Principals of Information Security		Date:	*1 April 2020*
Instructor: *Prof Kannan Srinathan*		Wrriten by:	Gaurav Batra (20171114)

POIS Evaluation Question-1

1 Problems

1. Design a zero-knowledge proof for the Discrete-Logarithm Problem (DLP), that is, given prime p, generator g and the element $y = g^x modp$, how does a prover claiming to know x, convince the verifier, without revealing x?

1.1 Zero Knowledge proof

Zero-knowledge proof is a method by which the Prover (P) can prove to the Verifier (V) that it knows a value x, without conveying any information apart from the fact that it knows the value x.
A true zero-knowledge proof needs to prove 3 criteria:

- **Completeness:** it should convince the Verifier that the Prover knows what they say they know.
- **Soundness:** if the information is false, it cannot convince the Verifier that the Prover's information is true.
- **Zero-knowledge-ness:** it should reveal nothing else to the Verifier.

1.2 Construction of ZKP from DLP

1.2.1 Assumptions

- Let P be the Prover and V be the Verifier.
- x is the private key known only to P.
- Let p, g and $y = g^x \bmod p$ be known to both P and V. (p is a prime and g is generator of the cyclic group $\mathbb{Z}_p^*$).
- Let length of prime p is n.
- DLP is a one way function.

1.2.2 Goal

The Prover P wants to convince the Verifier V that is knows x such that $y = g^x \bmod p$.

1.2.3 Construction

(a) P sends $t = g^r \bmod p$ to V, where r is any random number from the group $\mathbb{Z}_p^*$.
(b) V sends c to P, where c is any random number from the group $\mathbb{Z}_p^*$.
(c) P sends $z = cx + r$ to V.
(d) V calculates $k = g^z \bmod p$. If $k = (y^c t) \bmod p$, then accept, else reject.

The above construction is a ZKP as it satifies the following criteria:

1.2.4 Completeness

If P knows x, it can always send the correct $z = cx+r$ to V. Therefore proof accepts, whenever P knows x as:

$$\begin{aligned} y^c t \bmod p &= g^{cx} t \bmod p \\ &= g^{cx} g^r \bmod p \\ &= g^{cx+r} \bmod p \\ &= g^z \bmod p \end{aligned} \tag{1}$$

1.2.5 Soundness

If P does not know x, then the proof accepts only if he is able to output $z^{'} = cx^{'} + r$ such that $z^{'} = z$, that is $cx^{'} + r = cx + r$.
The probability of this happening is $O(\frac{1}{p})$, which is negligible if n is large. Thus, the proof is sound.

1.2.6 Zero-Knowledge

P sends z, t to V. Since $t = g^r \bmod p$, 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 Zero-Knowledge.

2. Moreover, using hash-functions (and assuming them to be random oracles) show how would to build a digital signature scheme based on your above zero-knowledge proof and the hardness of DLP?

   The construction in the last question can be made into a Digital Signature scheme if we can make it non-interactive, that is make it so that all the information is sent from P to V.
   The only information that V sends to P is c, which is a uniformly chosen element from $\mathbb{Z}_p^*$. We cannot let P choose c, as then, P would be able to fool V. This is accomplished by using a collision resistant hash function $\mathbb{H}(p, g, y)$ as c.
   Our construction is modified as follows:

   #### 1.2.7 Construction

   (a) P sends $t = g^r \bmod p$ to V, where r is any random number from the group $\mathbb{Z}_p^*$.
   (b) P sends $z = cx + r$ to V, where $c = \mathbb{H}(p, g, y)$.
   (c) V accepts if $g^z \bmod p = y^{\mathbb{H}(p,g,y)} t \bmod p$ else reject.

   Thus, based on the Zero-Knowledge Proof and hardness of DLP, we have constructed a digital signature using $\mathbb{H}$ as a random oracle.
   All the above proofs of Completeness, Soundness and Zero-Knowledge still hold here.

