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. RSA says that computing e^{th} root of 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., x^e 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 to construct a PKE scheme, we will use a similar strategy as in the FDH signature construction:

- Setup: Sample $N = pq, e, d$ where $ed = 1 \pmod{\varphi(N)}$. $pk = (N, e)$, $sk = d$
- Encrypt(pk, m): Sample $x \xleftarrow{R} \mathbb{Z}_N^*$ (Scheme is randomized!)
  Let $k \leftarrow H(x)$ where $H: \mathbb{Z}_N^* \to \mathcal{K}$ is an (ideal) hash function and $\mathcal{K}$ is the key-space for an symmetric authenticated encryption scheme
  Compute $y \leftarrow x^e$ and $ct' \leftarrow \mathrm{Enc}_{AE}(k, m)$
  Output (y, ct')
- Decrypt$(sk, ct' = (y, ct'))$: Compute $x = y^d \pmod{N}$ $k \leftarrow H(x)$, and output $m \leftarrow \mathrm{Dec}_{AE}(k, ct')$

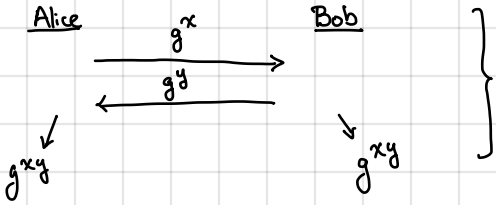
This is an example of hybrid encryption or KEM: y is used to encapsulate the key and ct' is an encryption under k

Theorem. If the RSA assumption holds 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 RSA — observe that x is uniform over $\mathbb{Z}_N^*$)
- 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.

Now that we have digital signatures, let's revisit the question of key exchange (with active security)



} completely vulnerable to an active network adversary that can intercept and inject packets

In addition, should guarantee that one compromised session should not affect other honest sessions
- Alice ↔ Eve should not compromise security of Alice ↔ Bob

Authenticated key exchange (AKE): provides security against active adversaries
- Requires a "root of trust" (certificate authority) → we need some binding between keys and identities

(one-time setup, at least for duration of validity period)

↑ the certificate binds Alice's public key pk_{Alice} to Alice's identity

- Certificates typically have the following format (X509):
  - Subject (entity being authenticated)
  - Public key (public key for subject for signature scheme)
  - CA: identity of the CA issuing the certificate
  - Validity dates for certificate
  - CA's signature on certificate

← the browser and operating system have a set of hard-coded certificate authorities and their respective public keys (usually several hundred authorities)

[public-key infrastructure (PKI)]