Ex4 - Foundation of Cryptography

Submitting: Michael Kellner

Through this solution $\langle , \rangle$ means a concatenation, and not inner product.

Q1

External Resources: https://crypto.stackexchange.com/questions/11612/how-can-an-encryption-scheme-be-indistinguishable-with-multiple-messages-but-vul for b

Collaborators: None

a

We want to show that a CPA-secure scheme implies that the scheme is also KPA-secure. To prove this, we assume by contradiction that there exists an non uniform PPT adversary $\{A_n\}_n$ such that A_n wins the t-message KPA-secure game with probability of $\frac{1}{2} + \varepsilon(n)$, where $\varepsilon(n)$ is a non-negligible function and t is some polynomial, for infinitely many $n \in \mathbb{N}$. Fix such an n as described, we use A_n to build an adversary B_n that wins the CPA-secure game. We know that A_n outputs its message with no knowledge of any of the generated keys. We denote the message outputted by $(m_{0,1}, \dots m_{0,t}), (m_{1,1}, \dots, m_{1,t})$. We note that by definition of the KPA-secure game if we denote:

$$S_0 = \{(pk, E_{ek}(m_{0,1}), \dots, E_{ek}(m_{0,t})) \mid pk, ek, sk \leftarrow Gen(1^n)\}$$

$$S_1 = \{(pk, E_{ek}(m_{1,1}), \dots, E_{ek}(m_{1,t})) \mid pk, ek, sk \leftarrow Gen(1^n)\}$$

Then, we know by our assumption that,

$$\Delta_{A_n}(S_0, S_1) = \varepsilon(n)$$

We define the following hybrids:

$$H_i = \{(pk, E_{ek}(m_{0,1}), \dots, E_{ek}(m_{0,i-1}), E_{ek}(m_{1,i}), \dots, E_{ek}(m_{1,t})) \mid pk, ek, sk \leftarrow Gen(1^n)\}$$

We note that $H_0 = S_1$, $H_{t+1} = S_0$, thus we know by the triangle inequality that there exists an $i \in [t]$ such that $\Delta_{A_n}(H_i, H_{i+1}) \geq \frac{\varepsilon(n)}{t+1}$. We use this i to devise our B_n (which is possible due to the non uniformity of B_n). B_n will perform as follows:

1. Given pk B_n activates A_n.
2. A_n outputs $(m_{0,1}, \dots m_{0,t}), (m_{1,1}, \dots, m_{1,t})$.
3. B_n will ask for encryptions of $r_1 = E_{ek}(m_{0,1}), \dots, r_{i-1} = E_{ek}(m_{0,i-1}), r_{i+1} = E_{ek}(m_{0,i+1}), \dots r_t = E_{ek}(m_{0,t})$ by the oracle access it's given.
4. B_n will then submit $m_{0,i}, m_{1,i}$ as its m_0, m_1.
5. When given $r_i = E_{ek}(m_b)$ for some b. B_n will feed $(pk, r_1, \dots r_t)$ to A_n.
6. When A_n outputs B_n will output 0 if A_n outputs $i+1$ and 1 if A_n outputs i.

It is clear the B_n is PPT since it makes $t-1$ queries, and uses A_n which is also PPT. We also note here that when B_n is given a sampling of $E_{ek}(m_{0,i})$ A_n receives a sampling of H_{i+1} and when B_n is give a sampling of $E_{ek}(m_{1,i})$ then A_n receives a sampling of H_i, this means that:

$$\Delta_{B_n}\left(\left\{E_{ek}(m_{0,i}) \mid pk, ek, sk \leftarrow Gen\left(1^n\right)\right\}, \left\{E_{ek}(m_{1,i}) \mid pk, ek, sk \leftarrow Gen\left(1^n\right)\right\}\right) = \Delta_{A_n}(H_i, H_{i+1}) \geq \frac{\varepsilon(n)}{t+1}$$

As we showed in class (or HW1) this implies that:

$$P\left[B_n \, wins \, CPA \mid pk, ek, sk \leftarrow Gen\left(1^n\right)\right] \geq \frac{1}{2} + \frac{\varepsilon(n)}{t+1}$$

