In my RSA.sol exercise, I hash the message first, then I sign the digest with the private key. Anyone can check the result with the public key. The goal is to prove the message really comes from the key owner.

With GitHub SSH, the idea is similar but the purpose is identity. I keep the private key on my laptop and I upload the public key to GitHub. When I connect, GitHub asks my computer to prove it owns the private key. My computer signs a challenge and GitHub verifies it using the public key. This is mainly for authentication, not for hiding the code I push.

Signing is used when you want proof and authorization. Encryption is used when you want secrecy.