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 <u>random</u>

<u>Idea:</u> hash the message first and sign the hash value (often called "hash-and-sign")

↳ <u>Another benefit:</u> Allows signing long messages (much larger than domain size of TDF)

<u>FDH construction</u>:

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

<u>Theorem.</u> 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.

<u>Proof Idea:</u> Signature is <u>deterministic</u>, 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 <u>random</u> input (when H is modeled as a random oracle)

How to simulate signing queries?

↳ Relies on "programming" the random oracle

<u>Recap</u>: RSA-FDH signatures:

Setup: Sample modulus N, e, d such that $ed = 1 \pmod{\varphi(N)}$ — typically $e = 3$ or $e = 65537$

Some (partial) attacks can exploit very small public exponent $(e=3)$

Output $vk = (N, e)$ and $sk = (N, d)$

Sign (sk, m): $\sigma \leftarrow H(m)^d$ [Here, we are assuming that H maps into $\mathbb{Z}_N^*$]

Verify (vk, m, σ): 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]

$vk = (N, e)$

<u>client</u> | <u>server</u> $(sk = d)$

$r \xleftarrow{R} \mathbb{Z}_N$

$y = H(m) \cdot r^e$ →

← $z = y^d$

↓

$\sigma = z / r$

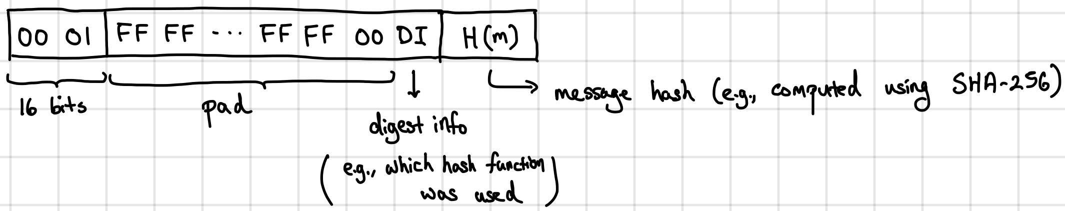
Observe that $\sigma = z/r = (H(m) \cdot r^e)^d / r = H(m) \cdot \frac{r^{ed}}{r} = H(m) \bmod N$ [since $ed = 1 \bmod \varphi(N)$]

Moreover, server does not learn the message: r^e is uniform over $\mathbb{Z}_N^*$ [with all but negligible probability] so it perfectly hides $H(m)$

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

↳ PKCS 1 v1.5 is a way to pad hashed message before signing:



↳ Padding important to protect against chosen message attacks (e.g., preprocess to find messages m_1, m_2, m_3 where $H(m_1) = H(m_2) \cdot H(m_3)$)
(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 $ed = 1 \pmod{\varphi(N)}$. Output $pk = (N, e)$ and $sk = (N, d)$
- Encrypt(pk, m): Output $c \leftarrow m^e$
- Decrypt(sk, ct): Output $m \leftarrow c^d$

Correct since $c^d = (m^e)^d = m^{ed} = m^1 = m \pmod N$

Correctness follows from correctness of TDP.

How about security? NO.

1. Security of TDP says that inverting random element should be difficult
   - ↳ Does not apply if messages chosen adversarially (e.g., semantic security definition)
   - ↳ 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!
   - ↳ in fact, vulnerable to message-recovery attacks in many settings

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 $(pp, td) \leftarrow$ Setup for the TDP scheme and output $pk = pp$ and $sk = td$
- Encrypt(pk, m): Sample $x \xleftarrow{R} \mathcal{X}$ from domain of TDP — Scheme is randomized!

  Let $k \leftarrow H(x)$ where $H : \mathcal{X} \to \mathcal{K}$ is an (ideal) hash function and $\mathcal{K}$ is the key-space for an symmetric authenticated encryption scheme

  Compute $y \leftarrow F(pp, x)$ and $ct' \leftarrow \mathrm{Enc}_{AE}(k, m)$

  Output (y, ct')
- Decrypt $(sk, ct' = (y, ct'))$: Compute $x \leftarrow F^{-1}(td, y)$, $k \leftarrow H(x)$, and output $m \leftarrow \mathrm{Dec}_{AE}(k, ct')$

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 — since 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 instantiation:

- Setup: Sample (N, e, d) where $N = pq$ and $ed = 1 \pmod{\varphi(N)}$. Output $pk = (N, e)$, $sk = (N, d)$
- Encrypt(pk, m): Sample $x \xleftarrow{R} \mathbb{Z}_N^*$ and compute $y \leftarrow x^e \pmod N$.

  Compute $k \leftarrow H(x)$ and compute $ct' \leftarrow \mathrm{Enc}_{AE}(k, m)$.

  Output (y, ct')
- Decrypt(sk, ct): Compute $x \leftarrow y^d \pmod N$, $k \leftarrow H(k)$, and output $m \leftarrow \mathrm{Dec}_{AE}(k, ct')$.