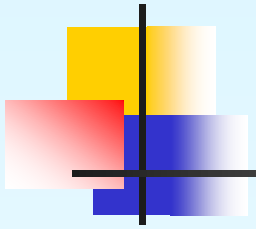


ECE 641 Term Project

Fei Long 21771618

Nan Wu 21771335

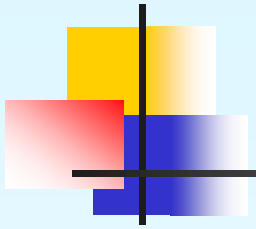
FPGA on Crptography using RSA Algorithm



(1) Introduction

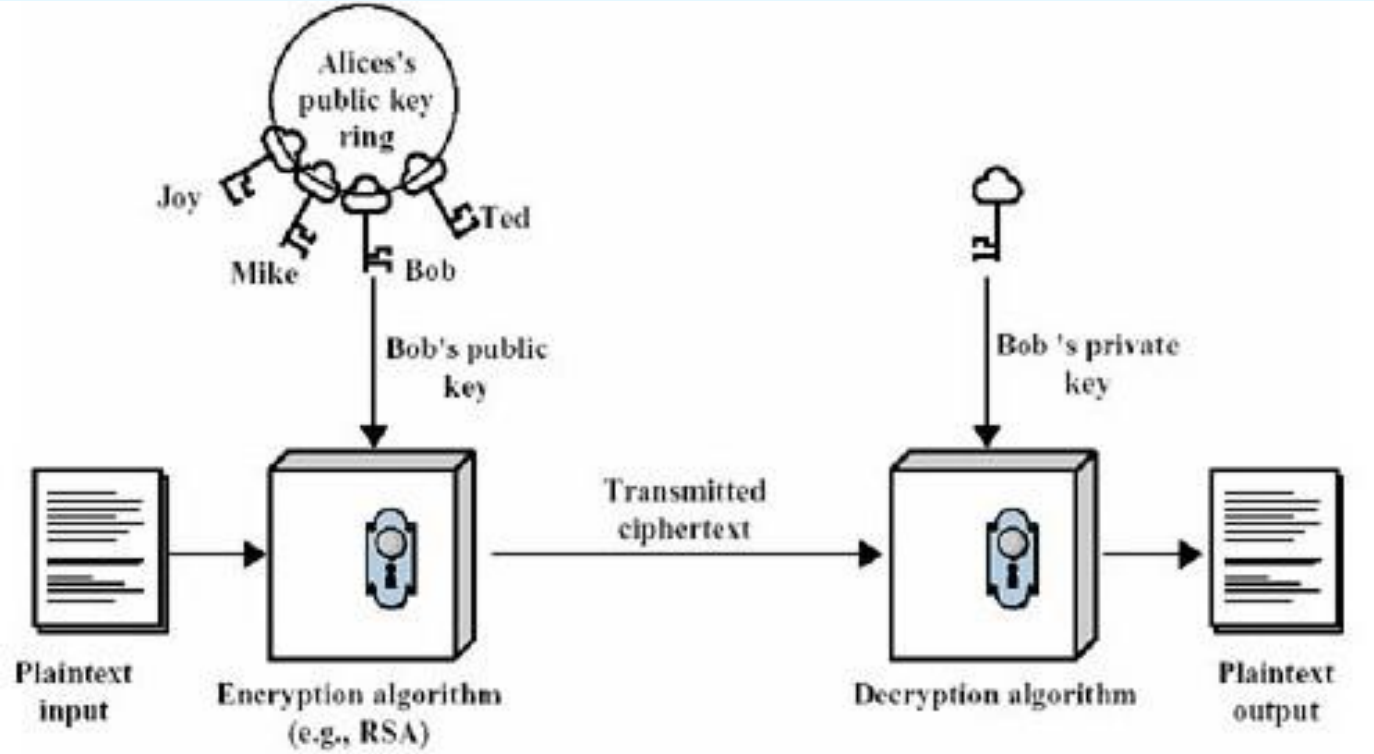
- RSA is an algorithm for public-private key cryptography
- RSA is widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys and the use of up-to-date implementations.
- The first algorithm known to be suitable for signing as well as encryption (to be discussed)
  - encryption key and decryption key are different

