Sarthak Singhal
(20171091)

PQR-1

4th April 2020

Questions

A. Design a zero-knowledge proof for the Discrete-Logarithm Problem (DLP), that is, given prime p, generator g and the element y $= g^x \, mod \, p$, how does a prover claiming to know x, convince the verifier, without revealing x?
B. Moreover, using hash-functions (and assuming them to be random oracles) show how to build a digital signature scheme based on your above zero-knowledge proof and the hardness of DLP?
C. Also, show how you would design collision-resistant hash functions based on the hardness of DLP.

Answers

ANS-1

Definition of ZKP

In cryptography, a zero-knowledge proof or zero-knowledge protocol is a method by which one party (the prover) can prove to another party (the verifier) that they know a value x, without conveying any information apart from the fact that they know the value x. The essence of zero-knowledge proofs is that it is trivial to prove that one possesses knowledge of certain information by simply revealing it; the challenge is to prove such possession without revealing the information itself or any additional information.(WIKI)

Construction

Let P be prover and V be verifier.

Let $y = g^x mod p$ be the DLP.

Let P wants to show to V that it knows x, without revealing x or any other information about x.

Given information:

- g,p and y are public
- x is the private key of P (not known to anyone else)

Step1 (P's computation)

- Chose any random number r from the group Zp*
- Construct $t = g^r mod p$
- Send this t to V

Step2 (V's computation)

- Chose any random number c from the group Zp*
- Send this to P

Step3 (P's computation)

- Construct $z = c * x + r$
- Send this z to V

Step4 (V's computation)

- Check if $g^z mod p = (y^c * t) mod p$
- If true, then V verifies that P knows x else not

Verification Proof

Given

- $z = c * x + r$
- $y = g^x mod p$
- $t = g^r mod p$

$RHS = (y^c * t) mod p$

$= ((g^{c*x} mod p) * t) mod p$

$= ((g^{c*x} mod p) * (g^r mod p)) mod p$

$= g^{c*x+r} mod p$

$= g^z mod p$

$= LHS$

Thus V will verify if $g^z mod p = (y^c * t) mod p$.

Completeness Proof

From above calculations, it is clear that RHS = LHS only when

$c * x^o + r = c * x^p + r$, where x^o is original x and x^p is x with prover.

If P knows x then $x^o = x^p$ and thus V will verify and hence the proof is complete.

Soundness Proof

If P doesn't know x then probability of $x^o = x^p$ is negligible (1/P) and thus probability of acceptance by V is also negligible. Thus the proof is sound.

Zero Knowledge

P sends z and t to V. Since t is DLP, no information about x can be revealed by revealing t. $z = c * x + r$ where c and r are random numbers so z is also random and no information about x is revealed. Thus P doesn't reveal any information about x by sharing z and t and thus the proof is ZKP.

ANS-2

- The above method cannot be used as digital signature as it is interactive, meaning information flows from both P and V to each other.
- If we can construct a non-interactive ZKP, we can use it as a digital signature.
- Only information that flows from V to P is of c which is a random number.
- Thus if P can send c to V the proof would become non-interactive and can be used as a digital signature.
- P would also need to prove to V that c is truly a random number.
- To ensure this, we can take $c = H(p, g, y)$ and send it to V.
- As $H(x, y, z)$ is a random oracle, c is truly a random number and thus our problem of proving c random is solved.

Using this, above algorithm can be modified as:

Step1 (P's computation) **[Signing]**

- Chose any random number r from the group Zp*
- Construct $t = g^r modp$
- Take $c = H(p, g, y)$ [Step 2 removed]
- Construct $z = c * x + r$ [Step 3 removed]
- Send t & z to V

Step2 (V's computation) **[Verifying]**

- Take $c = H(p, g, y)$
- Check if $g^z modp = (y^c * t)modp$
- If true, then V verifies that P knows x else not

Note that steps 2 & 3 are omitted.

Thus based on the above ZKP and hardness of DLP, we have constructed a digital signature using H as a random oracle.

ANS-3

- Now to construct a random oracle H() we can use H as a CRH as it satisfies properties of a random oracle.
- To design a CRH, we can use the Merkle-Damgard Transform which is a way of extending a fixed-length CRH function into a general one that receives inputs of any length.

