{\frac{t}{\tau}} b(t)\, dt \tag{2}$$

Suppose we buy b_i computers at time t_i. These computers compute $b_i K(t = t_i)$ keys so the number of keys computed at $t = t_c$ is

$$K = \sum_{i=0}^{t_c} b_i K(t = t_i)$$

For each interval dt we have $b_i = b(t_i + dt) - b(t_i) = b'(t)\, dt$, thus

$$\begin{aligned} K &= \int_0^{t_c} K(t) b'(t) dt \\ &= \int_0^{t_c} \frac{(t_c - t)}{r} f_0 e^{\frac{t}{\tau}} b'(t)\, dt \quad \text{(integrate by parts)} \\ &= \left[\frac{t_c - t}{r} f_0 e^{\frac{t}{\tau}} b(t) \right]_0^{t_c} + \frac{f_0}{\tau} \int_0^{t_c} \frac{t - (t_c - \tau)}{r} e^{\frac{t}{\tau}} b(t)\, dt \\ &= \frac{f_0}{r} \int_0^{t_c} \frac{t - (t_c - \tau)}{\tau} e^{\frac{t}{\tau}} b(t)\, dt \end{aligned}$$

since $b(0) = 0$ and $t_c - t = 0$ when $t = t_c$.

5. For the continous budget strategy, express the optimal parameter λ in terms of M and c. Express K in terms of f_0, τ, t_c, r, M. Using (1), do the numerical analysis.

As $b(0) = 0$, the number of machines at time t_c is $b(t_c)$. Since we want to maximize the number of machines, we use the whole budjet

$$b(t_c) = \frac{M}{c} \implies \lambda t_c = \frac{M}{c} \implies \lambda = \frac{M}{ct_c}$$

Numerically, we find $\lambda = 2.11 \times 10^3$ computers/sec.
Using the result from the previous question,

$$\begin{aligned} K &= \frac{f_0}{r} \int_0^{t_c} \frac{t - (t_c - \tau)}{\tau} e^{\frac{t}{\tau}} b(t) dt \\ &= \frac{\lambda f_0}{r} \int_0^{t_c} \frac{(t - (t_c - \tau))\, t}{\tau} e^{\frac{t}{\tau}} dt \\ &= \frac{\lambda f_0}{r} \left[\left(t^2 - (t_c - \tau) t + (t_c + \tau)\tau \right) e^{\frac{t}{\tau}} \right]_0^{t_c} \\ &= \frac{\lambda f_0}{r} \left(\tau^2 e^{\frac{t_c}{\tau}} - (t_c + \tau)\tau \right) \\ &= 1.55 \times 10^{26} \end{aligned}$$

6. Deduce from (2) that $b(t)$ becomes optimal as it tends towards the function b_{opt} defined by $b_{\mathsf{opt}}(t) = 0$ for all $t < t_c - \tau$ and $b(t)_{\mathsf{opt}} = \alpha$ for $t > t_c - \tau$. Further deduce that the sleeping strategy is the best one.

Equation (2) can be written as:

$$K = \frac{f_0}{r} \int_0^{t_c - \tau} \frac{t - (t_c - \tau)}{\tau} e^{\frac{t}{\tau}} b(t)\, dt + \int_{t_c - \tau}^{t_c} \frac{t - (t_c - \tau)}{\tau} e^{\frac{t}{\tau}} b(t)\, dt$$

We treat the two integrals separately:

- Since $b(t) geq 0$, we have $\frac{t-(t_c-\tau)}{\tau} e^{\frac{t}{\tau}} b(t) \leq 0$ for $t \in [0, t_c - \tau]$. Thus,

  $$\int_0^{t_c - \tau} \frac{t - (t_c - \tau)}{\tau} e^{\frac{t}{\tau}} b(t)\, dt \leq 0$$

  and is maximal when equal to 0, which happens when $b(t) = 0$.
- On the other hand, we have:

  $$\int_{t_c - \tau}^{t_c} \frac{t - (t_c - \tau)}{\tau} e^{\frac{t}{\tau}} b(t)\, dt$$

  which is maximum when $b(t) \leq \alpha$ is maximum.

