Advanced Cryptography — Final Exam

Serge Vaudenay

21.6.2023

- duration: 3h
- any document allowed
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will **not** answer any technical question during the exam
- readability and style of writing will be part of the grade
- writing with pencil is not allowed

1 The Even-Mansour Cipher

In this exercise we consider a block cipher over n-bit blocks, which uses a $2n$-bit key (K_1, K_2) and defined by

$$\mathsf{Enc}_{K_1,K_2}(x) = \pi(x \oplus K_1) \oplus K_2$$

where π is a known permutation of the set $\{0,1\}^n$. In the adversarial model, the adversary is allowed to make D queries to a chosen plaintext/ciphertext oracle (that is, the adversary selects the direction for each query§ — either encryption or decryption — and thei input block, then gets either the encryption or the decryption of that block depending on the selected direction) and T queries to an oracle implementing π and π^{-1} (that is, the adversary selects the direction and the input and gets the image of that input by either π or π^{-1} depending on the selected direction). We consider key recover attacks: the goal of the adversary is to recover the hidden key (K_1, K_2).

Q.1 Let $\Delta \in \{0,1\}^n$ be a non-zero constant. We consider an adversary making D random pairs (x_i, x'_i), $i = 1, \ldots, D/2$, such that $x'_i \oplus x_i = \Delta$. The adversary makes D chosen plaintext queries to get $y_i = \mathsf{Enc}_{K_1,K_2}(x_i)$ and $y'_i = \mathsf{Enc}_{K_1,K_2}(x'_i)$, $i = 1, \ldots, D/2$. Then, the adversary takes T random pairs (u_j, u'_j), $j = 1, \ldots, T/2$, such that $u'_j \oplus u_j = \Delta$, and queries the other oracle to get $v_j = \pi(u_j)$ and $v'_j = \pi(u'_j)$, $j = 1, \ldots, T/2$.
How to select D and T to have good chances for a pair $(i.j)$ to exist such that $u_j = x_i \oplus K_1$?

Q.2 How can the adversary isolate possible values for this pair (i, j) and estimate the expected number of incorrect values?

Q.3 Deduce a key recovery attack and estimate the success probability when DT is proportional to 2^n.

2 Finding Heavy Differentials

Throughout this exercise, n denotes an integer and p denotes a probability. In asymptotic analysis, n goes to infinity and p may depend on n. Given a function $f : \{0,1\}^n \to \{0,1\}^n$ and $\alpha, \beta \in \{0,1\}^n$, we define $\mathsf{DP}_f(\alpha, \beta) = \Pr[f(X \oplus \alpha) = f(X) \oplus \beta$, where $\oplus$ is the bitwise exclusive OR and $X \in \{0,1\}^n$ is uniform. When x is such that $f(x \oplus \alpha) = f(x) \oplus \beta$, we say that x *follows* the characteristic (α, β). We say that (α, β) is a *heavy* characteristic if $\mathsf{DP}_f(\alpha, \beta) > p$. The objective of this exercise is to find heavy characteristics by having a black-box access to f and no other information about f. We assume that one memory register can store a value in $\{0,1\}^n$ and that an operation over elements of this set cost one unit of time complexity.

Q.1 Design an algorithm with oracle access to f which is able to find heavy characteristics with time complexity $\mathcal{O}(2^{2n})$ and memory $\mathcal{O}(2^{2n})$.

Q.2 Given $\gamma \in \{0,1\}^n$, we define $g_\gamma(x) = f(x \oplus \gamma) \oplus f(x)$. We assume that when $X \in \{0,1\}^n$ is uniformly distributed, then the events "x follows (α, β)" and "$x \oplus \gamma$ follows (α, β)" are independent. When both events occur, we say that x is *good* for (α, β).
If (α, β) is heavy, prove that X is good for (α, β) with probability at least p^2 and that when such event occurs, then $g_\gamma(x) = g_\gamma(x \oplus \alpha)$.

Q.3 Given a heavy characteristic (α, β), if we pick $k = \left\lceil \sqrt{n} 2^{\frac{n}{2}} p^{-1} \right\rceil$ random values $x_1, \ldots, x_k$, show that except with negligible probability, there exist $\frac{n}{4}$ pairs (i, j) such thats $x_j = x_i \oplus \alpha$ and x_i is good. (Give a heuristic argument.)

Q.4 Complete the following algorithm and show that it can find heavy characteristics, except with negligible probability, and complexity lower than before. Precisely analyze the complexity.

pick $x_1, \ldots, x_k \in \{0,1\}^n$ at random for $k = \left\lceil \sqrt{n} 2^{\frac{n}{2}} p^{-1} \right\rceil$
initialize an array $\mathsf{Inv}[.]$ and the list L to empty
for $i = 1$ to k **do**
    $y \leftarrow g_\gamma(x_i)$
    insert x_i in the list $\mathsf{Inv}[y]$
    if $\mathsf{Inv}[y]$ has at least 2 elements **then** insert y in L
end for
initialize $v\{.,.\}$ to an empty dictionary and L' to the empty list
for each y in L **do**
    for each (x_i, x_j) pair of element of $\mathsf{Inv}[y]$ **do**
        $\alpha \leftarrow x_j \oplus x_i$, $\beta \leftarrow f(x_j) \oplus f(x_i)$
        if $v\{\alpha, \beta\}$ exists **then**
            $v\{\alpha, \beta\} \leftarrow v\{\alpha, \beta\} + 1$
        else
            $v\{\alpha, \beta\} \leftarrow 1$
        end if
        if $v\{\alpha, \beta\} \geq \frac{n}{4}$ **then** insert (α, β) in L' and abort the **for** loop
    end for

end for

...

3 Blind Signatures

We consider a blind signature primitive which is defined by the following algorithms:

- $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{sk}, \mathsf{pk})$ where λ is the security parameter;
- $\mathsf{SignC1}(\mathsf{pk}, m) \to (\mathsf{st}, \mathsf{query})$ where m is a message (bitstring);
- $\mathsf{SignS}(\mathsf{sk}, \mathsf{query}) \to \mathsf{resp}$;
- $\mathsf{SignC2}(\mathsf{st}, \mathsf{resp}) \to \sigma$;
- $\mathsf{Verify}(\mathsf{pk}, m, \sigma) \to \mathsf{true}/\mathsf{false}$.

When algorithms are executed in this order, correctness ensures that Verify returns true. The idea is that the signing process is run by the interaction between a client and a server. The server has the signing key sk and has authority to sign. The client knows which message m is to be signed but the server does not. The security notions are that signatures should be unforgeable (in a sense to specify in a question below) and query and σ should be unlinkable (in a sense to specify).

Q.1 Recall the EF-CMA security notions and explain why it does not fit to blind signatures.

Q.2 We try to formalize unforgeability by the notion of one-more forgeries. Following this game, the adversary wins by showing more signed messages than the number of queries to a $\mathsf{SignS}(\mathsf{sk}, .)$ oracle. Properly define the one-more forgery game and formalize security with respect to this notion.

Q.3 Formalize the notion of unlinkability, where the adversary is now the server.

Q.4 We tweak RSA so that it fits the notion of blind signature. We define KeyGen as in RSA and $\mathsf{SignS}(\mathsf{sk}, \mathsf{query}) = \mathsf{query}^d \bmod N$, where $\mathsf{sk} = (N, d)$. Propose some algorithms for $\mathsf{SignC1}$, $\mathsf{SignC2}$, and Verify in order to obtain a blind signature which is one-time unforgeable and unlinkable. (Give arguments for the security, no formal proof is required but insecure solutions will have a lower grade.)