INTE2401/2402 Lab 7

Student ID: _____ Name: _____

Big data is pervasive across scientific, medical and financial domains. The off-the-shelf cloud is leveraged to manage the explosive data due to its superior computation power and storage capacity. In spite of the considerable advantages of data outsourcing, it is difficult to overstate the acute privacy concerns raised by the growing data breaches. To protect the data privacy and security, both the academic and industry advocate integrating the crypto primitives to cloud-based systems.

In this week's lab, we demonstrate two off-premise data management frameworks. Resorting to the knowledge of encryption methods studied and exercised in the previous lectures and labs, we ensure the security of the data when exchanging. Specifically, in task 1, we illustrate the process of the digital signature for presenting the authenticity of messages. Besides, in task 2, we display the process of retrieving the message from the server in a private manner. Due to the time limitation, instead of implementing an entire system, we utilise online crypto tools to perform the workflow. The links of online crypto tools are provided as below.

Online AES encryption and decryption:
http://rubbingalcoholic.github.io/cowcrypt/demos/aes.html

Online RSA encryption and decryption:

http://www-cs-students.stanford.edu/~tjw/jsbn/rsa2.html

Online digital signature algorithm:

https://kjur.github.io/jsrsasign/sample/sample-rsasign.html

Noted that, both tasks are evaluated by filling the tables at the end of the tasks. You may hand the hardcopy to the on-duty lab tutor.

Task 1. Demo of message authentication by digital signature

Task 1 considers the message authentication service through RSA-based digital signature. Figure 1 (captured from lecture slides, week 5) illustrates the model of the digital signature. This model consists of two parties, the signer Bob, and the verifier Alice. Roughly speaking, Bob generates his digital signature specified for a given message and sends both message and signature to Alice. Alice then verifies the authenticity of the received message by comparing the hash of message with the decrypted signature.

In this task, we use the ***student ID*** as the message M, and we employ SHA-256 as the cryptography hash function. The online digital signature tool is provided above.

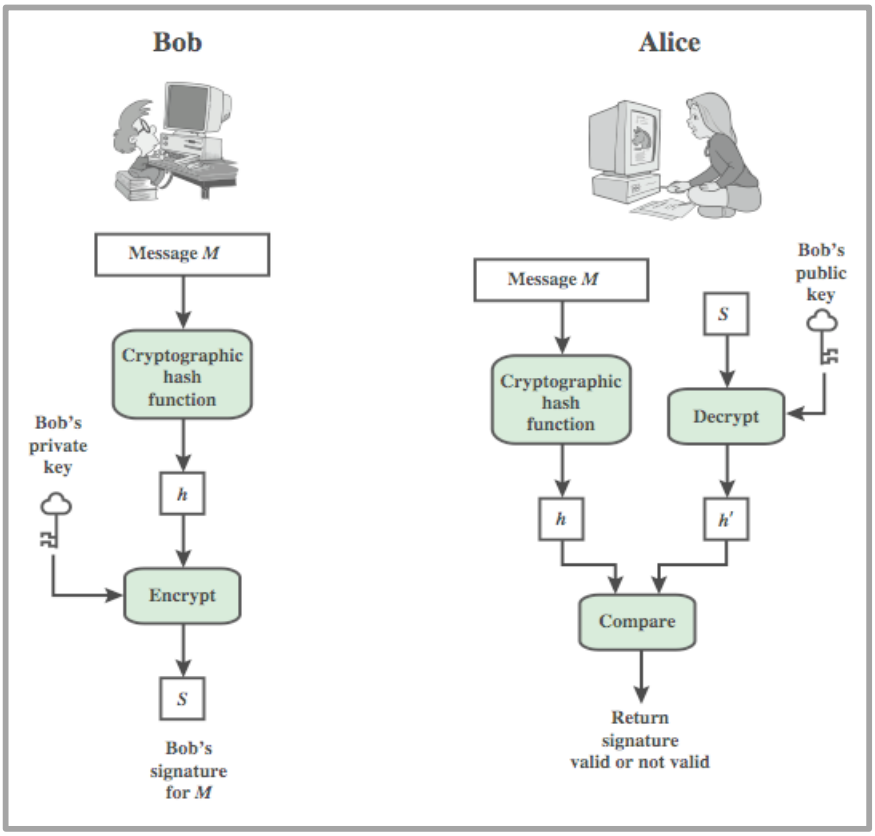

Figure 1. Model of digital signature

Question table:

Message M:	s3703131
RSA private key:	MIICWwIBAAKBgQDRhGF7X4A0ZVlEg594WmODVVUIiiPQs04aLmvfg8SborHss5gQ Xu0aIdUT6nb5rTh5hD2yfpF2WIW6M8z0WxRhwicgXwi80H1aLPf6lEPPLvN29EhQ NjBpkFkAJUbS8uuhJEeKw0cE49g80eBBF4BCqSL6PFQbP9/rByxdxEoAIQIDAQAB AoGAA9/q3Zk6ib2GFRpKDLO/O2KMnAfR+b4XJ6zMGeoZ7Lbpi3MW0Nawk9ckVaX0 ZVGqxbSIX5Cvp/yjHHpww+QbUFrw/gCjLiiYjM9E8C3uAF5AKJ0r4GBPl4u8K4bp bXeSxSB60/wPQFiQAJVcA5xhZVzqNuF3EjuKdHsw+dk+dPECQQDubX/lVGFgD/xY uchz56Yc7VHX+58BUkNSewSzwJRbcueqknXRWwj97SXqpnYfKqZq78dnEF10SWsr /NMKi+7XAkEA4PVqDv/OZAbWr4syXZNv/Mpl4r5suzYMMUD9U8B2JIRnrhmGZPzL x23N9J4hEJ+Xh8tSKVc80jOkrvGlSv+BxwJAaTOtjA3YTV+gU7Hdza53sCnSw/8F YLrgc6NOJtYhX9xqdevbyn1lkU0zPr8mPYg/F84m6MXixm2iuSz8HZoyzwJARi2p aYZ5/5B2lwroqnKdZBJMGKFpUDn7Mb5hiSgocxnvMkv6NjT66Xsi3iYakJII9q8C Ma1qZvT/cigmdbAh7wJAQNXyoizuGEltiSaBXx4H29EdXNYWDJ9SS5f070BRbAIl

	dqRh3rcNvpY6BKJqFapda1DjdcncZECMizT/GMrc1w==
Generated signature:	2fafed6edd2683a85ec99cfe589ea2b4168c94c5916440e944bcfd27ebdc8c7a 9009fc2b469695905bf4eb9f07027ba1d95234dabef3faa8a5077680f9b778de 57b1127659ca04b75e2e4ccb8eb22cb31ecbfde405cef52a5ea37309250a4d73 e83bf6898c9ccc67703c1ffec792be5d1fe907ed70b48e8f999cfcc3e50c6980

Task 2. Demo the process of getting student's profile

Task 2 considers a model of retrieving student's profile in a private manner. Figure 2 displays the workflow, containing two entities, the client student, and the RMIT servers. The cluster of RMIT servers store the student's sensitive profile and have the ability to perform RSA encryption and decryption, as well as the AES algorithm. Correspondingly, students have the ability to perform RSA encryption and AES algorithm.

For the purpose of preventing revealing AES key *K*, the student first encrypts it through RSA encryption with the public key *pk*. We denote the secret AES key as $\text{Enc}_{\text{pk}}(\text{K})$. Then the student sends the $\text{Enc}_{\text{pk}}(\text{K})$ to the RMIT servers. After receiving, one of the servers who holds the student's profile decrypts the $\text{Enc}_{\text{pk}}(\text{K})$ through RSA decryption with the private key *sk*. Once the server has the AES key *K*, it performs AES encryption which takes the input student's profile, and outputs the encrypted profile $\text{Enc}_{\text{K}}(\text{Profile})$. Then the server replies the encrypted profile to the student. Afterward, student decrypts the encrypted profile with AES decryption. In the end, the student gets his profile securely.

In this task, we use the ***student ID*** as the student's profile, and we employ AES (CBC mode) and 1024-bit key RSA to protect the data. Both online crypto tools are provided.

