Andrew Johnson

CS 427

HW4

|  |
| --- |
| Adv(m) |
| X → {0,1}λ |
| (r,s) = Enc(k,m) |
| r\* = x r |
| s\* = x r |
| m\* = Dec(k,(r\*,s\*)) |
| return m\* x == m |

1. (a) (b)

|  |
| --- |
| Dec(k,(r,s)) |
| c = s r |
| d = F-1(k,c) |
| Return d r |

P(real) = 1

P(fake) = 0

|  |
| --- |
| ADV() |
| t = GETMAC(k,0λ1λ) // F(k,0λ)||F(k,1λ) |
| t’ = tleft||tleft // F(k,0λ)|| F(k,0λ) |
| Return VER(0λ0λ,t’) |



P(real) = 1

P(fake) = 0

|  |
| --- |
| ADV(m1,m2) //m1,m2 = λ-bits |
| t = MAC(k, m1) |
| t’ = MAC(k, m2 t) |
| Return VER(m1||m2, t’) |



P(real) = 1

P(fake) = 0

1. (a) The proof will break down when we try to factor out Lmac-real because it needs to factor out k. If it does then k will become a private variable and ENC and DEC will throw an error since they would be trying to use a private variable.

|  |
| --- |
| ENC(k\*,m): |
| ke := F(k\*,0) |
| km := F(k\*,1) |
| c ← E.Enc(ke,mL) |
| t := M.MAC(km,c) |
| return (c,t) |
|  |
| DEC(k\*,m): |
| ke := F(k\*,0) |
| km := F(k\*,1) |
| if t M.MAC(km,c) |
| return err |
| return E.Dec(k­e­,c) |

|  |
| --- |
| ENC(k\*,m): |
| ke := e // e = {0,1}λ |
| km := m’ //m’ = {0,1}λ |
| c ← E.Enc(ke,mL) |
| t := M.MAC(km,c) |
| return (c,t) |
|  |
| DEC(k\*,m): |
| ke := e //same e as in ENC |
| km := m’ //same m’ as in ENC |
| if t M.MAC(km,c) |
| return err |
| return E.Dec(k­e­,c) |

(b)

≡

This is satisfied because F is a secure PRF.

Since ke­ and km are both in ENC and DEC and are the same in both functions. They can be used as private local variables of the library, and notice k\* isn’t used anymore. So we can take that out of the parameters if we want. Either way it brings us back to the original secure Encrypt-then-MAC construction, which we already proved. Thus the Modified Encrypt-then-MAC is CCA-Secure.