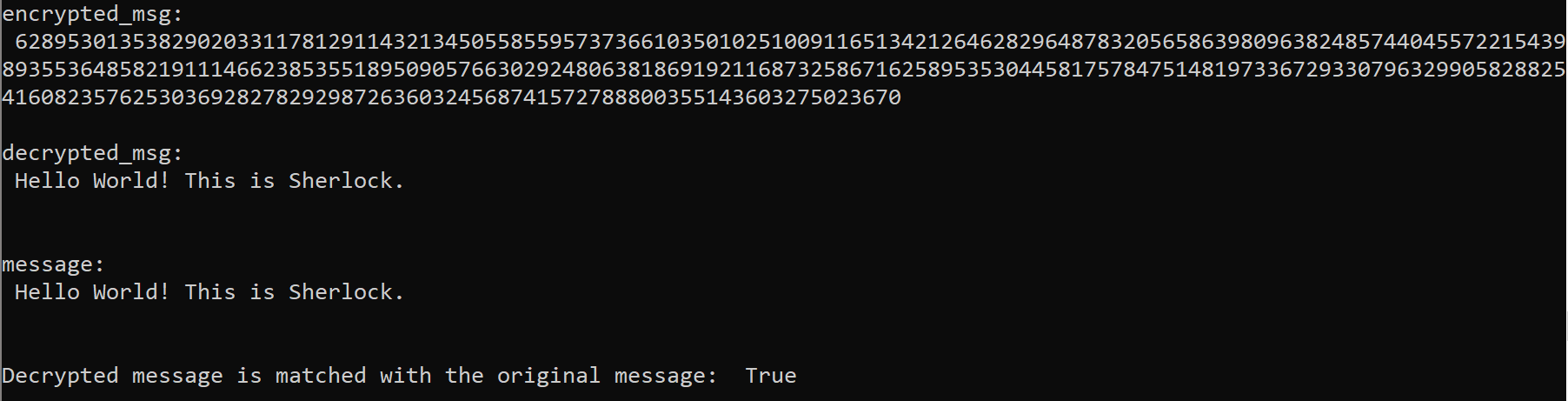
**Lab 7 report Wang Zijia 1002885**

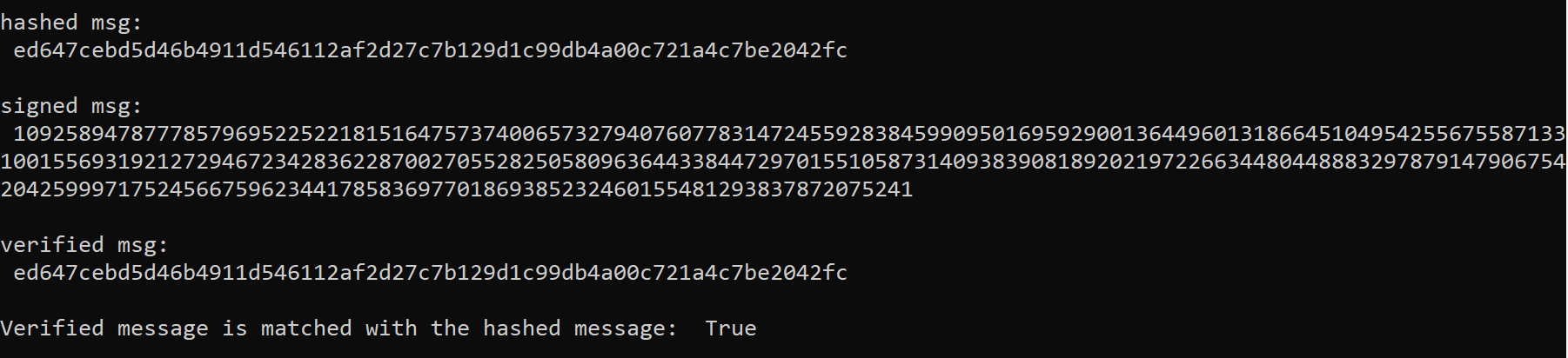
**Hand-in 1**

1. Demo encryption and decryption of RSA



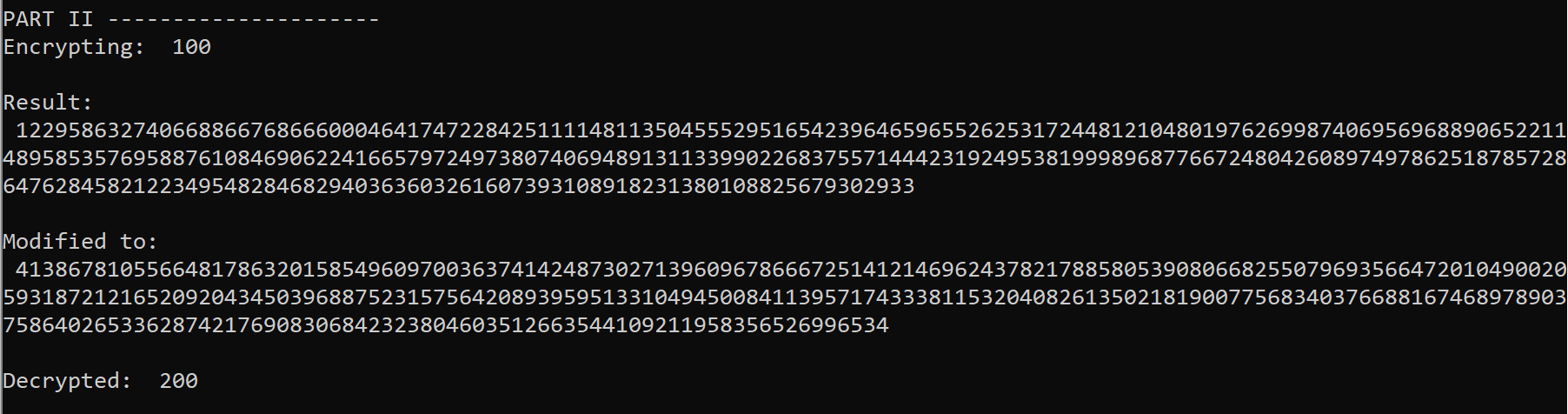
The decrypted message is identical to the original message.

1. Demo signing a message and verifying it



