AARHUS UNIVERSITY - COMPUTER SCIENCE

Protocol Theory - Handin 5

Peter Burgaard -201209175

March 8, 2017

EXERCISE 5

Assume P commits to two strings $b_1,\dots,b_t,b'_1,\dots,b'_t$ using commitments $c_1,\dots,c_t,c'_1,\dots,c'_t$ as in exercise 4. He claims that the strings are different, and want to convince V that this is the case while revealing no extra information. Note that he cannot point to an index j where $b_j \neq b'_j$ and use the above method on c_j, c'_j. This would reveal *where* the strings are different. Instead consider the following protocol:

1. P chooses a random permutation π on the set of indices $1,\dots,t$. He computes, for $i = 1,\dots t$ a commitment $d_i = commit_{pk}(s'_i, b'_{\pi(i)})$. In other words, permute both strings randomly and commit bit by bit to the resulting strings. Send $d_1,\dots,d_t,d'_1,\dots,d'_t$ to V.
2. V chooses a random bit b, sends it to P
3. If $b = 0$, P reveals π and uses the above method to convince V for all i that $c_{\pi(i)}$ contains the same bit as d_i. Similarly for $c'_{\pi(i)}$ and d'_i. If $b = 1$, P finds a position i, where $b'_{\pi(i)} \neq b'_{\pi(i)}$ and uses the above method to convince V that d_i, d'_i contain different bits.

- Completeness - Ague that an honest prover can always convinces the verifier.

  This is trivial. We can go through each case a see it holds. Since i follows the protocol then we know how the scheme can be used.

- Soundness - Assume the prover cannot solve discrete log, but he knows how to open the commitments $c_1,\dots,c_t,c'_1,\dots,c'_t$. Show that if P can, for some set of commitments $d_1,\dots,d_t,d'_1,\dots,d'_t$ answer V correctly both for $b = 0$ and $b = 1$, then there is at least one j, where P can open c_j, c'_j to reveal different bits. Note this implies that a cheating prover could make V accept t iterations of the protocol with probability at most 2^{-t}.

Assume P can convince V for both $b = 1$ and $b = 0$. We look at what we get from each case for both $b = 0$ and $b = 1$. If $b = 0$ then for some i where d_i, and d_i' commits to different bits, then c_i and c_i' will commit to different bits. Else for $b = 1$, then d_i and c_i commits to same bit and b_i and b_i' commit to different bits. The strings are different in both cases, so it's a sound protocol.

- Zero-Knowledge: Sketch a simulator for this protocol. Hint: given commitment c, if you set $d = cg^s \mod (p)$, then $cd^{-1} = g^s \mod (p)$. This means that even if the simulator does not know how to open c, it can create d and fake a proof that d contains the same bit as c. You do not have to formally prove that you simulator works.

  We construct a simulator to show zero-knowledge. We pick b randomly, and we do something different depending on whether it is 0 or 1. If $b = 0$, then we pick x_i abitrarely, and let $d_i = c_{\pi(i)} g^{-x_i}$, then we send it over to V, and we get a bit back. If the bit matches our b, then we are good, else we rewind. If it matches, then we need to provide s, which is x_i. In the other caes $b = 1$, we pick r_i and let $d_i = y \cdot g^{r_1}$ and $d_i' = g^{r_2}$, and send all of them over. If we get a bit 0 back, we rewind, else if the bit is 1 then we show they commit to different bit, and send over $r1 + r2$.