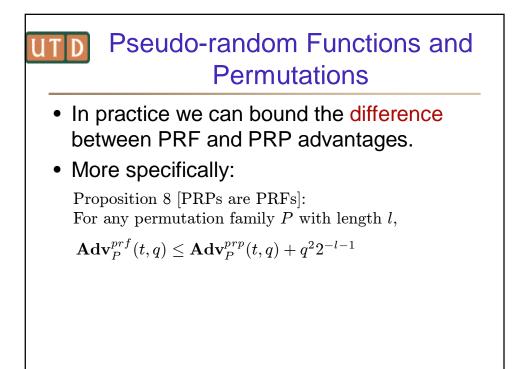


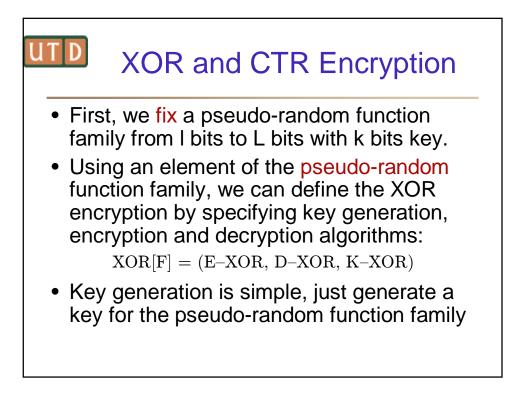
## **TD** Pseudo-random Functions and Permutations

• Now consider the best possible performance of a distinguisher.

$$\begin{aligned} \mathbf{Adv}_{F}^{prf}(t,q) &= max\{\mathbf{Adv}_{F,D_{fn}}^{prf}\} \\ \mathbf{Adv}_{P}^{prp}(t,q) &= \max_{D_{pn}}^{D_{fn}}\{\mathbf{Adv}_{P,D_{pn}}^{prp}\} \end{aligned}$$

• We informally say a block-cipher is secure if the best possible advantage of a distinguisher is low under reasonable time and query constraints

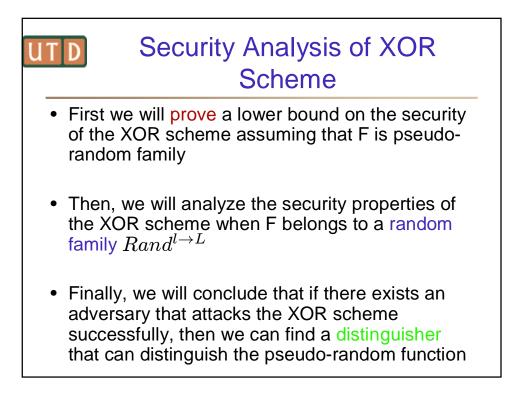




### XOR Encryption/Decryption

function E-XOR<sup>f</sup>(x)  $\mathbf{r} \leftarrow \{0, 1\}^l$ for i=1,...,n do  $\mathbf{y}_i = f(r+i) \oplus x_i$ return  $\mathbf{r} || y_1 y_2 ... y_n$ 

> function D-XOR<sup>f</sup>(z) Parse z as  $\mathbf{r} || y_1 y_2 ... y_n$ for i=1,...,n do  $\mathbf{x}_i = f(r+i) \oplus y_i$ return  $\mathbf{x} = \mathbf{x}_1 x_2 ... x_n$



#### Lower Bound on Insecurity

Proposition 9 [Lower bound on insecurity of XOR using a random function]: Suppose  $R = Rand^{l \leftarrow L}$ . Then, for any  $q_e, \mu_e$  such that  $\mu_e q_e/L \leq 2^l$ ,

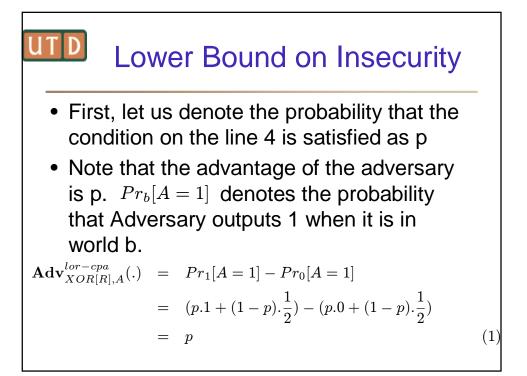
 $\mathbf{Adv}_{XOR[R]}^{lor-cpa}(.,t,q_e,\mu_e) \ge 0.316.\frac{\mu_e.(q_e-1)}{L.2^l}.$ 