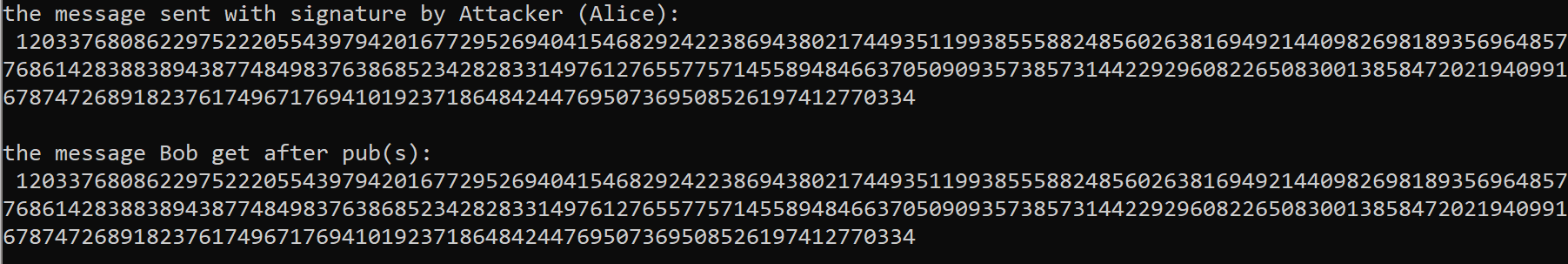
The verified message is identical to the hashed message.

1. Demo protocol attack to both encryption and digital signature



The original message (Alice sent this message to Bob) is 100.

After the attacker modified the message, Bob will receive a message with content of 200.