Public & Private Key Pair

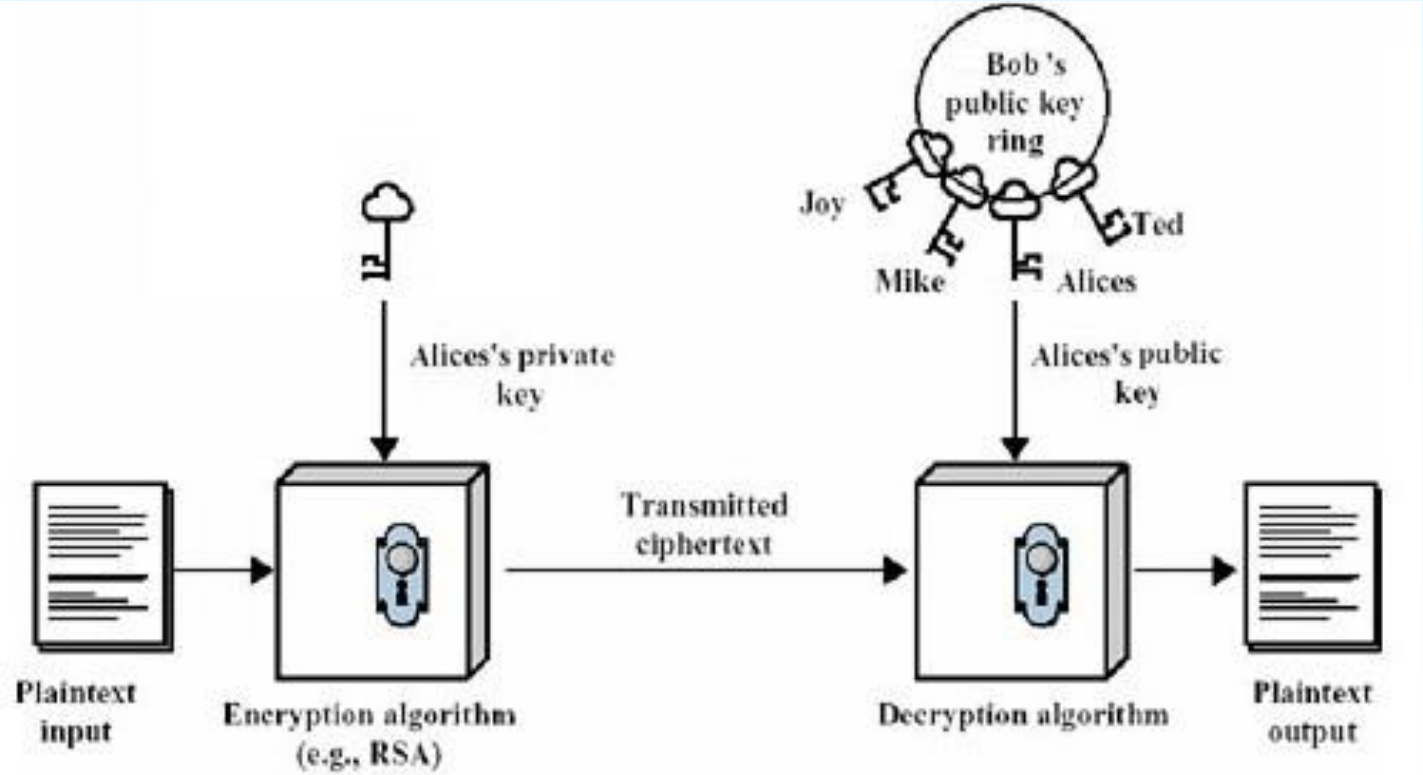


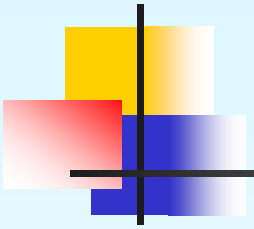
- Private key is kept only by the user
- The public key is published and can be known to everyone
- Messages encrypted with the public key can only be decrypted using the private key and vice versa
- Encryption and Digital Signature →

Encryption


Alices's public key ring
Joy
Ted
Mike
Bob
Bob's public key
Bob 's private key
Transmitted ciphertext
Plaintext input
Encryption algorithm (e.g., RSA)
Decryption algorithm
Plaintext output

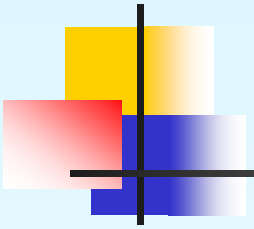
Digital Signature





Summary (1)

- To implement both encryption and digital signature using RSA –
- Plaintext is firstly encrypted using Bob's public key for encryption, and then encrypted using Alice's private key for signature
- At the receiver end, Bob uses Alice's public key to decrypt (verify the sender), then uses his own private key to decrypt (get the plaintext)

