DD2448 Foundations of cryptography

Homework krypto20

Persnr	Name	Email
970731-7559	**Jonas Conneryd**	**conneryd@kth.se**
961026-0995	Alexander Renaudin	aleren@kth.se
961021-5635	Olle Berglöf	oberglof@kth.se

Problem 1.

Task 1.1 (3T). NOT SOLVED Very trivial problem actually. I think I needn't say more than

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

Task 1.2 (3T). NOT SOLVED

Task 1.3 (4T). NOT SOLVED Ganska sÃ¤ker pÃ¥ att det Ã¤r hybrid argument som ska till hÃ¤r.

Problem 2.

Task 2.1 (1T). We use the defintion of Shannon entropy as stated in the course slides:

$$H(X) = -\sum_{x \in X} \Pr[x] \log_2(\Pr[x])$$

and from this expression directly compute the Shannon entropy of X: $H(X) \approx 3.175$.

Task 2.2 (2T). In this assignment, we use the information regarding Huffman encodings in Chapter 2 of [2]. In general, an *encoding* of a random variable X over a binary channel is simply a function $f : X \to \{0,1\}^*$, where we require that f is injective. An encoding allows a series of events of X to be encoded as a bitstring: If $S = x_1, x_2, x_n$ is a string of events of X, we encode S as a bitstring by $S' = f(x_1)|| \ldots f(x_n)$, where $||$ denotes concatenation. If each x_i happens according to a specified probability distribution of X so that

$$\Pr[x_1, \ldots x_n] = \Pi_i^n \Pr[x_i],$$

we say that the string $x_1, \ldots, x_n$ is produced by a *memoryless source*.

The *weighted average length* of an encoding f is

$$l(f) = \sum_{x \in X} \Pr[x]|f(x)|,$$

and is a measure of its efficiency; a shorter weighted average length l would imply that fewer bits would be required to describe a series of events of X by using a string.

The *Huffman encoding* of a random variable X is an injective encoding that minimizes $l(f)$, so with l as a measure of efficiency, the Huffman encoding is optimal. The Huffman encoding of X can be informally described as the encoding achieved by the following algorithm:

1. Set the code of each event to be initially empty.

2. Add ”0” to the code of the least likely event and ”1” to the code of the next least likely event.
3. Combine the least likely and next least likely elements into a single new event whose probability is the the combined probability of the ingoing events. (If this event is assigned a bit, all the ingoing events get that bit added to their code).
4. Go back to step 2 until only a single element remains.
5. Output the resulting encoding of X.

Task 2.3 (1T). Using the algorithm above, we arrive at an average expected length of the Huffman code $\mathsf{LenAvg} \approx 3.73$.

Task 2.4 (1T). Using the Huffman encoding, each event receives an *integer* number of bits as their code, so in a sense the Huffman encoding is a ”binary approximation” of the optimal code for each event. FoRTYDLIGA KANSKE.

Problem 3 (4T). NOT SOLVED

Problem 4.

Task 4.1 (2T). Given a group G_q of order q, a generator g of G, and an element h in G, find the smallest integer n such that $g^n = h$.

Task 4.2 (1T). Given a cyclic group G, for a randomly chosen generator g of G and a randomly chosen h in G, find the smallest integer n such that $g^n = h$.

Task 4.3 (2T). We describe the algorithm here and prove its properties in the next task. We use the approach of page 44 of [1]. We define $\mathcal{A}'$ as follows:
$\mathcal{A}'(g, h)$:

1. Choose $x \in \mathbb{Z}/q\mathbb{Z} - \{0\}$ and $y \in \mathbb{Z}/q\mathbb{Z}$ uniformly at random.
2. Form the pair $(g^x, h^x g^{xy})$.
3. Run $\mathcal{A}(g^x, h^x g^{xy}) = a$.
4. Output $a - y \pmod q$.

If group operations can be performed in polynomial time, which we assume, this algorithm is clearly polynomial time if $\mathcal{A}$ is. If a solves the discrete logarithm problem for the pair $(g^x, h^x g^{xy})$ so that $(g^x)^a = h^x g^{xy}$, then by multiplying by g^{-xy} on both sides we find

$$\begin{aligned}
(g^x)^{a-y} &= h^x \\
\iff g^{x(a-y)} &= h^x \\
\iff \left(g^{(a-y)}\right)^x &= h^x \\
\iff g^{(a-y)} &= h
\end{aligned}$$

To find the discrete logarithm, we take $a - y \pmod q$. KOLLA DETTA EN EXTRA GnNG Therefore this algorithm solves the discrete logarithm problem for the pair (g, h).