#### Lower Bound on Insecurity

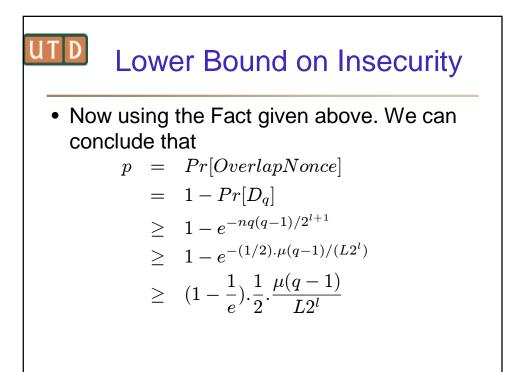
• To prove the claim, we specify an adversary:

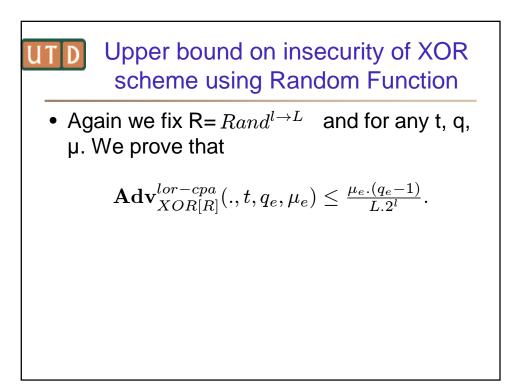
Algorithm  $A^{\mathcal{O}(\cdot,\cdot)}(k)$ 

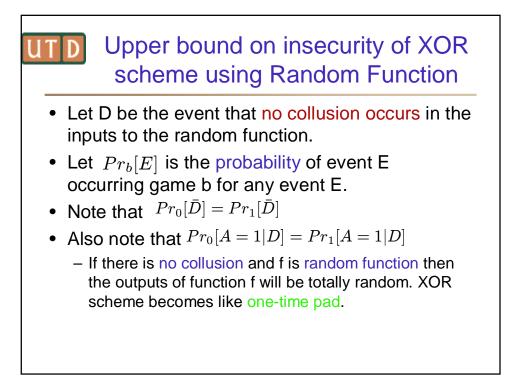
- (1) Let  $n = \mu/(Lq)$ . (This will be the number of blocks in all queried messages.)
- (2) Choose messages  $N_1, \ldots, N_q$ , all *n* blocks long, such that  $N_i[k] \neq N_j[k']$  for all  $i, j = 1, \ldots, q$ and  $k, k' = 1, \ldots, n$  satisfying  $(i, k) \neq (j, k')$ . (For example, set  $N_i[k]$  to the *L*-bit binary encoding of the integer n(i-1) + k for  $i = 1, \ldots, q$  and  $k = 1, \ldots, n$ .)
- (3) For  $i = 1, \ldots, q$  do:  $(r_i, y_i[1] \ldots y_i[n]) \leftarrow \mathcal{O}(0^{n l}, N_i)$ . We call  $r_i$  the *i*'th nonce.
- (4) If there is some  $i \neq j$  that  $|r_i r_j| < n$  (treat  $r_i, r_j$  as integers here!) then determine the values  $k, k' \in \{1, \ldots, n\}$  such that  $r_i + k = r_j + k'$ . Output  $\bigcirc$  if  $y_i[k] = y_j[k']$  and  $\bigcirc$  otherwise.
- (5) If there is no  $i \neq j$  that  $|r_i r_j| < n$ , output a coin flip.

