Review of Quantum Cryptography

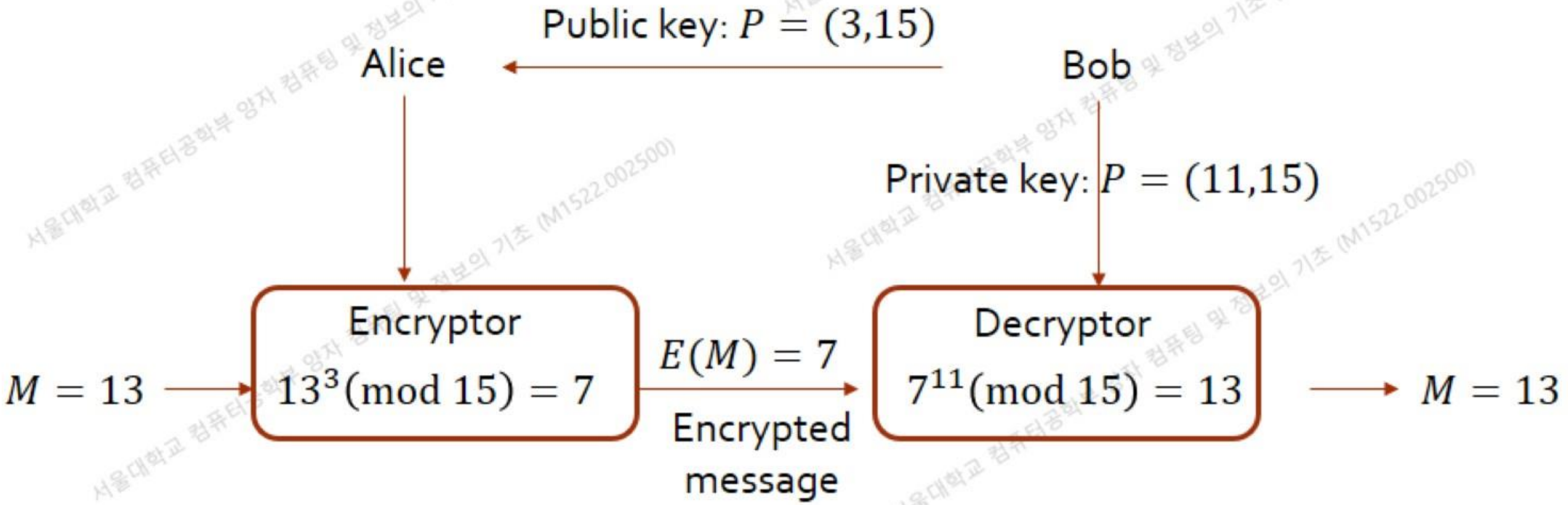
- Symmetric key system
  - Quantum key distribution system
- Public key system
  - Example: RSA (Rivest–Shamir–Adleman) public-key cryptosystem
  - The security of RSA is guaranteed by the difficulty of factoring a large number ➔ RSA factoring challenge
  - Also very useful for authentication

서울대학교 컴퓨터공학부 양자 컴퓨팅 및 정보의 기초 (M1522.002500)

Example of RSA encryption/decryption

- message to transmit: M
- Encryption: $E(M) = M^{e} (\mathrm{mod}\ n)$
- Decryption: $D\big(E\ (M)\big) = E(M)^{d} (\mathrm{mod}\ n)$

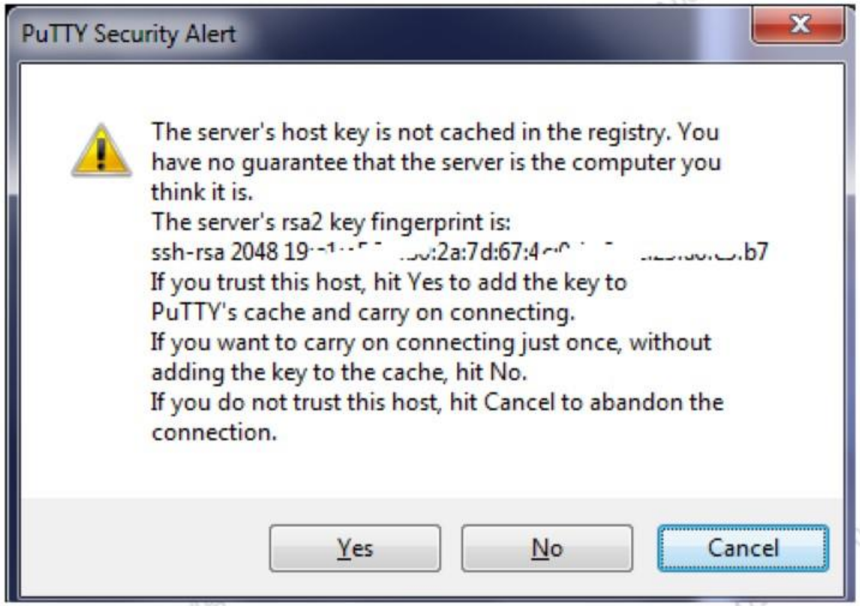
- Public key: $P = (e, n) = (3, 15)$
- Private key: $S = (d, n) = (11, 15)$
- Assume that the intended message M is 13.
- Encryption: $E(13) = 13^{3} (\mathrm{mod}\ 15) = 7$
- Decryption: $D\big(E\ (13)\big) = 7^{11} (\mathrm{mod}\ 15) = 13$