Task 4.4 (1T). Since $G_q \cong \mathbb{Z}/q\mathbb{Z}$ through the group isomorphism $\phi : G_q \to \mathbb{Z}/q\mathbb{Z}; a \mapsto g^a$ and x is chosen uniformly at random, the elements g^x and h^x are chosen uniformly at random from the set of generators of G_q. Since y is also chosen uniformly at random and multiplication by some fixed element in G_q is an isomorphism $G_q \to G_q$, the element $h^x g^{xy}$ is an element in G_q which is chosen uniformly at random, and in fact independently from g^x since x and y are chosen independently. Therefore the pair $(g^x, h^x g^{xy})$ is a pair chosen uniformly and independently at random. KOLLA IGEN

Suppose $\mathcal{A}$ violates the average case discrete logarithm assumption. Then for a randomly chosen generator g and a randomly chosen h in G_q, we have that $\Pr[(\mathcal{A}(g,h) = \log_g(h)]$ is non-negligible. We proved that the elements in the pair $(g^x, h^x g^{xy})$ are in fact chosen independently and uniformly at random, so $\Pr[(\mathcal{A}(g^x, h^x g^{xy}) = \log_{g^x}(h^x g^{xy})]$ is non-negligible. But then, since we can deterministically determine $\log_g(h)$ from $\log_{g^x}(h^x g^{xy})$ and do so in the algorithm $\mathcal{A}'$, we can conclude that $\Pr[\mathcal{A}'(g,h) = \log_g(h)]$ is in fact non-negligible. Therefore $\mathcal{A}'$ breaks the worst-case discrete logarithm assumption, as desired.

Problem 5.

Task 5.1 (5T). Let $r = \lceil \log q \rceil + t$. The strings $s \in \{0,1\}^{\lceil \log q \rceil + t}$ are in direct correspondence with the integers up to the number $N = \sum_{i=1}^{r} 2^{i-1}$ through a change of basis $s = x_1 \ldots, x_r \mapsto \sum_{i=1}^{N} x_i 2^{i-1}$. We can write $N = kq + a$ so that $N \pmod q = a$ and $\lfloor N/q \rfloor = k$. Using the thinking described in the hint, we can deduce that we have a bias when sampling numbers from $\{0,1\}^{\lceil \log q \rceil + t}$: For those numbers z in $\mathbb{Z}_q$ who fulfil $z \leq a$, a total of $k+1$ different samples in $\{0,1\}^{\lceil \log q \rceil + t}$ will correspond to z, namely $iq+z$ for $i = 0, 1, \ldots, k$ (each i being a 'sheet of paper' as in the hint). Therefore $P_Y(z) = (k+1)/(N+1)$. (We get a term +1 because 0 is also a possible number). Meanwhile, for $z > a$ we have k different samples corresponding to choosing z, namely $iq + z$ for $i = 0, 1, \ldots, k-1$ so $P_Y(z) = k/(N+1)$. Therefore we can write

$$\begin{aligned} \|P_Y - P_Z\| &= \frac{1}{2} \sum_{x \in \mathbb{Z}_q} |P_Y(x) - P_Z(x)| \\ &= \frac{1}{2} \sum_{x \leq a} \left| \frac{k+1}{N+1} - \frac{1}{q} \right| + \frac{1}{2} \sum_{x > a} \left| \frac{k}{N+1} - \frac{1}{q} \right|. \end{aligned}$$

We have

$$\begin{aligned} \left| \frac{k}{N+1} - \frac{1}{q} \right| &= \left| \frac{k}{kq+a+1} - \frac{1}{q} \right| \\ &= \left| \frac{kq - (kq+a+1)}{q(kq+a+1)} \right| \\ &= \left| \frac{-(a+1)}{q(kq+a+1)} \right| \end{aligned}$$

Similarly,

$$\begin{aligned} \left| \frac{k+1}{N+1} - \frac{1}{q} \right| &= \left| \frac{k+1}{kq+a+1} - \frac{1}{q} \right| \\ &= \left| \frac{(k+1)q - (kq+a+1)}{q(kq+a+1)} \right| \\ &= \left| \frac{q-(a+1)}{q(kq+a+1)} \right| \end{aligned}$$

so we can write

$$\begin{aligned}
\|P_Y - P_Z\| &= \frac{1}{2}\sum_{x\le a}\left|\frac{k+1}{N} - \frac{1}{q}\right| + \frac{1}{2}\sum_{x>a}\left|\frac{k}{N} - \frac{1}{q}\right| \\
&= \frac{1}{2}(a+1)\frac{q-(a+1)}{q(kq+a+1)} + \frac{1}{2}(q-(a+1))\frac{a+1}{q(kq+a+1)} \\
&= \frac{q(a+1)-(a+1)^2}{q(kq+a+1)}.
\end{aligned}$$

As a sanity check, note that if $a = q-1$ so that we sample from what is essentially $\mathbb{Z}_q$ (that is, nothing has been cut off the top sheet of paper), this expression, and hence the statistical distance, is 0 as we expect. Note that $kq + a + 1 = N + 1 = 2^{\lceil \log q \rceil + t}$. The expression in the numerator has derivative $q - 2a + 1$ which is zero when $a = (q-1)/2$. This is a local maximum since the second derivative is always negative, and fits intuition since $a = (q-1)/2$ corresponds to half of the top sheet being cut off, which should reasonably introduce maximal bias into the sampling. Using these results we find

$$\begin{aligned}
\|P_Y - P_Z\| &= \frac{q(a+1)-(a+1)^2}{q(kq+a+1)} \\
&= \frac{q(a+1)-(a+1)^2}{q2^{\lceil \log q \rceil + t}} \\
&\le \{a \leftarrow (q-1)/2\} \\
&\le \frac{q^2-1}{4q2^{\lceil \log q \rceil + t}} \\
&\le \frac{q^2-1}{4q^2 2^t} \\
&= \frac{1-1/q^2}{4\cdot 2^t}.
\end{aligned}$$

I take the assignment phrasing to mean that we should find a bound $\beta(t)$ which is independent from q. Since q is an odd prime and in particular $q \ge 3$, we can take $\beta(t) = \frac{1-1/3^2}{4\cdot 2^t} = \frac{2}{9\cdot 2^t}$. Note that in particular we get exponential decay in t.

