دانشکده‌ی علوم ریاضی

مقدمه‌ای بر رمزنگاری

پاسخ نامه سری ۷

نگارنده : آرش عاشوری

Problem 1

Scheme 1:
This scheme is secure. Suppose to the contrary that there exists an adversary A which has a non negligible advantage $\mu(n)$ in attacking this scheme. Now consider an adversary B attacking to the original scheme as the following:
It runs another signature scheme C as the original scheme. we name its keys by (sk_1, pk_1).
Then it chooses a random bit b by probability $\frac{1}{2}$ and sets $(PK_0, PK_1) = (pk_b, pk_{\bar{b}})$, which pk_0 is the challengers public key. Attaker A attacks to a signature scheme 1 with public key (PK_0, PK_1). Now for every message m which A sends, B sends it to the challenger and gets σ_0 and also it computes $\sigma_1 = S(sk_1, m)$ and sets $\sigma = (\sigma_b, \sigma_{\bar{b}})$ and Then sends σ to A.
Finally when A sends (m, σ_0, σ_1), B sends (m, σ_b) to the challenger.
By symmetry we have:

$\Pr(V(PK_0, m, \sigma_0) = 1) = \Pr(V(PK_1, m, \sigma_1) = 1)$
$\mu(n) = \Pr(V(PK_0, m, \sigma_0) = 1 \vee V(PK_1, m, \sigma_1) = 1) \leq 2\Pr(V(pk_0, m, \sigma_b) = 1)$
$\Pr(V(pk_0, m, \sigma_b) = 1) \geq \frac{\mu(n)}{2}$
Hence B has a non neglible advantage against the original scheme which is a contradiction. Proof is complete.

Scheme 2: This scheme is not secure. Suppose an adversary sends two messages $(0^n||0^n), (1^n||1^n)$ to the challenger and recieves $c_0||c_1$ as the signature of the message $(0^n||0^n)$ and $c_2||c_3$ as the signature of the message $(1^n||1^n)$.
Then adversary sends message $(0^n||1^n)$ and $(c_0||c_3)$ as its signature. This signature would be verified with pribability 1. Because:

$c_0 = S(sk_0, 0^n), c_3 = S(sk_1, 1^n)$
$\rightarrow V(pk_0, 0^n, c_0) = 1, V(pk_1, 1^n, c_3) = 1 \rightarrow V_2((pk_0, pk_1), (0^n||1^n), c_0||c_3) = 1$

Problem 2

Suppose $f : \{0,1\}^n \rightarrow \{0,1\}^n$ be a one way function. Let A be the following scheme:
Gen: A choose $2k$ values $x_1^0, x_2^0, ..., x_k^0, x_1^1, x_2^1, ..., x_k^1$ each uniformly random chosen from $\{0,1\}^n$ And computes $y_j^b = f(x_j^b)$ for every $b \in \{0,1\}$ and $j \in \{1,2,...,k\}$.
A chooses $(y_1^0, ..., y_k^0, y_1^1, ..., y_k^1$ as the public key and $(x_1^0, x_2^0, ..., x_k^0, x_1^1, x_2^1, ..., x_k^1)$ as the secret key.
Signature: A gets the message $m = m_1 m_2 ... m_k \in \{0,1\}^k$ and computes
$Signature(sk, m) = x_1^{m_1} x_2^{m_2} ... x_k^{m_k}$.
Verify: A gets $(m, \sigma = z_1 z_2 ... z_k)$ and it outputs 1 if for every $j \in \{1,2,...,k\} : f(z_j) = y_j^{m_j}$ and it outputs 0 otherwise.

