Learning Goals

- Understand the differences among different cryptographic primitives (PRG, PRF, encryption) and what security properties they provide.
- Be fluent reasoning about security definitions (proving security, breaking insecure constructions, interpreting security definitions).
- Know "standard" cryptographic constructions used in practice (OTP, secret sharing, block cipher modes) and understand their security properties.

Topics

- Unconditionally secure crypto
  - Defining security in terms of indistinguishable libraries
  - One-time pad
  - Threshold secret sharing (simple XOR-based construction; Shamir secret sharing based on polynomial interpolation)
- Computational security basics
  - Polynomial-time adversaries, negligible advantage, indistinguishable libraries
  - Pseudorandom generators (extending the stretch)
  - Pseudorandom functions
  - Pseudorandom permutations, a.k.a. block ciphers (Feistel construction)
- Encryption
  - Security against chosen plaintext attacks (unsuitability of deterministic encryption; revealing only the plaintext length)
  - Simple PRF-based scheme $\mathsf{Enc}(k, m) = (r, F(k, r) \oplus m)$
  - Block cipher modes (~~ECB~~, CBC, CTR)
  - Padding and padding-oracle attacks

Security Definitions

One-time secrecy for encryption:

$\mathcal{L}^{\Sigma}_{\text{ots-L}}$
$\underline{\text{QUERY}(m_L, m_R \in \Sigma.\mathcal{M})\text{:}}$
$k \leftarrow \Sigma.\mathsf{KeyGen}$
$c \leftarrow \Sigma.\mathsf{Enc}(k, m_L)$
return c

$\mathcal{L}^{\Sigma}_{\text{ots-R}}$
$\underline{\text{QUERY}(m_L, m_R \in \Sigma.\mathcal{M})\text{:}}$
$k \leftarrow \Sigma.\mathsf{KeyGen}$
$c \leftarrow \Sigma.\mathsf{Enc}(k, m_R)$
return c

t-out-of-n threshold secret sharing:

$\mathcal{L}^{\Sigma}_{\text{tsss-L}}$

$\textsc{query}(m_L, m_R \in \Sigma.\mathcal{M}, U)$:
  if $|U| \geq \Sigma.t$: return `err`
  $s \leftarrow \Sigma.\mathsf{Share}(m_L)$
  return $(s_i)_{i \in U}$

$\mathcal{L}^{\Sigma}_{\text{tsss-R}}$

$\textsc{query}(m_L, m_R \in \Sigma.\mathcal{M}, U)$:
  if $|U| \geq \Sigma.t$: return `err`
  $s \leftarrow \Sigma.\mathsf{Share}(m_R)$
  return $(s_i)_{i \in U}$

Security of a pseudorandom generator:

$\mathcal{L}^{G}_{\text{prg-real}}$

$\textsc{query}()$:
  $s \leftarrow \{0,1\}^{\lambda}$
  return $G(s)$

$\mathcal{L}^{G}_{\text{prg-rand}}$

$\textsc{query}()$:
  $z \leftarrow \{0,1\}^{\lambda+\ell}$
  return z

Security of a pseudorandom function:

$\mathcal{L}^{F}_{\text{prf-real}}$

$k \leftarrow \{0,1\}^{\lambda}$

$\textsc{query}(x \in \{0,1\}^{\text{in}})$:
  return $F(k, x)$

$\mathcal{L}^{F}_{\text{prf-rand}}$

$T :=$ empty assoc. array

$\textsc{query}(x \in \{0,1\}^{\text{in}})$:
  if $T[x]$ undefined:
    $T[x] \leftarrow \{0,1\}^{\text{out}}$
  return $T[x]$

CPA security of encryption:

$\mathcal{L}^{\Sigma}_{\text{cpa-L}}$

$k \leftarrow \Sigma.\mathsf{KeyGen}$

$\textsc{challenge}(m_L, m_R \in \Sigma.\mathcal{M})$:
  if $|m_L| \neq |m_R|$ return null
  $c := \Sigma.\mathsf{Enc}(k, m_L)$
  return c

$\mathcal{L}^{\Sigma}_{\text{cpa-R}}$

$k \leftarrow \Sigma.\mathsf{KeyGen}$

$\textsc{challenge}(m_L, m_R \in \Sigma.\mathcal{M})$:
  if $|m_L| \neq |m_R|$ return null
  $c := \Sigma.\mathsf{Enc}(k, m_R)$
  return c

Pseudorandom ciphertexts in the presence of chosen plaintext attacks: ($\Sigma.\mathcal{C}(\ell)$ refers to the set of possible ciphertexts for plaintexts of length ℓ)

$\mathcal{L}^{\Sigma}_{\text{cpa\$-real}}$

$k \leftarrow \Sigma.\mathsf{KeyGen}$

$\textsc{challenge}(m \in \Sigma.\mathcal{M})$:
  $c := \Sigma.\mathsf{Enc}(k, m)$
  return c

$\mathcal{L}^{\Sigma}_{\text{cpa\$-rand}}$

$\textsc{challenge}(m \in \Sigma.\mathcal{M})$:
  $c \leftarrow \Sigma.\mathcal{C}(|m|)$
  return c