Meaning we showed a non uniform PPT adversary $\{B_n\}_n$ that wins the CPA-secure game with non-negligible advantage $\frac{\varepsilon(n)}{t+1}$. This contradicts the fact (G, E, D) is CPA-secure, meaning our assumption is wrong and (G, E, D) is also KPA-secure, as we wanted to show. ■

b

We want to show that if t-message KPA-secure (for any polynomial t) scheme exists, then there exists schemes that are not CPA-secure. If t-message KPA-secure exists, then OWF exists. As we shown in class, this means that there exists a PRF. Let there be $F_k : \{0,1\}^n \to \{0,1\}^n$ pseudo-random keyed function (For example the GGM PRF definition qualifies).We have seen in class the following secret key encryption scheme (G, E, D):

- $sk \leftarrow G(1^n)$ where $sk \leftarrow \{0,1\}^n$.
- $E_{sk}(m) = \langle r, F_{sk}(r) \oplus m \rangle$ where $r \leftarrow \{0,1\}^n$ (if we get 0^n we sample again).
- $D_{sk}(\langle r, c \rangle) = c \oplus F_{sk}(r)$

As we have shown in class, this scheme is t-message KPA-secure for every polynomial t (which we have called in class multi-message secure). We modify this scheme to be only t-message KPA-secure, and not CPA secure. We define (G', E', D') the following way:

- $sk_1, sk_2 \leftarrow G'\left(1^n\right)$ where $sk_1, sk_2 \leftarrow \{0,1\}^n$
- $E'_{(sk_1,sk_2)}(m) = \begin{cases} \langle E_{sk_2}(m), sk_1 \rangle & m \neq sk_1 \\ \langle m, sk_1 \rangle & m = sk_1 \end{cases}$
- $D'_{(sk_1,sk_2)}(\langle r, c, sk_1 \rangle) = \begin{cases} c & |r| = 0 \\ D_{sk_2}(\langle r, c \rangle) & |r| \neq 0 \end{cases}$

It is clear that the scheme is correct. We want to show that this scheme is still t-message KPA-secure. Let there be t a polynomial and a non uniform PPT adversary $\{A_n\}_n$ in the KPA-secure game. Then, A_n outputs t pair of message $(m_{0,1}, \ldots, m_{0,t})$, $(m_{1,1}, \ldots, m_{1,t})$ independently of the secret keys sampling. Thus, since the keys are sampled at random,

$$P_{sk_1,sk_2 \leftarrow G'(1^n)}\left[\exists i \in \{0,1\}, j \in [t] \, s.t \, m_{i,j} = sk_1\right] \leq \frac{2 \cdot t(n)}{2^n}$$

Where as when $\forall i \in \{0,1\}, j \in [t] \, m_{i,j} \neq sk_1$, A_n will receive,

$$\left(\langle E_{sk_2}(m_{b,1}), sk_1 \rangle, \ldots, \langle E_{sk_2}(m_{b,t}), sk_1 \rangle\right)$$

For some b .Meaning that in this case we get a distribution that is statistically identical to a one of an adversary A'_n in a t-message KPA-secure game against (G, E, D). Thus A_n can only win this by $\frac{1}{2} + \mu(n)$ where μ is negligible. Thus,

$$P\left[A_n \, win \, KPA \, againt \, (G', E', D')\right] =$$

$$\left(P_{sk_1,sk_2 \leftarrow G'(1^n)}\left[\exists i \in \{0,1\}, j \in [t] \, s.t \, m_{i,j} = sk_1\right] \cdot\right.$$

$$P_{sk_1,sk_2\leftarrow G'(1^n)}\left[A_n \, win \, KPA \, against \, (G',E',D') \mid \exists i \in \{0,1\}, j \in [t] \, s.t \, m_{i,j} = sk_1\right]+$$

$$P_{sk_1,sk_2\leftarrow G'(1^n)}\left[\forall i \in \{0,1\}, j \in [t] \, s.t \, m_{i,j} \neq sk_1\right] \cdot P_{sk_1,sk_2\leftarrow G'(1^n)}\left[A_n \, win \, KPA \, against \, (G,E,D)\right] \leq$$

$$\frac{2 \cdot t(n)}{2^n} \cdot 1 + 1 \cdot \left(\frac{1}{2} + \mu(n)\right) \leq \frac{1}{2} + \mu'(n)$$