The attacker pretends that she herself is Alice and sends the fake message with fake signature to Bob. The attacker generates the signature first and gets the corresponding message by encrypting it using Alice’s public key.

When Bob receives the message and signature pair, he will encrypt the signature by using Alice’s public key which is identical to the message sent by the attacker.

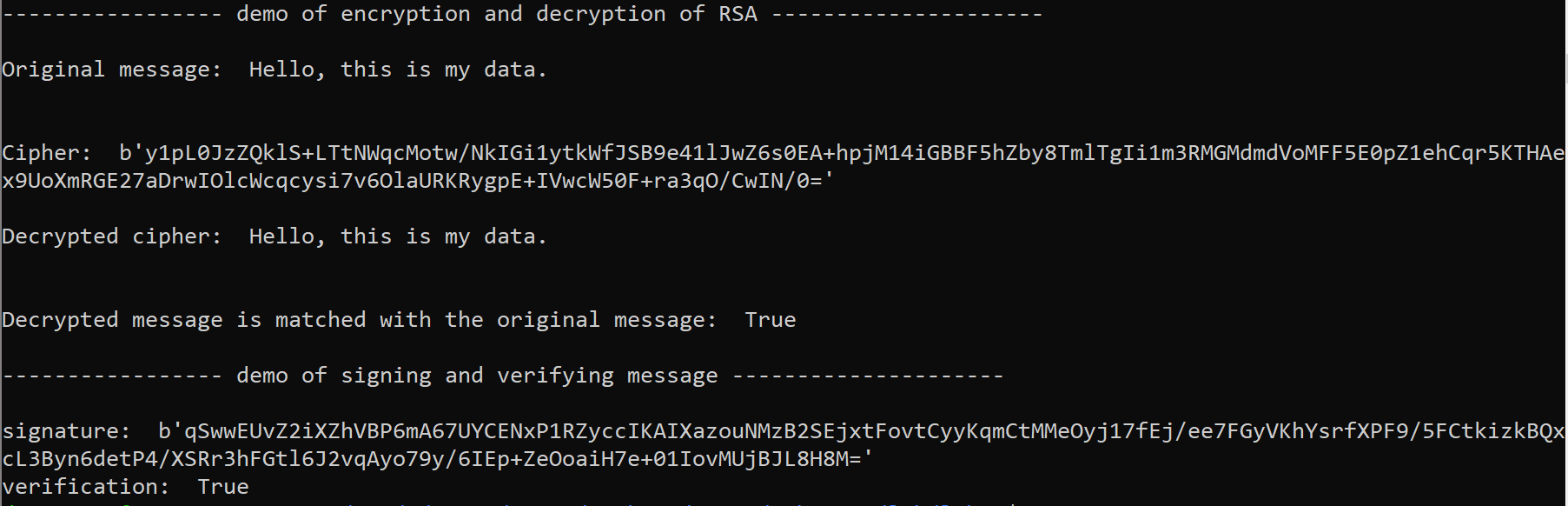
1. Explain the limitation of protocol attack

For the first attack, if the attacker has idea about the original message, then he can modify the message purposely. Otherwise, he can only modify the message randomly which means he cannot control the content of the modified message.

For the second protocol attack, since the attacker generated message from signature, the content of the message is also not controlled, which means by doing this attack, the attacker is not able to modify the message purposely.

**Hand-in 2**

1. Demo decryption of a message from a friend using RSA



1. Explain the purpose of Optimal Asymmetric Encryption padding (OAEP) to encrypt and decrypt using RSA. Explain how it works.

* Optimal Asymmetric Encryption padding (OAEP) uses **RSA encryption** and integrated a **padding scheme**.

OAEP uses a Feistel network with a pair of random oracles G and H. These operator on the plaintext before it is encrypted.

* Purpose:
  + Add **randomness** to the process
  + **Prevent partial decryption of ciphertexts** (or other information leakage) by ensuring that the adversary cannot recover and portion of the plaintext without being able to invert the trapdoor one-way permutation.
* How it works:

*n* is the number of bits in the RSA modulus.