Now we prove that this scheme is one time secure. Suppose to the contrary that there exist a attacker B with non neglible advantage $\mu_(n)$. Consider the following attacker C to the one way function f. C chooses $2k-1$ values $z_1, ... z_{2k-1}$ each uniformly random from $\{0,1\}^n$ and computes $f(z_j)$ for every $j \in \{1,2,...,2k-1\}$. Then challenger chooses a random $z \in \{0,1\}^n$ and sends $f(z)$ to C. C sets a public key $y_1^0, ..., y_k^0, y_1^1, ..., y_k^1$ by a random permutation of $f(z)$ and $f(z_j)$ for $j \in \{1,2,...,2k-1\}$ and its correspondence secret key $x_1^0, x_2^0, ..., x_k^0, x_1^1, x_2^1, ..., x_k^1$. C doesnt know one element of this secret key (i.e., z).
C gives the public key to B and it sends at most one message m to the C to be signed. By probability $\frac{1}{2}$, C cant sign the message because the random permutation of public key. In this case C sends 0^n for its guess of z. In other case it signs m and then B send another message m_1 and its guessed signature σ_1. Because $m \neq m_1$ they are different in at least one bit. Hence by at least probability $\frac{1}{k}$ there are different in the bit which C doesnt know whats the secret key. Hence in this case if the signature σ_1 is right then C has found z. Hence C gets z by probability greater than $\frac{\mu(n)}{2k}$ which is not neglible. But this contadicts to f being a one way function. Proof is complete.

Problem 3

a) Let $n(.,.)$ be a polynomial. A n-hinting PRG scheme consists of two PPT algorithms $Setup, Eval$ with the following syntax.

$Setup(\lambda, l)$: The setup algorithm takes as input the security parameter $\lambda \in \mathbb{N}$, and length parameter $l \in \mathbb{N}$, and outputs public parameters pp and input length $n = n(\lambda, l)$.

$Eval(pp, s \in \{0,1\}^n)$: The evaluation algorithm takes as input the public parameters pp, an n bit string s, and outputs $z_0 z_1 ... z_n$, which each z_i is l bits.

Now for any PPT adverasry A and $\lambda, l \in N$ consider the folowing experiment:

1) Challenger runs $Setup(\lambda, l)$ and gives pp and n to A.
2) Challenger chooses a random bit b.
3) If $b = 0$, then challenger chooses a matrix z $(2 \times n)$ with each index chosen uniformly random from U_l, and a z_0 chosen uniformly random from U_l, otherwise (i.e., $b = 1$) it chooses a uniformly random string s from U_n and computes $x_0 x_1 ... x_n = Eval(pp, s)$ and also for every $i \in \{1, 2, 3, ..., n\}$ chooses y_i uniformly random from U_l. Then it computes $z_0 = x_0$, and for every $i \in \{1, 2, 3, ..., n\}, b \in \{0, 1\}$ if $b = x_{0i}$, $z(i, b) = x_i$ and otherwise $z(i, b) = y_i$
4) Challenger sends z and z_0 to A.
5) A chooses a bit $\bar{b}$.

A hinting PRG scheme $(Setup, Eval)$ is said to be secure if for any PPT adverasry A, polynomial $L(.)$, there exists a negligible function $negl()$ such that for all $\lambda \in \mathbb{N}, l = l(\lambda)$, for the above experiment the following hold:
$|\Pr(b = \bar{b}) - \frac{1}{2}| \leq negl(\lambda)$

b) Let $G : \{0,1\}^n \rightarrow \{0,1\}^{l(n)}$ be a PRG, then define $\bar{G} : \{0,1\}^{n+1} \rightarrow \{0,1\}^{l(n)+1}$ as:
$\bar{G}(s_1 s_2 ... s_{n+1}) = G(s_1 ... s_n) s_{n+1}$
We show that $\bar{G}$ is PRG but not a hinting PRG.
Suppose that $\bar{G}$ is not a PRG. Then there exist an attacker A with non neglible advantage $\mu(n)$. Using A we construct an attacker B to G. Suppose challenger sends $x_1 x_2 ... x_{l(n)}$ to B in the experiment, then B chooses a random bit x and sends $x_1 x_2 ... x_{l(n)} x$ to A. If A chooses $\bar{G}$, B chooses G and if A chooses $U_{l(n)+1}$, B chooses $U_{l(n)}$. Hence if A chooses right, B chooses right too, and conversely if A choose wrong it chooses wrong too. Hence their advantage is the same, which is a contradiction because G is a PRG. Hence $\bar{G}$ is PRG. Now we show that its not a hinting PRG. Suppose in the experiment of hinting PRG challenger sends z_0 and z_i^b for every $b \in \{0, 1\}$ and $i \in \{1, 2, ..., n+1\}$.