Task 5.2 (4T). We think of the X_i as being drawn sequentially: X_1 can be any nonzero vector. Since there are q^k vectors in $\mathbb{Z}_q^k$, the probability of the sampling of X_1 being successful is $(q^k-1)/q^k$, with the 1 appearing to account for the zero vector. Moreover, the probability of a randomly chosen vector w in $\mathbb{Z}_q^k$ being linearly independent from a given set of i vectors $v_1, \ldots, v_i$ is the probability of w not being in $\mathsf{Span}(v_1, \ldots, v_i)$. Since $v_1, \ldots, v_i$ are assumed to be linearly independent, there are q^i vectors in $\mathsf{Span}(v_1, \ldots, v_i)$. The probability of w not being in $\mathsf{Span}(v_1, \ldots, v_i)$ is therefore $(q^k - q^i)/q^k$. Because the sampling of X_i for each i are all independent events, we can write

$$\begin{aligned}
\Pr[X_1, \ldots X_k \text{ lin. ind.}] &= \prod_{i=1}^{k} \Pr[X_i \notin \mathsf{Span}(X_1, \ldots, X_{i-1})] \\
&= \prod_{i=1}^{k} \frac{q^k - q^{i-1}}{q^k} \\
&= \prod_{i=1}^{k} \left(1 - q^{i-1-k}\right) \\
&\ge (1-1/q)^k.
\end{aligned}$$

Therefore we get

$$\Pr[\mathsf{Span}(X_1, \ldots, X_k) \ne \mathbb{Z}_q^k] = 1 - \Pr[X_1, \ldots X_k \text{ lin. ind.}] \le 1 - (1-1/q)^k,$$

so we get $l(q,k) = 1-(1-1/q)^k$. This expression is close to 0 for $q >> k$, as we hope for.

Task 5.3 (4T). We use a similar reasoning as in the previous assignment: The first sampled vector can be any vector $F(S)$, even for $S = 0$. If we have a set of i vectors of the form $F(S_k), k = 1,\dots,i$, the probability of sampling a vector w of the form $F(S_w)$ which is not in $\mathsf{Span}(v_1,\dots,v_i)$ amounts to S_w not being equal to S_k for any k, since v_k and v_l are linearly independent if and only if $S_k \neq S_l$ (or if S_k or S_l are zero). This follows from noting that the matrix

$$V = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ F(S_1) & F(S_2) & \dots & F(S_k) \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ S_1 & S_2 & \dots & S_k \\ \vdots & \vdots & \vdots & \vdots \\ S_1^{k-1} & S_2^{k-1} & \dots & S_k^{k-1} \end{bmatrix}$$

is (the transpose of) a *Vandermonde matrix* CITERA whose determinant is well known to be $\det(V) = \prod_{1\leq i<j\leq k}(S_j - S_i)$, which is nonzero (so its columns are linearly independent) in $\mathbb{Z}_q$ if and only if all $S_i, i = 1,\dots,k$ are distinct. It follows that $\Pr[X_i \notin \mathsf{Span}(X_1,\dots,X_{i-1})] = (q-i-1)/q$, corresponding to S_i not being equal to $S_n, n = 1,\dots,i-1$, with the S_n all being distinct. Therefore, we can write

$$\begin{aligned} \Pr[X_1,\dots,X_k \text{ lin. ind.}] &= \prod_{i=1}^{k} \Pr[X_i \notin \mathsf{Span} X_1,\dots,X_{i-1}] \\ &= \prod_{i=1}^{k} (q-(i-1))/q \\ &\geq (1-(k-1)/q)^k \end{aligned}$$

so

$$\Pr[\mathsf{Span}(X_1,\dots,X_k) \neq \mathbb{Z}_q^k] = 1 - \Pr[X_1,\dots X_k \text{ lin. ind.}] \leq 1-(1-(k-1)/q)^k.$$

Again, this is very close to 0 for $q >> k$, as hoped for and expected. As a sanity check, this expression is thankfully 0 for $k = 1$.

Problem 6.

Task 6.1 (2T). NOT SOLVED

Task 6.2 (2T). NOT SOLVED

Task 6.3 (2T). NOT SOLVED

Task 6.4 (2T). NOT SOLVED

References

[1] Steven D. Galbraith. *Mathematics of Public Key Cryptography. Version 2.0.* 2018.

[2] Douglas R. Stinson. *Cryptography. Theory and Practice (3ed).* Chapman & Hall/CRC, third edition, 2006.