Signatures from trapdoor permutations (the full domain hash):

In order to appeal to security of TDP, we need that the argument to \( F^{-1}(td, \cdot) \) to be random

Idea: hash the message first and sign the hash value (often called "hash-and-sign")

- Another benefit: Allows signing long messages (much larger than domain size of TDF)

FDH construction:
- Setup: Sample \((pp, td) \leftarrow \text{Setup for the TDP and output } \) \(Vk = pp, sk = td\)
- Sign \((sk, m)\): Output \( \sigma = F^{-1}(td, H(m)) \)
- Verify \((vk, m, \sigma)\): Output 1 if \( F(pp, \sigma) = H(m) \) and 0 otherwise

Theorem. If \( F \) is a trapdoor permutation and \( H \) is an ideal hash function (i.e., "random oracle") then the full domain hash signature scheme defined above is secure.

Proof Idea: Signature is deterministic, so to succeed, adversary has to forge on an unqueried message \( m \).
- Signature on \( m \) is preimage of \( F \) at \( H(m) \)
- Adversary has to invert \( F \) at random input (when \( H \) is modeled as a random oracle)
- How to simulate signing queries?
- Relies on "programming" the random oracle

Recap: RSA-FDH signatures:
- Setup: Sample modulus \( N, e, d \) such that \( ed = 1 \text{ mod } \varphi(N) \) - typically \( e = 3 \) or \( e = 65537 \)
- Output \( vk = (N, e) \) and \( sk = (N, d) \)
- Sign \((sk, m)\): \( \sigma = H(m)^d \) [Here, we are assuming that \( H \) maps into \( \mathbb{Z}_N^* \)]
- Verify \((vk, m, \sigma)\): Output 1 if \( H(m) = \sigma^e \) and 0 otherwise

An aside: blind signatures from RSA [client can interact with a server to obtain signature on a message \( m \) without server learning the message that was signed]

\[
\begin{align*}
\text{client} & \quad \text{server} \\
(r, s) \in \mathbb{Z}_N^* & \quad (sk = d) \\
\text{client} & \quad \text{server} \\
y = H(m) \cdot r^e & \quad \text{server} \\
& \quad \text{server} \\
y^d & \quad s = \frac{r^e y^d}{r} \\
\text{server} & \quad \text{client} \\
\sigma = \frac{s}{r} & \quad \text{client}
\end{align*}
\]

Observe that \( \sigma = \frac{s}{r} = \frac{(H(m) \cdot r^e)^d}{r} = H(m) \cdot \frac{r^{ed}}{r} = H(m) \mod N \) [since \( ed = 1 \mod \varphi(N) \)]

Moreover, server does not learn the message: \( r^e \) is uniform over \( \mathbb{Z}_N^* \) [with all but negligible probability]
Standard: PKCS1 v1.5 (typically used for signing certificates)

- Standard cryptographic hash functions hash into a 256-bit space (e.g., SHA-256), but FDH requires full domain
- PKCS1 v1.5 is a way to pad hashed message before signing:

![Diagram]

- Padding important to protect against chosen message attacks (e.g., prepossess to find messages \( m_1, m_2, m_3 \) where \( H(m_1) \neq H(m_2) \cdot H(m_3) \))
  (but this is not a full-domain hash and cannot prove security under RSA — can make stronger assumption...)
Also possible to use RSA to build PKE:

"Textbook RSA" (How NOT to encrypt): Consider the following candidate of a PKE scheme from RSA:

- Setup: Sample \((N, e, d)\) where \(N = pq\) and \(ed \equiv 1 \pmod{\phi(N)}\). Output \(pk = (N, e)\) and \(sk = (N, d)\)
- Encrypt \((pk, m)\): \(c = m^e \iff \text{Correct since } c^d = (m^e)^d = m^d = m \pmod{N}\)
- Decrypt \((sk, c)\): \(m = c^d \) \(\iff \text{Correct since } c^d = (m^e)^d = m^d = m \pmod{N}\)

Correctness follows from correctness of TDP.

1. Security of TDP says that inverting random element should be difficult
   \(\iff\) Does not apply if messages chosen adversarially (e.g., semantic security definition)
   \(\iff\) Does not say anything about hiding preimage (e.g., \(F(pp, x)\) can leak information about \(x\) so long as leakage is not sufficient to fully recover \(x\) — this is a weaker property than full indistinguishability)

2. This scheme is deterministic: cannot be semantically secure!

NEVER use textbook RSA!

To use RSA/TDPs to construct a PKE scheme, we will use a similar strategy as in the FDH signature construction:

- Setup: Sample \(pp = (pp, td)\) \(\iff\) Setup for the TDP scheme and output \(pk = pp\) and \(sk = td\)
- Encrypt \((pk, m)\): Sample \(x \in X\) from domain of TDP
  \(\iff\) Scheme is randomized!
  Let \(k \leftarrow H(x)\) where \(H: X \rightarrow Km\) is an (ideal) hash function and \(Km\) is the key space for an
  symmetric authenticated encryption scheme
  Compute \(y \leftarrow F(pp, x)\) and \(ct = EncAEC(k, m)\)
  Output \((y, ct)\)
- Decrypt \((sk, ct' = (y, ct'))\): Compute \(x = F'(td, y)\), \(k = H(x)\), and output \(m = DecA(k, ct')\)

This is an example of hybrid encryption or KEM: \(y\) is used to encapsulate the key and \(ct'\) is an encryption under \(k\).

Theorem. If \(F\) is a trapdoor permutation and \(H\) is modeled as a random oracle, then the above encryption scheme is
semantically secure. [In fact, this scheme is CCA-secure in the random oracle model]

Proof intuition. Given a ciphertext \((y, ct')\) and public key \(pk = pp\):

- Adversary cannot compute \(X\) from \(y\) (by security of TDP — once \(X\) is uniform)
- Adversary cannot evaluate \(H\) on \(X\), so \(k\) is uniformly random and hidden from adversary
- Semantic security follows from semantic security of symmetric encryption scheme

RSA initialization:

- Setup: Sample \((N, e, d)\) where \(N = pq\) and \(ed \equiv 1 \pmod{\phi(N)}\). Output \(pk = (N, e)\) and \(sk = (N, d)\)
- Encrypt \((pk, m)\): Sample \(X \in \mathbb{Z}_N^2\) and compute \(y = X^e \pmod{N}\).\n  \(\iff\) Output \((y, ct)\)
  Compute \(k = H(x)\) and compute \(ct' = EncAEC(k, m)\).
- Decrypt \((sk, ct)\): Compute \(X = y^d \pmod{N}\), \(k = H(k)\), and output \(m = DecA(k, ct')\)