If the last bit of z^0_{n+1} be 0 or the last bit of z^1_{n+1} be 1, B chooses $\bar{G}$ in the experiment and chooses uniform distribution otherwise.

If challenger sends the data using $\bar{G}$, then B chooses $\bar{G}$. Because if the last input of PRG be s_{n+1} then the last bit of $z^{s_{n+1}}_{n+1}$ would be s_{n+1}.
And if challenger sends the data using unifoem distribution, B chooses $\bar{G}$ with probability $\frac{3}{4}$. Hence the advantage of B is $\frac{1}{4}$ which is not neglible. Hence $\bar{G}$ is not hinting PRG.

c) Suppose we have a CPA-secure public key encryption $\Pi(n) = (Gen, Enc, Dec)$, a Hinting PRG $H : \{0,1\}^n \to \{0,1\}^{n(n+1)}$ and a PRG G. And suppose algorithm ENC with parametr n uses a random $x \in \{0,1\}^n$ for encrypyion and message space is $\{0,1\}^n$. Let $\Pi'(n) = (GEN', ENC', DEC')$ be a public key encryption on the message space $\{0,1\}^n$ as following:

$GEN'(1^n)$: It runs $GEN(1^n)$, $2n$ times and obtains $pk = \{pk_{b,i}\}_{b\in\{0,1\}, i\in\{1,2,\ldots,n\}}$, $sk = \{sk_{b,i}\}_{b\in\{0,1\}, i\in\{1,2,\ldots,n\}}$.
And also runs *Setup* algorithm for hinting PRG for $\lambda = l = n$.

$ENC'(pk, m)$:
It first chooses a uniformly random tag $t = t_1 t_2 \ldots t_n$ where every t_i is from $\{0,1\}^{l(n)}$, which $l(n)$ is the length of the ouput of PRG G. Then it chooses a uniformly random seed s form $\{0,1\}^n$ and computes $H(s) = z_0 z_1 \ldots z_n$ and then computes the main ciphertext $c = z_0 \oplus m$.
And for every $i \in \{1,2,\ldots,n\}$, the signal ciphertexts c_{1i}, c_{2i}, c_{3i} are computed as follows:
It chooses x_i, h uniformly random from $\{0,1\}^n$ and:
If the ith bit s be zero then:

$$
\begin{aligned}
c_{0i} &= Enc(pk_{0i}, z_i, x_i) \\
c_{1i} &= Enc(pk_{1i}, h, 0^n) \\
c_{2i} &= G(x_i)
\end{aligned}
$$

And if the ith bit s be 1 then:

$$
\begin{aligned}
c_{0i} &= Enc(pk_{0i}, h, 0^n) \\
c_{1i} &= Enc(pk_{1i}, z_i, x_i) \\
c_{2i} &= G(x_i) + t_i
\end{aligned}
$$

Then:

$$Enc(pk, m) = \{c, t, \{c_{0i}, c_{1i}, c_{2i}\}_{i \in \{1,2,...,n\}}\}$$

$DEC'(pk, m)$: It first uses $\{sk_{0i}\}$ and obtains $y_i = DEC(sk_{i0}, c_{0i})$. It then checks if $G(y_i) = c_2$. If so, it guesses that $s_i = 0$, else it guesses that $s_i = 1$. With this estimate for s, the decryption algorithm can compute $H(s) = z_0 z_1 ... z_n$ and then compute $c \oplus z_0$ to learn the message m.

Then the decryption algorithm needs to check that the guess for s is indeed correct. If the ith bit of s is guessed to be 0, then the decryption algorithm checks that c_{0i} is a valid ciphertext - it simply checks if $ENC(pk_{0i}, y_i, z_i) = c_{0i}$. If the ith bit of s is guessed to be 1, then the decryption algorithm first recovers the messaage $\bar{y}_i = DEC(sk_{1i}, c_{1i})$. and checks if $c_{1i} = ENC(pk_{1i}, \bar{y}_i, z_i)$. , and also checks that $c_{2i} = G(\bar{y}_i) + t_i$. Finally, if all these checks pass, the decryption algorithm outputs $z_0 \oplus c$.