*k*0 and *k*1 are integers fixed by the protocol.

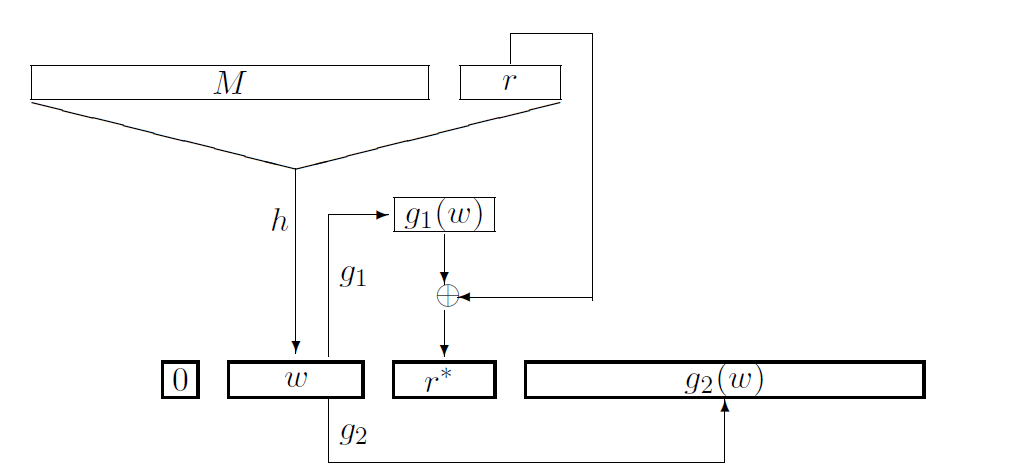
*m* is the plaintext message, an (*n* − *k*0 − *k*1)-bit string

*G* and *H* are [random oracles](https://en.wikipedia.org/wiki/Random_oracle) such as [cryptographic hash functions](https://en.wikipedia.org/wiki/Cryptographic_hash_function).

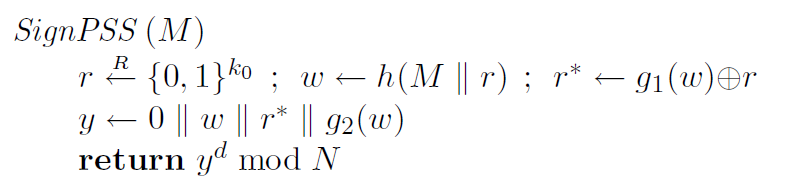
⊕ is an xor operation.

* + To encrypt:
    - messages are padded with *k*1 zeros to be *n* − *k*0 bits in length.
    - *r* is a randomly generated *k*0-bit string
    - *G* expands the *k*0 bits of *r* to *n* − *k*0 bits.
    - *X* = *m*00..0 ⊕ *G*(*r*)
    - *H* reduces the *n* − *k*0 bits of *X* to *k*0 bits.
    - *Y* = *r* ⊕ *H*(*X*)
    - The output is *X* || *Y* where *X* is shown in the diagram as the leftmost block and *Y* as the rightmost block
  + To decrypte:
    - recover the random string as *r* = *Y* ⊕ *H*(*X*)
    - recover the message as *m*00..0 = *X* ⊕ *G*(*r*)

1. Explain the purpose of Probabilistic Signature Scheme (PSS) to sign and verify using RSA. Explain how it works.



* Purpose:
  + PSS is randomized and will produce a **different signature value each time** (unless you use a zero-length salt).
* How it works:
  + To encrypt:
    - The signature scheme hashes the message to be signed by using a hash function.
    - The resulting hash is transformed into an encoded message.
    - A signature primitive is applied to the encoded message by using the private key to produce the signature.



* + To decrypt:
    - The scheme hashes the message to be signed by using the same hash function that was used to sign the message.
    - A verification primitive is then applied to the signature by using the public key of the key pair to recover the message.
    - The scheme verifies that the encoded message is a valid transformation of the hash value.

