Exam Paper, Multiplicative Complexity, Cryptography and Cryptanalysis

- University of Warsaw, PhD Open Course -

Nicolas T. Courtois, January 2013

Solve exactly **THREE out of four exercises**. for example 1+2+3 or 2+3+4. Question 1 should be solved in a standard way and can be omitted if the student is able to solve 2+3+4. It is NOT acceptable to solve all questions 1+2+3+4, student must indicate which questions will be graded.
In contrast Questions 2,3 and 4 should be solved in many different ways and grading will be based on originality: and each student should try to find their own way. It equally to acceptable to propose one complete solution or several incomplete solutions: for example: maybe this works... but what is missing is this... Try to be precise about what is needed. Maybe another researcher can solve it: find the exact missing piece of the puzzle.
The solution paper should not exceed 4 pages in A4 with 10pt font or equivalent in handwriting and scanned into a pdf. It is nice to get also the Latex source (not required).
Email: **n.courtois MALPA cs.ucl.ac.uk**

Exercise 1. Let $GF(2)$ be a field with two elements.

1. Look at these two polynomials X^3+1, X^3+X+1, which one is irreducible in $GF(2)[X]$? Let $P(x)$ be this polynomial and $P'(X)$ the other. Give a complete proof that $P(X)$ is indeed irreducible.

2. Factor $P(X)$ and $P'(X)$ in $\mathbb{F}_2[X]$.

3. We define a field F as the set of all monomials modulo $P(X)$, or in other terms $F = GF(2)[X]/P(X)$, with the addition and the multiplication of polynomials modulo $P(X)$. How many elements has this field?

4. How many solutions in F has the equation $x^2 = x$? Write all or them and prove that there is no more.

5. Compute $1, X, X^2, \ldots$ modulo $P(X)$.

6. Is P a primitive polynomial?

7. How many solutions in $GF(2)$ has the equation $x^2 = 1$?

8. How many solutions in F? Prove that there is no more.

9. And in $GF(3)$?

10. And in Z_n with $n = pq$ a product of two large primes? (Hint: Chinese Remainder Theorem).

11. And in Z_6?

12. And in Z_4?

Exercise 2. Let $GF(2)$ be a field with two elements.
Let $F = Inv$ be the full-size inverse function with the usual $0 \mapsto 0$:

$$Inv(X) = \begin{cases} X^{-1} & \text{in } GF(2^n) \text{ if } X \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Let $F = T \circ \Psi \circ Inv$. Where Ψ is a natural mapping of $GF(2^n)$ into $GF(2)^n$ which comes from the implementation of the finite field $GF(2^n)$. T is a known multivariate affine transformations, given by a $n \times n$ matrix with coefficients in $GF(2)$ and a vector in $GF(2)^n$.
We define the Multiplicative Complexity (MC) of a function as the minimum number of AND gates needed to implement that function.
Let $n = 3$. Prove that $MC(Inv) = 3$ Any proof is acceptable, can be done with a computer or by hand. Originality is important: every student can try to prove in a different way or propose several methods. You need however to convince the reader/marker that your proof is correct.

In the same way try to prove that $MC(Inv) = 5$ for $n = 4$.

More generally try to prove some lower and upper bounds on MC for other values of n.

Exercise 3. Attack on an cipher based on affine equiv. of Inv. We consider the following symmetric encryption system:
Let

$$F = T \circ \Psi \circ Inv \circ \Psi^{-1} \circ S$$

where S, T are two SECRET multivariate affine transformations, each given by a $n \times n$ matrix with coefficients in $GF(2)$ and a vector in $GF(2)^n$. Let $n = 32$. We assume the most general adaptive chosen plaintext and chosen ciphertext attack. The attacker can get encryptions and decryptions of any attack.
Compute the key size of this cipher.
Propose a key recovery attack which is faster than 2^{100}.
If this is too difficult, try to break at least the case with $T \circ S = Id = S \circ T$.

rounds	1							8	9							16
keys	$\mathbf{k_0}$	k_1	k_2	k_3	k_4	k_5	k_6	k_7	$\mathbf{k_0}$	k_1	k_2	k_3	k_4	k_5	k_6	k_7
rounds	17							24	25							32
keys	$\mathbf{k_0}$	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_7	k_6	k_5	k_4	k_3	k_2	k_1	$\mathbf{k_0}$

Figure 1: Key schedule in GOST

Exercise 4. GOST is a Feistel cipher with 32 rounds. The block size is 64 bits. The key size is 256 bits. In each round we have a round function $f_k(X)$ with a 32-bit key which uses a 32-bit segment of the original 256-bit key which is divided into eight 32-bit sub-keys $k = (k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7)$. One 32-bit sub-key is used in each round, and their exact order as on Fig. 1.

Each round of a Feistel scheme can be written as $f_k = S \circ F_k : GF(2)^{2n} \rightarrow GF(2)^{2n}$ where k is a key on 32 bits, S is a function which swaps the two halves, $n = 32$, and F_k is defined as $(x, y) \mapsto (x \oplus P(x \boxplus k), y)$, $\oplus$ is a bitwise XOR, $\boxplus$ is an addition modulo 2^{32}, and P is a bijective mapping $GF(2)^n \rightarrow GF(2)^n$.

Show that $Enc_k(X) = \mathcal{E}^{-1} \circ S \circ \mathcal{E}^3$.

A company called Bear Telecom has implemented a protocol. There is an ATM and a smart card. The smart card and the ATM share a secret key k on 256 bits which is unique for this card and both parties know it.

First the ATM wants to prove to the card that is it authentic it sends a pair A, B on 64+64 bits where A is random and B is the encryption of A with the first 16 rounds of GOST.

Then the card needs to prove to the ATM it is authentic, and it sends a pair C, D such that D the encryption of C for the **last** 16 rounds of GOST where C is computed by a deterministic function from A, B so that each time A, B are the same, C is the same and if the attacker can produce one correct pair A, B he can only obtain one pair C, D and not many such pairs.

The card does NOT let the ATM know if the pair A, B was authentic. If the card has obtained an incorrect pair A, B it replies with an incorrect pair C, D which is computed as a deterministic function from A, B. The time to respond and electrical behavior of the card is assumed to be identical in both cases, so that the terminal cannot distinguish if the pair was correct by observing the card. We assume that the whole process of querying the card with A, B and obtaining C, D takes 1 millisecond.

We assume that it takes time of at most $2^{100-2 \cdot k}$ to break GOST given k pairs for 8 rounds of GOST for any $k < 20$. However the attacker has access to 16 rounds of GOST.

The victim inserts his card into a false ATM which is controlled by a criminal. Show that the criminal can recover the secret key of the card and therefore produce a perfect clone of the bank card. In other words argue that it is feasible **in the real life** to recover the full GOST key by a mathematical attack by using standard inputs and outputs of the smart card A, B, C, D.

Evaluate the data, memory and time complexity of the attack.