Exam TMA4160 Cryptography

Suggested solutions

December 15, 2018

This sketch excludes a lot of tedious calculations. In your answers, it would probably have been a good idea to include some of those tedious calculations, or at least explain how they are done.

Problem 1

Given no information, it is reasonable to assume that the message is in English and that the English alphabet is used.

We do an exhaustive search:

AVVLHZF	JEEUQIO	SNNDZRX
BWWMIAG	KFFVRJP	TOOEASY ⟵
CXXNJBH	LGGWSKQ	UPPFBTZ
DYYOKCI	MHHXTLR	VQQGCUA
EZZPLDJ	NIIYUMS	WRRHDVB
FAAQMEK	OJJZVNT	XSSIEWC
GBBRNFL	PKKAWOU	YTTJFXD
HCCSOGM	QLLBXPV	ZUUKGYE
IDDTPHN	RMMCYQW	

The only plausible decryption is TOOEASY. (There is no need to generate every possible decryption, as above. After finding one plausible decryption you could stop.)

[The total effort is writing the alphabet 7 times at most.]

Problem 2

a.

We choose $L = 7 \approx \sqrt{46}$, and compute the table (baby steps):

i	0	1	2	3	4	5	6	7	−7
g^i	1	5	25	31	14	23	21	11	30

Note that the $i = 7, -7$ is not part of the table, but is needed to find the giant step to use.

Then we compute xg^{-jL} (giant steps). First, 43 is not in the table. Next, $43 \cdot 30 = 21 = g^6$ according to the table.

$$43 = (g^7)^1 \cdot g^6 = g^{13}.$$

[6 multiplications modulo 47 and one inversion modulo 47.]

b.

Note that $47 - 1 = 46 = 2 \cdot 23$, so 3 is invertible modulo 46. We get

$$38^3 \cdot 43^5 = 38^3 \cdot 5^{13 \cdot 5} = g^{24} \qquad \Leftrightarrow \qquad 38 = g^{(24-13\cdot 5)/3},$$

or

$$\log_5 38 \equiv (24 - 13 \cdot 5)/3 \equiv 5 \cdot 31 \equiv 17 \pmod{47}.$$

[One multiplication modulo 46 and one inversion modulo 46.]

Problem 3

(Note it is important to use a field for this kind of MAC, otherwise the security could be significantly lower. Also, we do not need a bijection between our alphabet and the field, hence our injection of the English alphabet into the field is not surjective.)

We have that KELP corresponds to $(10, 4, 11, 15)$, while HELP corresponds to $(7, 4, 11, 15)$. In other words

$$3 = 22 - 19 = f((k_1, k_2), \text{KELP}) - f((k_1, k_2), \text{HELP}) = (10 - 7)k_1 = 3k_1 \qquad \Rightarrow \qquad k_1 = 1.$$

Then

$$k_2 = 19 - 7 - 4 - 11 - 15 = 11.$$

Problem 4

a.

This can certainly be done using Gaussian elimination over $\mathbb{F}_2$, but by inspection we find that rows 2, 3 and 4 sum to $(8, 4, 2, 2, 0)$, while rows 1, 4 and 5 sum to $(6, 2, 2, 2, 2)$. (This inspection is slightly easier if we write out the matrix modulo 2 as a $0,1$-matrix.)

(From this, we see that that rows 1, 2, 3 and 5 should also sum to zero modulo 2, which is correct. It is also easy to see that the three rows 2, 4 and 5 are linearly independent over $\mathbb{F}_2$, so there are no more such collections.)

b.

An easy computation modulo 1363 shows that

$$\begin{aligned}
275^2 &\equiv 660 \equiv 2^2 \cdot 3 \cdot 5 \cdot 11 \\
483^2 &\equiv 216 \equiv 2^3 \cdot 3^3 \\
640^2 &\equiv 700 \equiv 2^2 \cdot 5^2 \cdot 7 \\
647^2 &\equiv 168 \equiv 2^3 \cdot 3 \cdot 7 \\
961^2 &\equiv 770 \equiv 2 \cdot 5 \cdot 7 \cdot 11
\end{aligned}$$

[5 squarings modulo 1363.]

c.

