Combining symmetric and asymmetric encryption

Let's suppose we communicate with We could send our public key to Amazon and get their public key. Usually not what happens in reality.

Amazon doesn't normally encrypt each reply with a public key of yours. Instead, HTTPS creates a shared, temporary **session key**.

In simplified form:

1. Amazon sends its certificate, containing its public key.
2. Your browser verifies the certificate really belongs to Amazon.
3. Your browser and Amazon securely agree on a random session key.
4. Both directions use that same session key with fast symmetric encryption.

So once the connection is set up, your messages and Amazon's replies are both encrypted with the shared session key.

Modern HTTPS usually uses ephemeral Diffie-Hellman key exchange to create this key, rather than directly encrypting all traffic with Amazon's public key. This also helps protect past sessions if Amazon's private key were compromised later.