We deduce then that $b(t) = b_{\mathsf{opt}}(t)$ is the optimal. It corresponds to the sleeping strategy.

2 Homomorphic Signatures

The purpose of this exercise is to study the security of a digital signature scheme. The scheme is defined as follows:

- **Setup:** Two prime numbers p, q such that $p = 1 \mod q$ and an element $g \in \mathbb{Z}_p^\star$ of order q,
- **Key Generation:** The secret key is a tuple $sk = (x, y)$ and the corresponding public key is $pk = (pk_1, pk_2) = (H(x), H(y))$ where $H(\alpha) = g^\alpha \mod p$.
- **Signature:** Given a message $m \in \mathbb{Z}_q$. The signer computes

$$\sigma = x + m \times y \mod q$$

p, q, g, pk are made public while $sk = (x, y)$ is kept secret by the signer.

1. What is the problem preventing us from inverting H? Is H a trapdoor function?

Discrete Logrithm. No.

2. Give a typical size of q in bits.

160 (accept between 128 and 512)

3. Give an algorithm to find g.

Take $h \in_R \mathbb{Z}_p^\star - \{1\}$ and compute $g = h^{\frac{p-1}{q}}$ until $g \neq 1$

4. Give an algorithm to generate p and q. What is its complexity (heuristically)?

```
while true do
    Pick at random q
    if q is prime then
        Pick at random a
        if p = aq + 1 is prime then
            return (p, q).
        end if
    end if
end while
```

Complexity: $O(\ell^5)$ where ℓ is the bit-length of p.

5. How does the signature verification work?

It works by checking that:

$$H(\sigma) \stackrel{?}{=} pk_1 \times pk_2^m \mod p$$

6. Show that if an adversary has access to the signatures of two different messages then he is able to retrieve the secret key.

An adversary obtaining two pairs (m_1, σ_1) and (m_2, σ_2) can write:

$$\left\{ \begin{array}{lcl} \sigma_1 & = & x + m_1 \times y \mod q \\ \sigma_2 & = & x + m_2 \times y \mod q \end{array} \right. \implies \left\{ \begin{array}{lcl} y & = & (\sigma_1 - \sigma_2) \times (m_1 - m_2)^{-1} \mod q \\ x & = & \sigma_1 - m_1 \times y \mod q \end{array} \right.$$

3 MAC From Hash Functions

In this exercise, we will study the security of some MAC constructions based on hash functions. Through this exercise, we will consider a hash function H based on an iterated hash function H_0 with ℓ-bit block messages and the Merkle-Damgård strenghtening pad. That is, $m\|\mathsf{pad}(m)$ has a length multiple of ℓ and $H(m) = H_0(m\|\mathsf{pad}(m))$. (See fig.1.)

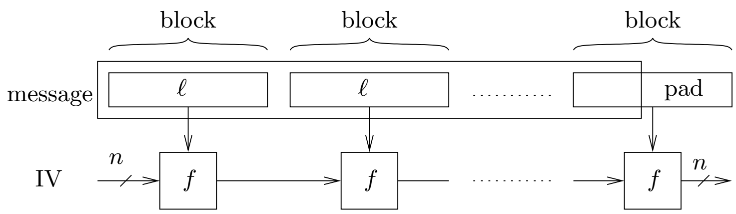

Figure 1: The Merkle-Damgård scheme

1. Recall the standard padding scheme $\mathsf{pad}(m)$ (or an equivalent one).

> Use $1\|0\ldots0\|\delta$ where the number of 0's is the smallest positive λ such that $\delta + 1 + \lambda + 64 \equiv 0 \mod \ell$ and δ denotes the bit-length of the message encoded on 64 bits.

2. Recall three different security properties of a cryptographic hash function.

> (a) Resistance to preimage attack
> (b) Resistance to 2nd preimage attack
> (c) Resistance to collisions

3. What is a MAC forgery? Detail the various security models with respect to the Oracles to which the adversary has access.

A MAC forgery is an attack in which an adversary *not* knowing the secret key K tries to forge a pair (m, c) such that $\mathsf{Verify}_K(m, c) = 1$.
The security models for such an attack are:

(a) known message attack

(b) chosen message attack

4. **The Prefix Method**. We consider the MAC algorithm defined given a ℓ-bit secret key K and a message m as

$$\mathsf{MAC}_K(m) = H_0(K \| m \| \mathsf{pad}(m))$$

where $\|$ denotes the concatenation operation.

