Fundamentals of Cryptography: Midterm

Wednesday Nov 5, 3-5PM

Problem 4.

(a)(b)(e)

Problem 5.

(e)(b)(c)(d)(a)

Problem 6.

(a)(b)
A counter-example for (c): $f_0 = f_1$.
A counter-example for (d): f_0 is a strong PRP, and f_1 is the inverse of f_0.

Problem 7.

Part A. Let b_i denote the first bit of s_i. Then $b_1, \ldots, b_{\ell+\lambda}$ is a linear function of s. The linear function is determined by M. Let V denote the output space of the linear function, then $\dim V \leq \ell$.

In particular $(b_1, \ldots, b_{\ell+\lambda}) \in V$. If the distinguisher knows M (therefore knows V), it can easily distinguish $(b_1, \ldots, b_{\ell+\lambda})$ from random.

If M is unknown, note that $(b_i, \ldots, b_{i-1+\ell+\lambda}) \in V$ for any i. So the distinguisher can check if there is a ℓ-dim subspace that covers all $(\ell + \lambda)$-bit consecutive substrings. The output of G_M will always pass this test. But a sufficiently long random string (e.g., $(\ell + 1)(\ell + \lambda)$-bit long) will not pass this test will overwhelming probability.

Part B. CSS stream cipher is not a (secure) PRG.

Sketch: CSS stream cipher improves the security relying on non-linear operations (addition mod 256 is NOT linear over the boolean field). But the lowest bit of the addition is the XOR of the inputs' lowest bits.

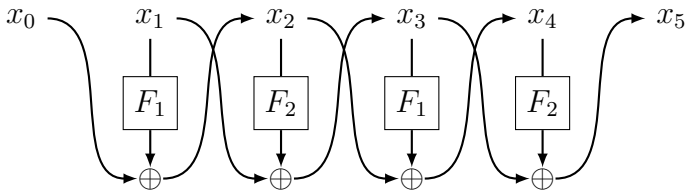
Assume CSS stream cipher is a secure PRG. Define a new generator G' that outputs the lowest bit of every output byte of CCS stream cipher, G' must also be a secure PRG. But G' is a linear PRG analyzed in Part A, and thus is not secure.

Problem 8.

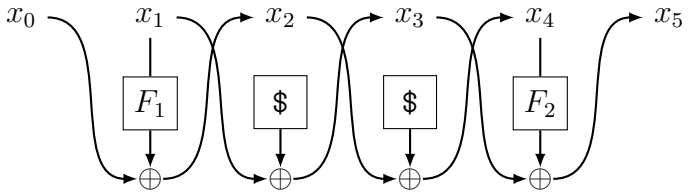
F''' is a strong PRP.

W.l.o.g., we assume the distinguisher never queries known. Let $x_0^i, x_1^i, x_2^i, x_3^i, x_4^i, x_5^i$ denotes the input, intermediate value, output corresponding to the i-th query.

- Real world: The distinguisher has oracle access to $\text{Feistel}_{F(k_1,\cdot),F(k_2,\cdot),F(k_1,\cdot),F(k_2,\cdot)}$.
- World 1: PRFs are replaced by random functions. The distinguisher has oracle access to $\text{Feistel}_{F_1,F_2,F_1,F_2}$.

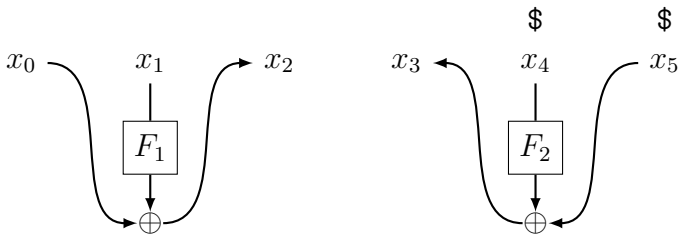


- World 2: In the two middle rounds, random functions are further replaced by a "random box". Upon a query, it will always sample fresh random output.



- World 3: In a forward query: x_2^i is computed, and is ignored. x_4^i, x_5^i are freshly uniformly sampled. x_3^i is reversed computed.

  Symmetrically for backword queries.



- Ideal world: The distinguisher has oracle access to a random permutation $\{0,1\}^{2\lambda} \to \{0,1\}^{2\lambda}$.

The real world and World 1 are indistinguishable due to the security of PRF. World 2 and World 3 are identical. The ideal world and World 3 are statistically indistinguishable. It remains to bound the distinguishing advantage between World 1 and World 2.

If $\{x_2^i\}$ are distinct and do not intersects with $\{x_4^i\}$, then replacing F_2 in the second round makes no difference. Similarly, if $\{x_3^i\}$ are distinct and do not intersects with $\{x_1^i\}$, then replacing F_1 in the third round makes no difference. Let Repeat denote the bad

event.

$$\mathsf{Repeat} = \begin{Bmatrix} x_2^i = x_2^j \text{ for some distinct } i, j \\ x_2^i = x_4^j \text{ for some } i, j \\ x_3^i = x_3^j \text{ for some distinct } i, j \\ x_3^i = x_1^j \text{ for some } i, j \end{Bmatrix}$$

Then Pr[Repeat] is the same in World 1 and World 2, and is an upper bound of the distinguishing advantage between World 1 and World 2.

It is easier to bound Pr[Repeat] in World 3. In World 3, the distinguisher learns no information about F_1, F_2, so the distinguisher can only make non-adaptive queries. The randomness of F_1, F_2 ensures x_2^i, x_3^i will not collides with other values with overwhelming probability.

Problem 9.

Π_{AE} is a secure authenticated encryption.

It is rather obvious that Π_{AE} is CPA-secure. If we replace $F(k_{PRF}, \cdot)$ by a truly random function (change the adversary's advantage by at most $\text{negl}(\lambda)$), the CPA-security of Π_{CPA} will protect the new scheme.

It remains to show that the adversary cannot generated a new valid ciphertext given the encryption oracle. Assume the adversary outputs a valid ciphertext c.

- Let $m\|v\|t = \mathsf{Dec}_{\text{CPA}}(k_{\text{CPA}}, c)$. Since the ciphertext is valid, $c = \mathsf{Enc}_{\text{AE}}(k, m; v)$, and t is a valid tag of $m\|v$.
- Since we a using a strongly secure MAC, with overwhelming probability, $(m\|v, t)$ must have been generated by the MAC algorithm before the adversary outputs c. In the game MAC is only invoked in an encryption query, and the oracle will give $\mathsf{Enc}_{\text{CPA}}(k_{\text{CPA}}, m\|v\|t; F(k_{PRF}, v))$ to the adversary.
- $\mathsf{Enc}_{\text{CPA}}(k_{\text{CPA}}, m\|v\|t; F(k_{PRF}, v)) = \mathsf{Enc}_{\text{AE}}(k, m; v) = c$. So the adversary fails to generate a new valid ciphertext.

Problem 10.

The following two folder has the same hash value.

$$h(\texttt{"B"} \| h(0) \| \texttt{"C"} \| h(0) \| 1)$$

```
folder A1                    folder A2
| folder B                   | file B the content is 0
| file C the content is 0     | folder C
```

Problem 11.

h is not collision-resistant. h does not resist preimage attack on any given output.

For any z, here is an efficient algorithm that samples a random preimage in $h^{-1}(z)$.

- Randomly pick $x \oplus y = k$.
- Compute $y = P^{-1}(k, z \oplus k)$.
- Compute $x = k - y$.
- Output (x, y).