Where μ' is negligible. Thus we have showed that the scheme is t-message KPA-secure (for any polynomial t). We show now that it not CPA-secure. We describe the following non uniform PPT adversary adversary $\{B_n\}_n$:

1. B_n will sample $m \leftarrow \{0,1\}^n$ and ask for it encryption, $\langle r, c, sk_1 \rangle$
2. B_n chooses $m_0 = sk_1$, $m_0 \neq m_1 \leftarrow \{0,1\}^n$.
3. When receiving $E'_{(sk_1,sk_2)}(m_b) = \langle r, c, sk_1 \rangle$, B_n outputs 0 if and only if $|r| = 0$.

We want to show that this adversary wins the CPA-secure game with probability 1. This is clear since we know by the definition of (G', E', D') that $|r| = 0$ if and only if sk_1 was encrypted. Thus B_n outputs 0 if and only if $b = 0$. This means that B_n wins the CPA-secure game with probability 1. Thus we have shown a t-message KPA-secure scheme (for any polynomial t) that is not CPA-secure, as we wanted to show. ■

c

We want to show that any PKBE that is 1-message KPA-secure is CPA-secure. We prove this by contradiction. We assume a non uniform PPT adversary $\{A_n\}_n$ where A_n wins the CPA game with probability of $\frac{1}{2} + \varepsilon(n)$ for infinitely many $n \in \mathbb{N}$, where $\varepsilon(n)$ is a non-negligible function. Fix such an n. We use A_n to build an adversary B_n which wins the 1-message KPA-secure game. B_n works as follows:

1. B_n will output $m_{0,1} = 0$, $m_{1,1} = 1$
2. Given pk and $E_{ek}(m_{b,1})$, B_n will simulate A_n. If and when A_n asks for an encryption of $m \in \{0,1\}$ B_n will calculate $E_{ek}(m) = E_{pk}(m)$ and feed it to A_n .This is possible since (G, E, D) is a PKBE. Of course B_n will need to sample some randomness to calculate $E_{ek}(m)$, but it will be polynomial bounded since E is efficient.
3. A_n will submit two message m_0,m_1, since there are only two messages, we know that they must be 0,1. Thus, we can assume that without loss of generality $m_0 = 0$, $m_1 = 1$ (otherwise, we just need to negate the output bit of A_n in the end).
4. B_n will feed A_n with $E_{ek}(m_{b,1})$ as $E_{ek}(m_b)$ in the CPA-secure game.
5. If A_n queries for more $E_{ek}(m)$ for some $m \in \{0,1\}$, B_n will calculate them as in 2 and feed them to A_n.
6. When A_n outputs b', B_n will output b' as well.

It is clear the B_n is PPT since it simulates A_n which is PPT. We want to that B_n wins the 1-message KPA-secure game with probability of $\frac{1}{2} + \varepsilon(n)$. We note that B_n simulates the view of A_n in the CPA-secure game exactly (this is possible as B_n can know which messages A_n will submit since there are only two message, and it can encrypt messages since it is given $ek = pk$). Since $m_0 = m_{0,1}$, $m_1 = m_{1,1}$ we get that B_n wins the 1-message KPA-secure game if and only if A_n wins the CPA-secure game. Meaning, B_n wins the 1-message KPA-secure game with probability $\frac{1}{2} + \varepsilon(n)$. This is a contradiction to the fact that (G, E, D) is 1-message KPA-secure. Hence, our assumption is wrong, meaning (G, E, D) is a CPA-secure PKBE, as we wanted to show. ■

Q2

External Resources None
Collaborators: None

We want to show that the scheme is indeed a valid commitment scheme. We need to show that it is binding and hiding.