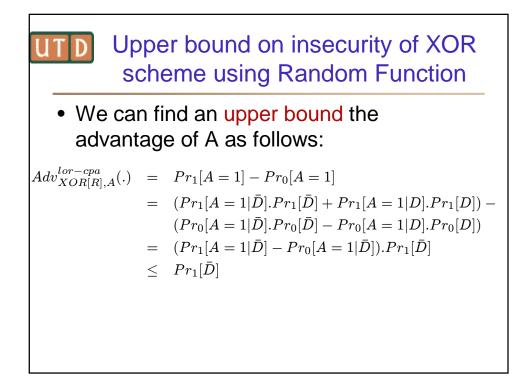


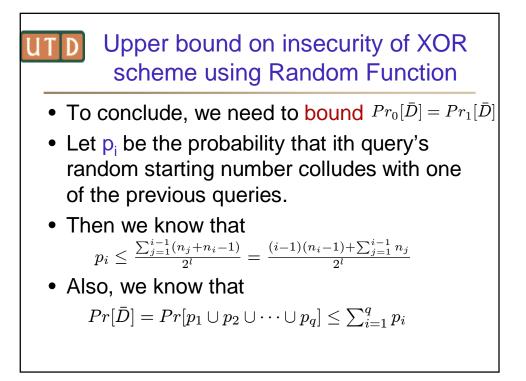
# • Use the set of the probability $D_i$ that ith query does not cause any overlap. • $Pr[D_{i+1}|D_i] \le \frac{2^l - in}{2^l} = 1 - \frac{in}{2^l}.$ • Fact For any real number x with $0 \le x \le 1$ we have $(1 - e^{-1})x \le 1 - e^{-x} \le x$ • Let us calculate an upper bound on prob. that no query overlaps. $Pr[D_q] = \prod_{i=1}^{q-1} Pr[D_{i+1}|D_i] \le \prod_{i=1}^{q-1} (1 - \frac{in}{2^l}) \le \prod_{i=1}^{q-1} e^{-in/2^l} = e^{-nq(q-1)/2^{l+1}}$

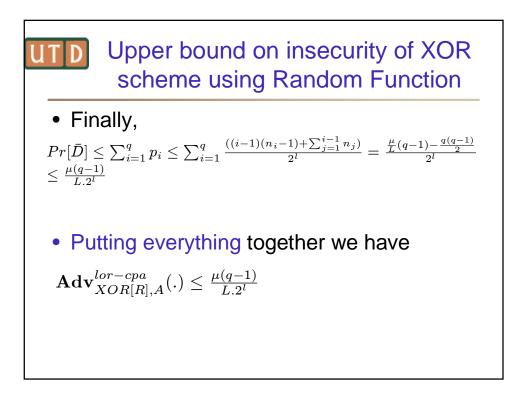


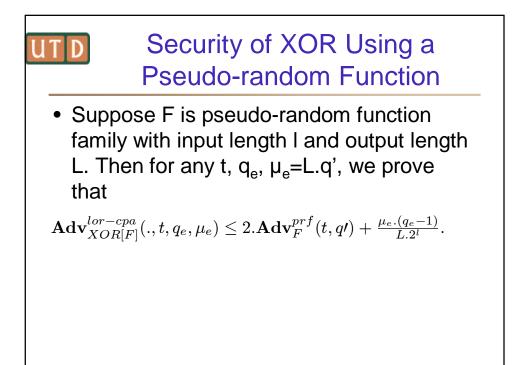


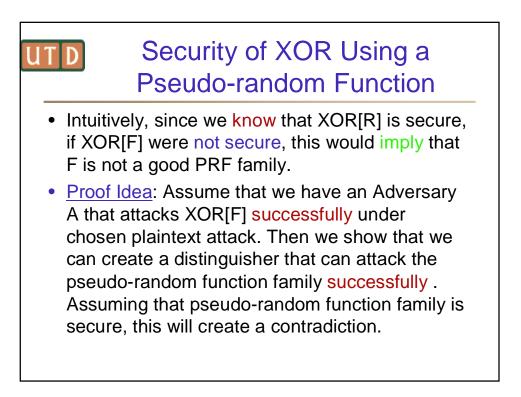


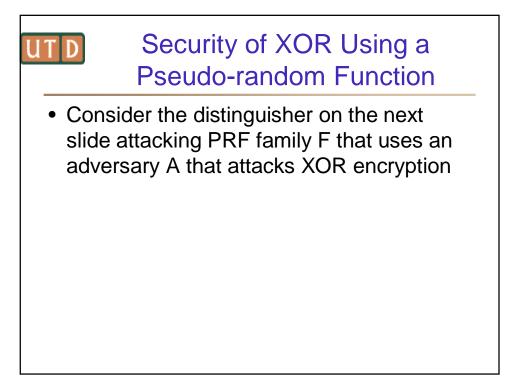












#### Security of XOR Using a Pseudo-random Function

Algorithm  $D^{f}(k)$ (1)  $b \leftarrow \{0,1\}$ . (This represents a choice to play either left or right oracle for A.) (2) Run A, responding to its oracle queries as follows. When A makes an oracle query  $(M_1, M_2)$ , let  $z \leftarrow \varepsilon - XOR^{f}(M_b)$ , and return z to A as the answer to the oracle query. (It is important here that D can implement the encryption function given an oracle for f.) (3) Eventually A stops and outputs a guess d to indicate whether it thought its oracle was the left oracle or the right oracle. If d = b then output 1, else output 0.

