HomeWork - 5

Vishnuta Kudrethi

1) Commitment Schemes [4 points]

a) We could use symmetric encryption between the user and the server to hide bids that are made by the user. We could also use public key encryption where bids made by the users can be encrypted before sending them to the server.

However, the scheme fails if the auction organizer is malicious or 'dishonest'.

This can be handled by using signatures on the dispute so made by the user can be resolved with the organizer in the commitment scheme.

b) Users commit their bids and place them on the public message board. When the auction ends, the bids are publicly revealed and traced back to the commitments. Thereby ensuring the users of the bids that they made.

c) Given: $c = g^x h^r$ and $h = g^a$

Let, $g^x h^r = g^{x'} h^{r'}$

$$g^x (g^a)^r = g^{x'} (g^a)^{r'} \Rightarrow g^{x+ar} = g^{x'+ar'}$$

$$\Rightarrow x + ar = x' + ar'$$

Solving for $r' = (x - x')a^{-1} + r \Rightarrow$ which is a uniform distribution over all x.

Therefore, the scheme is perfectly hiding.

d) Let us assume that the adversary can find (x, r), (x', r'), where $x \neq x'$, and $g^x h^r = g^{x'} h^{r'}$.
If this is the case; given that $h = g^a$, as before a can be computed as $(x - x')(r' - r)^{-1}$.
This is contradictory to discrete log assumption.
Therefore, the scheme is computationally binding.

2) TOR Guard Nodes and Attacks [6 points]

a) If A can compromise a node near the B's circuit, say for example it can control the first node of the B's circuit. A could send crafted messages to B and by eavesdropping at the first node of B's circuit; A can identify its own messages and hence determine the IP address of B as the crafted messages were directed to B.

b) Yes, it would change the success of the attack described above. The probability of the attack becomes really low if B's first node is replaced with a long-term guard node.

c) We can do a DDoS attack on the entry node of B and make the entry node non-functional; thereby forcing B to choose from a random set of nodes. So, if A has enough number of nodes such that the probability of B choosing A's malicious node at random is high, then, A will be able to determine B's IP address efficiently.

If we want the attack to work in 1 week (7 days), and assuming an attack to succeed takes about a day. Then, B can choose maximum of 7 nodes with a probability of $1/2$ being the number of nodes controlled by A.

$\therefore, \left(\frac{1}{x}\right)^7 = 1/2$; $x \approx 0.95$ (approx); x being the number of nodes controlled by A.

→ Hence, in 10,000 nodes, A controls $\frac{9.5}{100} \times 10000 = 950$ nodes.

→ Cost to host a node = \$100

Total cost borne by A to host 95% nodes = $\frac{950 \times 100}{10}$

$= \frac{950,000}{10} = \$95,000$

→ Cost of DDoS attack = $7 \times \$1000 = \$7000$

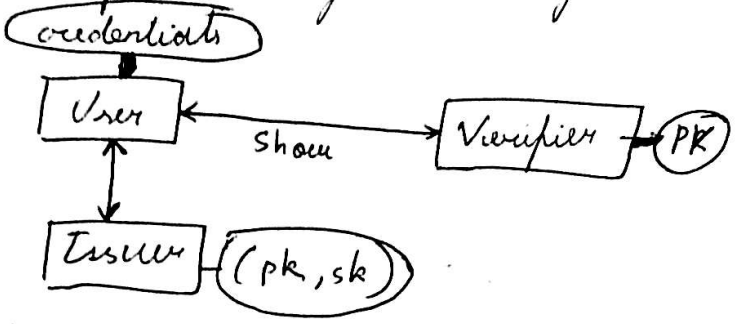
Therefore, total cost = $7000 + 95,000 + 100$

$\approx$ \$102,100

It is to be noted that If the number of days to perform the attack increases, the cost borne by A to perform the attack also increases quite linearly.

3) Anonymous Credential Systems.

a) Anonymous credential system is used to enable strong authentication and privacy at the same time. It ~~also~~ ~~provide authen~~ uses credentials as a mean to authenticate an entity. It involves a user, an issuer and a verifier. The general layout is as follows:



The issuer is a trusted entity that issues certificate. The user engages in an issue protocol with an issuer to obtain a valid credential on a certain set of user attributes. The credential is valid under the issuer's public key pk, of which only the issuer knows the corresponding secret key sk. The user then convinces the verifier that she has a certain set of attributes by engaging in a show protocol with the verifier.

Limitations:

1) The user has to reveal all the attributes so that the verifier can check the signature.
2) The verifier can impersonate by reusing the credentials with respect to other verifiers.

b) The user sends to the verifier: $y = r + cx \bmod q$, where x is the secret known to the user, r is a random chosen by the user and c is a malicious entity chosen by the verifier.

Known that $r \xleftarrow{R} \mathbb{Z}_q$, then $y \xleftarrow{R} \mathbb{Z}_q$; which state that no matter what the value of c is, it (V) will not be able to generate x.

c) $c = H(A, B)$ where H is a collision resistant hash function. We can see that A cannot select a 'c' value such that $y = r + cx$ passed the verifier test because that would mean H is not collision resistant, or in other words A was able to find a collision. It is to be noted that $A = g^r$ and $B = g^{-r}$. are publishers.

d) When a user establishes a nym with an organisation, they go through the nym generation process: $N = (a, b)$ and the user calculates $(\bar{a}, \bar{b}) = (g^r, g^{rx})$. The user throughout the system will be asked to prove x which will be done with the protocol Π.

This relationship is important because for the security of the system because if a user is impersonating credentials, he would need to transfer the secret x as well, which is as good as revealing / stealing ones identity. Therefore, this system forces users to not share their credentials and hence preserving the security of the system.