Binding

Let there be some $r \in \{0,1\}^{3n}$, if there exists $s_0, s_1 \in \{0,1\}^n$ such that $Com(0;s_0) = Com(1;s_1)$ for this r, this means that:

$$G(s_0) = r \oplus G(s_1)$$

$$G(s_0) \oplus G(s_1) = r$$

for these s_0, s_1. We know that there exists 2^{2n} couples of s_0, s_1. Thus the size of $H = \{G(s_0) \oplus G(s_1) \mid s_0, s_1 \in \{0,1\}^n\}$ is less equal to 2^{2n}. Hence,

$$P_{r \leftarrow U_{3n}}\left[\exists s_0, s_1 \, s.t \, Com(0;s_0) = Com(1;s_1)\right] = P_{r \leftarrow U_{3n}}\left[r \in H\right] \underset{(1)}{=} \frac{|H|}{2^{3n}} \leq \frac{2^{2n}}{2^{3n}} = 2^{-n}$$

Where one is due to the fact the r is sampled at random from $\{0,1\}^{3n}$. Thus, the scheme is binding (with no assumption as to the computational abilities of S).

Hiding

We assume by contradiction that there exists an $r \in \{0,1\}^{3n}$ such that for infinitely many $n \in \mathbb{N}$ the non uniform PPT distinguisher $\{B_n\}_n$ distinguishes $Com(0;U_n)_r$, $Com(1;U_n)_r$ with non negligible advantage $\varepsilon(n)$. We fix such an n. We will use B_n to build a non uniform PPT adversary A_n such that A_n distinguishes $G(U_n)$,U_{3n} with non negligible advantage. Given $y \in \{0,1\}^{3n}$ A_n will work as follows:

1. A_n will sample a bit $b \leftarrow \{0,1\}$ at random.
2. If $b = 0$ A_n will call B_n with y. Otherwise, A_n will call B_n with $y \oplus r$.
3. When B_n returns b'. If $b' = b$ A_n will return 0, otherwise it will return 1.

It is clear that A_n is PPT since A_n is PPT. We now show that A_n indeed distinguishes with non negligible advantage. We note that:

$$P\left[A_n(G(x)) = 0 \mid x \leftarrow \{0,1\}^n\right] = \frac{1}{2}P\left[B_n(G(x)) = 0 \mid x \leftarrow \{0,1\}^n\right] + \frac{1}{2}P\left[B_n(G(x) \oplus r) = 1 \mid x \leftarrow \{0,1\}^n\right] =$$

$$\frac{1}{2}P\left[B_n(t) = 0 \mid t \leftarrow Com(0;U_n)_r\right] + \frac{1}{2}P\left[B_n(t) = 1 \mid t \leftarrow Com(1;U_n)_r\right] = \frac{1}{2} \pm \varepsilon(n)$$

Where as:

$$P\left[A_n(x) = 0 \mid x \leftarrow \{0,1\}^{3n}\right] = \frac{1}{2}P\left[B_n(x) = 0 \mid x \leftarrow \{0,1\}^{3n}\right] + \frac{1}{2}P\left[B_n(x \oplus r) = 1 \mid x \leftarrow \{0,1\}^{3n}\right] =$$

$$\frac{1}{2}P\left[B_n(x) = 0 \mid x \leftarrow \{0,1\}^{3n}\right] + \frac{1}{2}P\left[B_n(x) = 1 \mid x \leftarrow \{0,1\}^{3n}\right] = \frac{1}{2}$$

Thus,

$$\Delta_{A_n}(G(U_n), U_{3n}) = \left|P\left[A_n(G(x)) = 0 \mid x \leftarrow \{0,1\}^n\right] - P\left[A_n(x) = 0 \mid x \leftarrow \{0,1\}^{3n}\right]\right| = \left|\frac{1}{2} \pm \varepsilon(n) - \frac{1}{2}\right| = \varepsilon(n)$$

Meaning we showed a non uniform PPT distinguisher such that for infinitely many $n \in \mathbb{N}$ A_n distinguishes with non-negligible advantage. This is a contradiction to the that G is PRG. Hence, our assumption is wrong and for every r $\left\{Com(0;U_n)_{n,r}\right\}_n \approx_c \left\{Com(1;U_n)_{n,r}\right\}_n$, as we wanted to show. ■

Q3

External Resources: None

Collaborators: None

We describe a ZK proof system for the Sudoko game. Let P know the solution M to a given Sudoko puzzle, N (where M, N are $n^2 \times n^2$ matrices, and N possibly has empty cells) . The protocol is as follows:

1. P will sample a random permutation $\varphi : [n^2] \to [n^2]$. We denote $[\varphi(M)]_{i,j} = \varphi(M_{i,j})$. P will send to V a commitment, $Com(\langle \varphi(M), \varphi \rangle)$.
2. V samples $i \leftarrow [4]$, $j \leftarrow [n^2]$, and sends them to P.
3. If $i = 1$, P decommits the j^{th} row in $Com(\varphi(M))$.
4. Else if $i = 2$, P decommits the j^{th} column in $Com(\varphi(M))$.
5. Else if $i = 3$, P decommits the j^{th} subs-square in $Com(\varphi(M))$.
6. Else (meaning $i = 4$), P decommits the known squares (originally filled ones) in $Com(\varphi(M))$ and φ
7. For $i = 1, 2, 3$ V verifies that the j^{th} row, column, sub-square is valid (meaning all the number in $[n^2]$, exists), and accepts if and only if it is.
8. For $i = 4$, V verifies that the φ^{-1} of the decommited square is equal to N (original puzzle). V accepts if and only if the validation is successful.

We show this is indeed a ZK proof.

- **Completeness -** It is easy to see that for a valid solution M for every (i, j) V will accept with probability 1 (since a permutation of a valid solution is also a valid solution of some Sudoko puzzle, so each row,column and sub-square are valid, and if M is a valid solution to N $\varphi^{-1}(\varphi(M))$ and N must coincide). Meaning, the proof system has perfect completeness.
- **Soundness -** Let there be a malicious prover P^*. We know that P^* commit an invalid solution M^*. This means that there exists a row, column or sub-square that is invalid in M^*, or that M^* doesn't coincide with N. Since the Com scheme is perfectly binding, and φ is a committed permutation, the commitment fixes the value of M^* for V (although he doesn't know them). If M^* doesn't coincide with N, V will rejects with probability $\frac{1}{4}$ (when $i = 4$). Otherwise, if there exists a row,column or sub-square that are invalid, V will find out with probability $\frac{1}{4} \cdot \frac{1}{n^2}$ (by selecting the correct (i, j) which are sampled independently and uniformly). Meaning that for and invalid M^* V rejects with probability of at least $\frac{1}{4} \cdot \frac{1}{n^2}$. Thus

$$P[\langle P^*, V \rangle (N, M^*) = 1] = 1 - P[\langle P^*, V \rangle (N, M^*) = 0] \leq 1 - \frac{1}{4n^2} \leq 1 - n^{-\Omega(1)}$$

- **Expected Polynomial Time Simulator -** Let there be a V^* (with arbitrary long output) . We describe the simulator S, when given N:
  1. S samples $i' \leftarrow [4]$, and a random permutation $\varphi : [n^2] \to [n^2]$.
  2. IF $i' = 1$ S commits $\left\langle \varphi\left(\begin{bmatrix} 1 & \cdots & n^2 \\ \vdots & \cdots & \vdots \\ 1 & \cdots & n^2 \end{bmatrix}\right), \varphi \right\rangle$.
  3. Else if $i' = 2$ S commits $\left\langle \varphi\left(\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \cdots & \vdots \\ n^2 & \cdots & n^2 \end{bmatrix}\right), \varphi \right\rangle$.

4. Else if $i'=3$ S commits $\left\langle \varphi\left(\begin{bmatrix} 1 & \cdots & n & \dots & 1 & \cdots & n \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ n(n-1) & \cdots & n^2 & \cdots & n(n-1) & \cdots & n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \cdots & n & \dots & 1 & \cdots & n \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ n(n-1) & \cdots & n^2 & \cdots & n(n-1) & \cdots & n^2 \end{bmatrix}\right), \varphi \right\rangle$.
5. Else, meaning $i'=4$, S commits $\langle \varphi(N), \varphi \rangle$ (N has empty squares in it that stay empty under φ).
6. When given (i,j) from V^* if $i=i'$, S decommits appropriately (send the j^{th} decommited row,column, sub-square for $i=1,2,3$ accordingly, or the known squares in N and φ for i=4), and concludes the simulation.
7. If $i \neq i'$ S returns to 1 and performs the simulation again.

■