Public key: $P = (3{,}15)$
Alice
Bob
Private key: $P = (11{,}15)$
Encryptor
$13^{3} (\mathrm{mod}\ 15) = 7$
$E(M) = 7$
Encrypted message
Decryptor
$7^{11} (\mathrm{mod}\ 15) = 13$
$M = 13$
$M = 13$

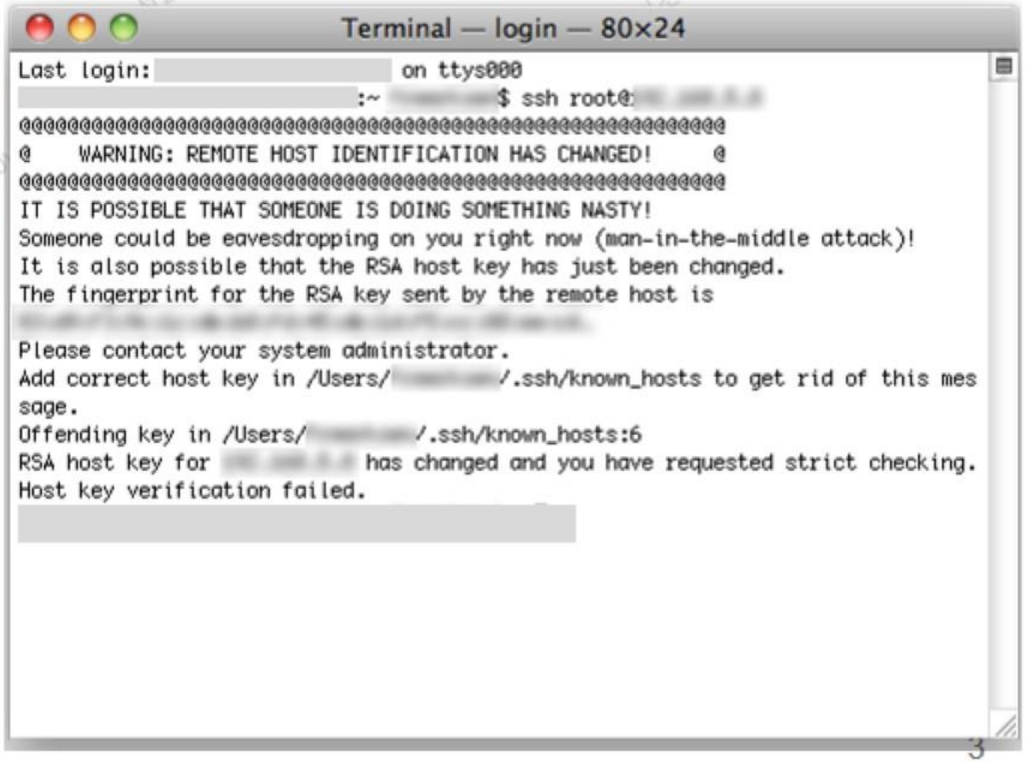
- In fact, $M^{e \cdot d} (\mathrm{mod}\ n) = M^{33} (\mathrm{mod}\ 15) = M$ for $0 \leq M < 15$

서울대학교 컴퓨터공학부 양자 컴퓨팅 및 정보의 기초 (M1522.002500)

SSH key fingerprint



The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's rsa2 key fingerprint is:
ssh-rsa 2048 19[redacted]:2a:7d:67:4[redacted].b7
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.



```
Last login: [redacted] on ttys000
[redacted]:~ [redacted]$ ssh root@[redacted]
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
[redacted]
Please contact your system administrator.
Add correct host key in /Users/[redacted]/.ssh/known_hosts to get rid of this mes
sage.
Offending key in /Users/[redacted]/.ssh/known_hosts:6
RSA host key for [redacted] has changed and you have requested strict checking.
Host key verification failed.
[redacted]
```

서울대학교 컴퓨터공학부 양자 컴퓨팅 및 정보의 기초 (M1522.002500)

Example of RSA key generation

- Generation of public/private keys for RSA
  - Select two large prime numbers, p and q.
  - Compute the product $n \equiv pq$.
  - Select at random a small odd integer, e, that is relatively prime to $\phi(n) = (p-1)(q-1)$.
  - Compute d, the multiplicative inverse of e, modulo $\phi(n)$.
  - The RSA public key is the pair $P = (e, n)$. The RSA private key is the pair $S = (d, n)$.