Show that given the MAC of a known message m the adversary is able to output a forgery on a message $m' \neq m$.

Let us define the message $m' = m \| \mathsf{pad}(m) \| m^\star$ for any message $m^\star$.
Then the adversary can compute the MAC of m' by hashing $m^\star$ and using $\mathsf{IV} = \mathsf{MAC}_K(m)$.

5. **The Suffix Method**. Let us consider another variant defined given a secret key K and a message m as

$$\mathsf{MAC}_K(m) = H_0(m \| \mathsf{pad}(m) \| K)$$

Show that if H is *not* collision-resistant and the adversary has access to an Oracle to which he can submit a *chosen* message m and get $\mathsf{MAC}_K(m)$ then he is able to forge the MAC of a message $m' \neq m$.

Notice first that

$$H(m_1 \| \mathsf{pad}(m_1)) = H(m_2 \| \mathsf{pad}(m_2)) \implies \mathsf{MAC}_K(m_1) = \mathsf{MAC}_K(m_2)$$

The attack then consists of submitting m_1 to the MAC Oracle and get $\mathsf{MAC}_K(m_1)$ which is also the MAC of m_2.

4 Broadcast Encryption and Traitor Tracing

This exercise will be graded separately, like the continuous evaluation surveys.

1. The problem of broadcast encryption schemes is that...

   - ☐ it uses symmetric encryption only.
   - ☐ it is vulnerable to exhaustive search.
   - ■ the content decryption key has to be sent to a precise list of non-revoked receivers.
   - ☐ it is not implemented so far.

2. What is the disadvantage of common stateful broadcast encryption schemes?

   - ☐ Receiver cannot be revoked.
   - ■ Sender and receiver must be synchronized.
   - ☐ They are patented.
   - ☐ Content provider cannot monitor receivers.

3. Which one of the following schemes is *not* a broadcast encryptions scheme?

   - ☐ SSD (Goodrich, Sun and Tamassia).
   - ■ SAS (Pasini and Vaudenay).
   - ☐ LSD (Halevi and Shamir).
   - ☐ SD (Naor, Naor and Lotspiech).

4. Tick the *false* statement about the Advanced Access Content System (AACS).

   - ☐ BlueRay disks use AACS.
   - ☐ AACS is based on the Naor-Naor-Lotspiech scheme.
   - ☐ HD-DVDs use AACS.
   - ■ AACS is a stateful broadcast encryption scheme.

5. The purpose of traitor tracing schemes is...

   - ■ to discourage subscribers from giving away their keys.
   - ☐ to detect pirate decoders.
   - ☐ to send a video signal to a large number of subscribers.
   - ☐ to make pirate decoders decrypt a wrong signal.

6. Tick the *correct* statement. A k-collusion resistant traitor tracing scheme...

   - ☐ makes sure that no subset of k subscribers receive the same key.
   - ■ can identify a coalition of $k - 1$ subscribers who released a decryption key.
   - ☐ is such that no subset of k subscribers can decrypt the broadcast signal.
   - ☐ always remains secure if we have $k + 1$ traitors.

7. Which of these technologies was not broken so far?

   - ☐ TEA in XBOX.
   - ☐ CRYPTO1 in Mifare RFID tag.
   - ☐ Keeloq for cars.
   - ■ ECDSA in MRTD.

8. Tick the item *unrelated* to SPA in cryptography.

   - ☐ Attacks based on power consumption analysis.
   - ■ Medication based on water treatment.
   - ☐ Secure password authentication.
   - ☐ Simple power analysis.

9. Tick which item was *not* used for a side-channel attack so far.

   - ☐ Time.
   - ■ Color.
   - ☐ Power.
   - ☐ Faults.

10. If my quiz has 7 correct and 3 incorrect answers my grade is...

    - ☐ 5
    - ☐ 4.5
    - ☐ 3.5
    - ■ 3

Any attempt to look at
the content of these pages
before the signal
will be severly punished.

Please be patient.