3. Also, show how would you design collision-resistant hash functions based on the hardness of DLP?

   Now to construct the random oracle $\mathbb{H}(p, g, y)$, we can a use a collision-resistant hash function. We can create a collision-resistant hash function which is based on the hardness of DLP as follows:

1.2.8 Construction

(a) Design a hash function $\mathbb{H}^r : \{0,1\}^{2n} \rightarrow \{0,1\}^n$ as:

$$\mathbb{H}^r(x, y) = (g^x z^y) \bmod p \; where \; z = g^k \bmod p \text{ where } k \text{ is a random number}$$

This function is collision resistant as if a PPTM adversary can find a collision in the above hash function implies that it can solve DLP which is shown below:

Proof: Let there be a collision in $\mathbb{H}^r$, that is $\mathbb{H}^r(x_1, y_1) = \mathbb{H}^r(x_2, y_2)$. Since $x_1 y_1 \neq x_2 y_2$, WLOG we can take $y_1 \neq y_2$.

$$\begin{aligned}
\mathbb{H}^r(x_1, y_1) &= (g^{x_1} z^{y_1}) \bmod p \\
\mathbb{H}^r(x_2, y_2) &= (g^{x_2} z^{y_2}) \bmod p \\
\therefore (g^{x_1} z^{y_1}) \bmod p &= (g^{x_2} z^{y_2}) \bmod p \\
(g^{x_1 - x_2}) \bmod p &= (z^{y_2 - y_1}) \bmod p \\
(g^{x_1 - x_2}) \bmod p &= (g^{k(y_2 - y_1)}) \bmod p \\
\therefore k &= \frac{x_1 - x_2}{y_2 - y_1}
\end{aligned} \tag{2}$$

Hence we have found k such that $z = g^k \bmod p$ and thus have solved DLP.

(b) To get arbitrary length hash function, we can use the Merkle-Damgard Transform which is a way of extending a fixed length collision-resistant function into a general one that receives input of any length.

Given $\mathbb{H}^r : \{0,1\}^{2n} \rightarrow \{0,1\}^n$, we need to build $\mathbb{H} : \{0,1\}^* \rightarrow \{0,1\}^n$ as follows:

i. Let the message be $m \in \{0,1\}^*$. We pad m such that $|m|$ is a multiple of n.
ii. Divide m into b blocks each of size n, that is $m = m_1, m_2, \ldots, m_b$.
iii. Define $z_0 = 0^n$.
iv. For every i in $(1, 2, \ldots, b)$, compute $z_i = \mathbb{H}^r(z_{i-1} || m_i)$.
v. Ouptut $z = \mathbb{H}^r(z_b || |m|)$.

If $\mathbb{H}^r$ is collision resistant hash function, then so is $\mathbb{H}$, which is proved as follows:

Proof: Let $m = m_1, m_2, \ldots, m_b$ with $|m| = t$ and $m^{'} = m^{'}_1, m^{'}_2, \ldots, m^{'}_{b'}$ with $|m^{'}| = t^{'}$. Let there be a collision, that is $\mathbb{H}(t) = \mathbb{H}(t^{'})$.

i. If $t \neq t^{'}$, as $\mathbb{H}(t) = \mathbb{H}(t^{'}) \implies \mathbb{H}^r(z_b || t) = \mathbb{H}^r(z^{'}_{b'} || t^{'})$. As $t \neq t^{'}$, this implies a collision in $\mathbb{H}^r$. Hence contradiction.
ii. if $t = t^{'}$,
   A. Let z and $z^{'}$ be the intermediate hash values of m and $m^{'}$ during the computation of $\mathbb{H}$.
   B. Since $m \neq m^{'}$ and they are of same length, $\exists$ at least one index i such that $m_i \neq m^{'}_i$.
   C. Let i^* be the highest index for which it holds that $z_{i^*-1} || m^*_i \neq z^{'}_{i^*-1} || m^{'}_{i^*}$.
   D. If $i^* = b$, then $(z_{i^*-1} || m^*_i)$ and $(z^{'}_{i^*-1} || m^{'}_{i^*})$, constitute a collision else $m = m^{'}$.
   E. If $i^* < b$, then maximality of i^* implies $z_{i^*} = z^{'}_{i^*}$.
   F. This again implies collision in $\mathbb{H}^r$. Hence, we reach a contradiction.