We get the two relations

$$\begin{aligned}
(483 \cdot 640 \cdot 647)^2 &\equiv 2^8 \cdot 3^4 \cdot 5^2 \cdot 7^2 \equiv (2^4 \cdot 3^2 \cdot 5 \cdot 7)^2 \text{ and} \\
(275 \cdot 647 \cdot 961)^2 &\equiv 2^6 \cdot 3^2 \cdot 5^2 \cdot 7^2 \cdot 11^2 \equiv (2^3 \cdot 3 \cdot 5 \cdot 7 \cdot 11)^2.
\end{aligned}$$

We find that

$$\gcd(483 \cdot 640 \cdot 647 - 2^4 \cdot 3^2 \cdot 5 \cdot 7, 1363) = 29$$

and that $1363/29 = 47$.

[4 multiplications modulo 1363 and one gcd computation.]

Problem 5

a.

Since ϕ is a ring isomorphism, we see that if (x_p, y_p) and (x_q, y_q) satisfy the curve equation modulo p and q, respectively, then $(x, y) = \phi((x_p, y_p), (x_q, y_q))$ also satisfies the curve equation modulo n.

Likewise, if (x, y) satisfy the equation modulo n, it will also satisfy the equation modulo p and q.

In other words, $((x_p, y_p), (x_q, y_q)) \in U_p \times U_q$ if and only if $(x, y) \in U_n$.

b.

If (1) can be evaluated, then $x_2 - x_1$ is invertible modulo n, which means that $x_2 - x_1$ is non-zero modulo both p and q. This means that P_p and Q_p are distinct points on the curve with distinct X-coordinates, and likewise for P_q and Q_q. So equations corresponding to (1) can be used to compute $P_p + Q_p$ and $P_q + Q_q$.

Then, again because ϕ is a ring isomorphism, it follows that $\phi(P_p + Q_p, P_q + Q_q) = (x_3, y_3)$.

On the other hand, if (1) cannot be evaluated, then $x_2 - x_1$ is not invertible modulo n, which means that $x_2 - x_1$ is zero modulo either p or q (but not both, since $x_1 \neq x_2$). In other words, $\gcd(x_2 - x_1, n)$ is a non-trivial factor of n.

In this case, either P_p and Q_p have the same X-coordinate, or P_q and Q_q have the same X-coordinate.

c.

Since $Q_p = (a-1)P_p$, we get that $Q_p + P_p = (a-1)P_p + P_p = aP_p = \mathcal{O}$, so $Q_p = -P_p$.

Also $Q_q = (a-1)P_q$, but in this case $Q_p + P_p = aP_p \neq \mathcal{O}$, since the order of P_p does not divide a. It follows that P_q and Q_q does not have the same X-coordinates, which means that $x_1 \neq x_2$. (There is a gap in the above argument: If $a - 2 \equiv 0 \pmod{b}$, we get that $Q_q = (a-1)P_q = P_q$ and that $x_1 = x_2$. In this case $\gcd(y_2 - y_1, n)$ would give us a non-trivial factor of n.)

However, since P_p and Q_p have the same X-coordinate, we know that p divides $\gcd(x_2 - x_1, n)$, so $x_2 - x_1$ is not invertible, so (1) can not be evaluated.

Why interesting?

The idea is to choose a random point $P = (x, y)$ in $\mathbb{Z}_n \times \mathbb{Z}_n$ and a random A, and then compute B as

$$B = y^2 - x^3 - Ax.$$

With overwhelming probability, A and B define elliptic curves E_p and E_q as above, while $P \in U_n$, and consequently defines points P_p and P_q.

Now we guess a multiple a of the order of P_p (typically as the factorial of some number), and hope that it will not also be a multiple of the order of P_q.

Now we use the usual equations to compute $Q = \phi((a-1)P_p, (a-1)Q_p)$ (which by **b.** and a similar argument for doubling points, we can do with arithmetic modulo n; if it goes wrong we usually find a factor of n), and if we guess right we find a factor of n.

If we guess wrong, we try again with new A and B.

By the same analysis as we used for the index calculus factoring algorithm, we can show that this factoring algorithm is quite fast.

The most interesting property is that its run-time depends most strongly on the smallest prime factor of n. In other words, if p is small, this algorithm can be very fast. For cryptographic purposes, we will usually not be in this situation, but for non-cryptographic purposes, this can be quite useful.

This algorithm is also one of the generalisations of Pollard's $p - 1$ method that I mentioned briefly in class.