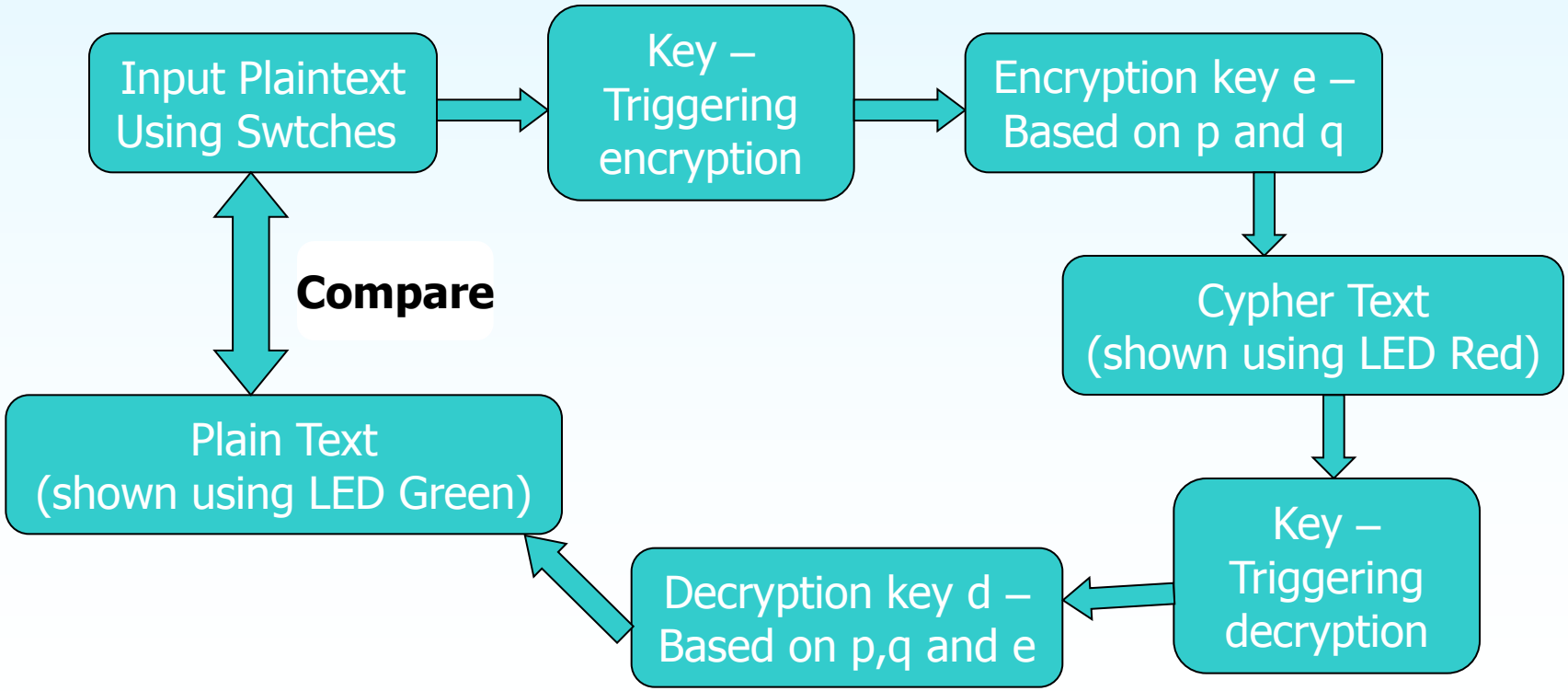


(2) Main Difficulty Faced

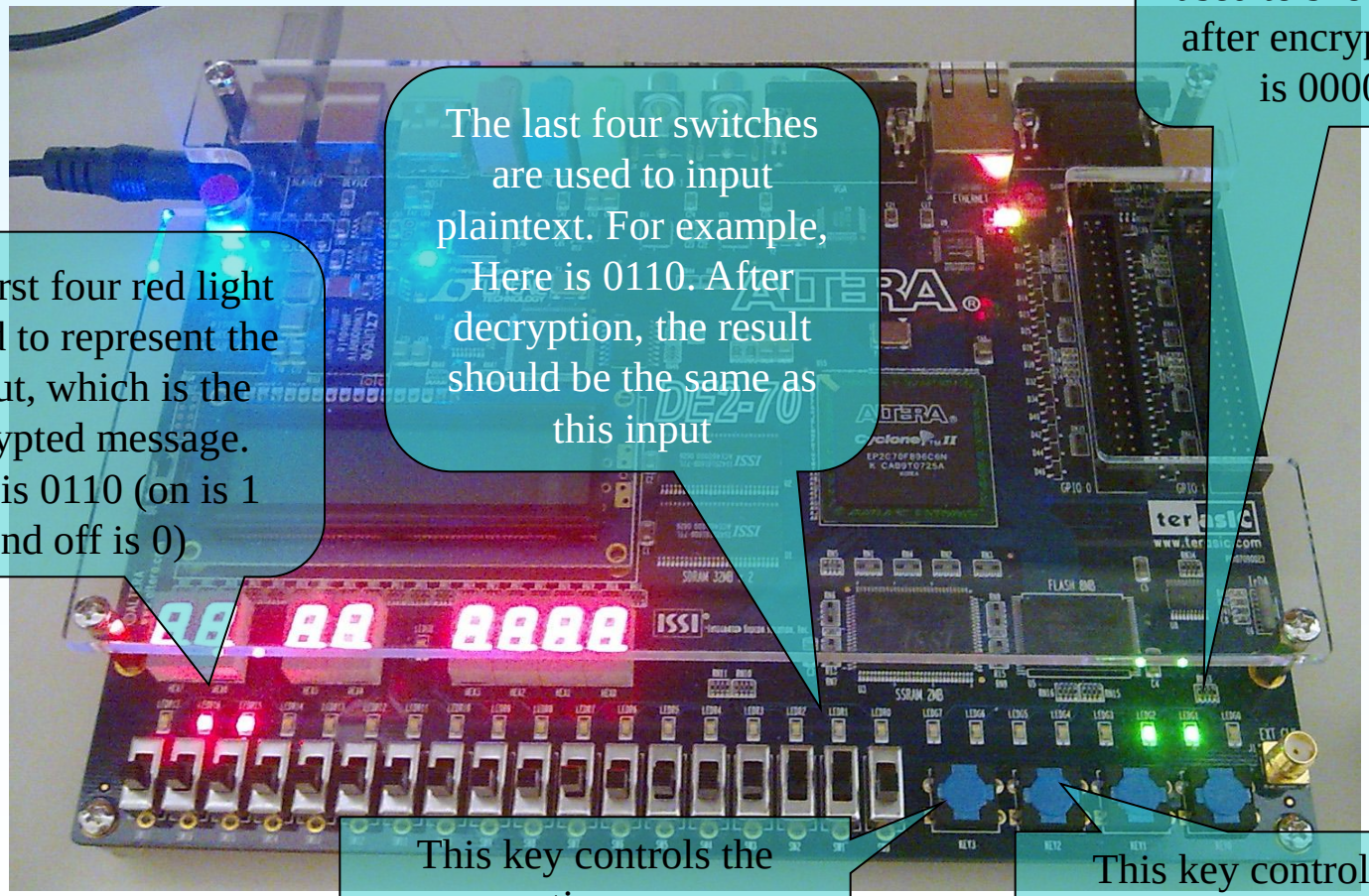
- We used ModelSim for simulation and debugging at the first stage since ModelSim is something more similar to Visual Studio
- For most of the cases, even if we succeeded when using ModelSim, there exist many errors when simulating using Quartus II
- ModelSim – easier to debug, compile and simulate
- Quartus II – more powerful, especially for burning
  - it seems that once the simulation is success, burning to board has very high prob. to be successful