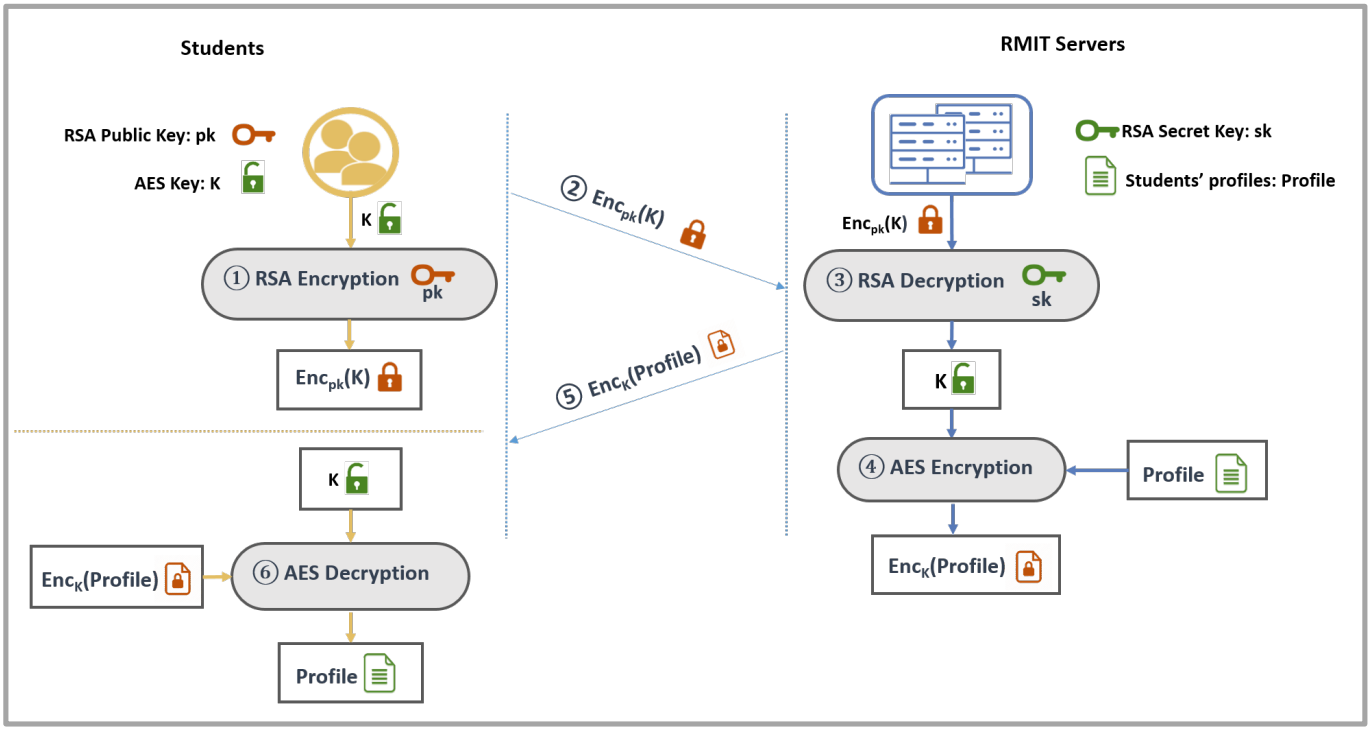

Figure 2. Process of getting student's profile

Question table:

Profile:	s3703131
AES key:	0123456712345678234567893456789a012345671234567823456789345678 9a
p (hex):	d090ce58a92c75233a6486cb0a9209bf3583b64f540c76f5294bb97d285eed33 aec220bde14b2417951178ac152ceab6da7090905b478195498b352048f15e7d
q (hex):	cab575dc652bb66df15a0359609d51d1db184750c00c6698b90ef3465c996551 03edbf0d54c56aec0ce3c4d22592338092a126a0cc49f65a4a30d222b411e58f
Public exponent e (hex):	10001
Private exponent d (hex):	8e9912f6d3645894e8d38cb58c0db81ff516cf4c7e5a14c7f1eddb1459d2cded 4d8d293fc97aee6aefb861859c8b6a3d1dfe710463e1f9ddc72048c09751971c 4a580aa51eb523357a3cc48d31cfad1d4a165066ed92d4748fb6571211da5cb1 4bc11b6e2df7c1a559e6d5ac1cd5c94703a22891464fba23d0d965086277a161
Encrypted AES key $Enc_{pk}(K)$:	6703422938b7710609a1f2673b7044ea279c55f9f7dafef5260d4e6aa1fbba52 bde6dcdeb4f6f63d360fffa907f335b80ebca1f809af3807d9bdfafbe39c25b1 0861ab33f03af8e2604db5c368d957c96ceacabc1b221dae6280a2bb7fb735fb 600c7b24dea88a03f3e784c632878537b28354aaa7110036720231adfdeae3fe
Encrypted profile Enc_K(Profile):	HlpI1hnM2DQ0CJrDKINHYQ==