Construction

Let <Genh,h> be a fixed-length CRH with input length 2L and output length L. Construct a variable-length CRH <Gen,H> as follows:

$Gen(1^n)$:

Upon input 1^n, run the key-generation algorithm Genh of the fixed-length CRH and output the key. Let it be s.

$H^s(M)$:

- Let the message be M of length x ($x<2^L$) and key be s.
- Pad M with zeroes so that its length is exactly a multiple of L.
- Now divide M into B blocks each of size L.

- Now $M=(m_1||m_2||...||m_B)$.
- Define $Z_0 = 0^L$ (initialization vector)
- For every i in (1,...,B), compute $Z_i = h^s(Z_{i-1} || m_i)$
- Output $Z=h^s(Z_B || x)$

Note: $x<2^L$ is just to ensure that x fits in one block. This is not a hard requirement.

Claim is that if h is CRH, constructed H is also a CRH.

Proof

- We will prove that if there is a collision in H, there will also be a collision in h.
- This would imply that h is not a CRH.
- This would be a contradiction as h is a CRH.
- Hence our assumption of getting a collision in H would be wrong which would prove H to be a CRH.

Let $M = (m_1 || m_2 || m_B)$ with length x and $M' = (m'_1 || m'_2 || m'_{B'})$ with length x'.

Let there be a collision in H.

Case-1 ($x \neq x'$):

- Since $H(x) = H(x') \Rightarrow h^s(Z_B || x) = h^s(Z'_{B'} || x')$.
- Since $x \neq x'$ => collision in h^s.
- Hence contradiction.

Case-2 ($x = x'$)

- Let Z and Z' be intermediate hash values of M and M' during the computation of H.
- Since $M \neq M'$ and they are of same length, $\exists$ at least one index i ($1 \leq i \leq B$) such that $m_i \neq m'_i$.

- Let i* be the highest index for which it holds that $Z_{i^*-1} || m_{i^*} \neq Z'_{i^*-1} || m'_{i^*}$
- If i* = B, then ($Z_{i^*-1} || m_{i^*}$) and ($Z'_{i^*-1} || m'_{i^*}$) constitute a collision because we know that H of both the messages is same along with length implying $Z_B = Z'_B$ (if $Z_B \neq Z'_B$ there is already a collision).
- If i*<B, then maximality of i* implies $Z_{i^*} = Z'_{i^*}$
  - Because $\forall i > i^*$, $Z_{i-1} || m_i = Z'_{i-1} || m'_i$
  - Let $i = i^* + 1$
  - => $Z_{i^*} || m_{i^*} = Z'_{i^*} || m'_{i^*}$
  - => $Z_{i^*} = Z'_{i^*}$ and $m_{i^*} = m'_{i^*}$
- This again implies collision in ($Z_{i^*-1} || m_{i^*}$) and ($Z'_{i^*-1} || m'_{i^*}$)
- Thus in all cases we get a collision in h^s which is a contradiction.

Hence constructed H^s is truly a CRH if we have h^s.

Constructing h^s

We can construct h^s based on the hardness DLP as follows:

$$h(x, y) = (g^x * z^y) mod p \text{ where } z = g^k mod p$$

Claim: If someone finds collision in h, then he can solve DLP (i.e. He can find k given z,g,p)

Proof

Let there be a collision in h.

$$h(x1, y1) = h(x2, y2)$$

Since $x1y1 \neq x2y2$, WLOG lets take $y1 \neq y2$

$$h(x1, y1) = (g^{x1} * z^{y1}) mod p$$

$$h(x2, y2) = (g^{x2} * z^{y2}) mod p$$

$$=> g^{x1} z^{y1} \equiv g^{x2} z^{y2}$$

=> $g^{x1-x2} \equiv z^{y2-y1}$

=> $g^{x1-x2} \equiv g^{k(y2-y1)}$

=> $(g^{x1-x2})^{(y2-y1)^{-1}} \equiv (g^{k(y2-y1)})^{(y2-y1)^{-1}} \equiv g^k$

=> $k = (x1 - x2)/(y2 - y1)$

Thus if we find collision in our hash function, we can compute k efficiently and thus can solve DLP, which is not possible.

Hence h is a CRH.