  - not for ModelSim

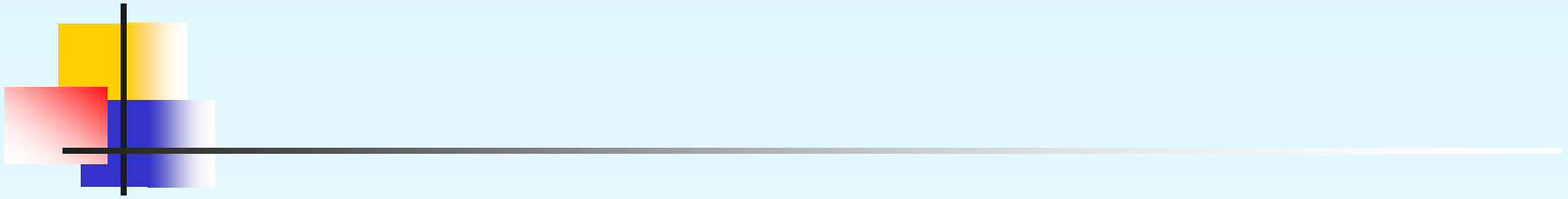
Demonstration

- Main flow chart for demo



Cont.


The first four red light is used to represent the output, which is the decrypted message. Here is 0110 (on is 1 and off is 0)
The last four switches are used to input plaintext. For example, Here is 0110. After decryption, the result should be the same as this input
Those green lights are used to show the result after encryption. Here is 00000110.
This key controls the encryption process
This key controls the decrption process



Thank you