- $p = 3$ and $q = 5$.
- $n \equiv pq = 15$.
- $\phi(n) = (p-1)(q-1) = 8$
  $e = 3$
- $d = 11$ ➔ $e \cdot d = 33$
  ➔ $e \cdot d \left(\mathrm{mod}\ \phi(15)\right) =$
  $33\ (\mathrm{mod}\ 8) = 1$
- RSA public key: $P = (3{,}15)$
  RSA private key: $S = (11{,}15)$

서울대학교 컴퓨터공학부 양자 컴퓨팅 및 정보의 기초 (M1522.002500)

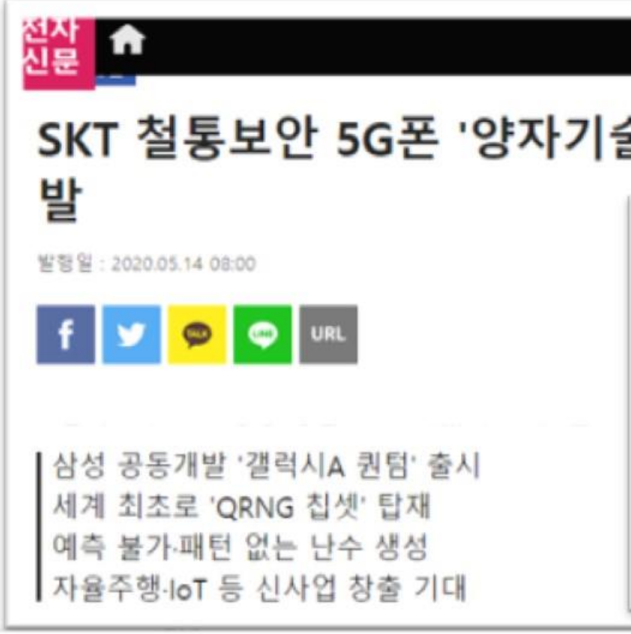
Summary of RSA

- Appendix 4.1, 4.2, 5
- Generation of public/private keys for RSA
  - Select two large prime numbers, p and q.
  - Compute the product $n \equiv pq$.
  - Select at random a small odd integer, e, that is relatively prime to $\phi(n) = (p-1)(q-1)$. $\phi(n)$ is called Euler('s totient) function.
  - Compute d, the multiplicative inverse of e, modulo $\phi(n)$.
  - The RSA public key is the pair $P = (e, n)$. The RSA private key is the pair $S = (d, n)$.
- Assume that the length of message M is $\text{floor}(\log_2 n)$ bits.
- Encryption: $E(M) = M^e (\text{mod } n)$
- Decryption:$D\big(E\ (M)\big) = E(M)^d (\text{mod } n)$
- Sketch of proof that the above procedure will recover M
  - Assume: M is relatively prime (or co-prime) to n.
  - $D\big(E\ (M)\big) = E(M)^d (\text{mod } n) = M^{ed}(mod\ n) = M^{1+k\phi(n)}(mod\ n)$
  - $= M \cdot M^{k\phi(n)}(mod\ n) = M(mod\ n)$
  - $M^{k\phi(n)}(mod\ n) = 1(mod\ n)$ ← Theorem A4.9

서울대학교 컴퓨터공학부 양자 컴퓨팅 및 정보의 기초 (M1522.002500)

QRNG

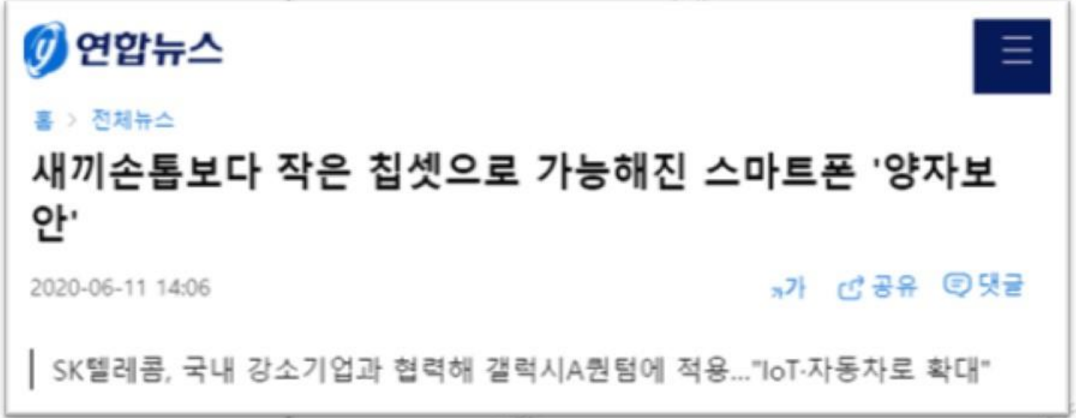
및 정보의 기초 (M1522.002500)



SKT 철통보안 5G폰 '양자기술 대중화' 첫 발

발행일 : 2020.05.14 08:00

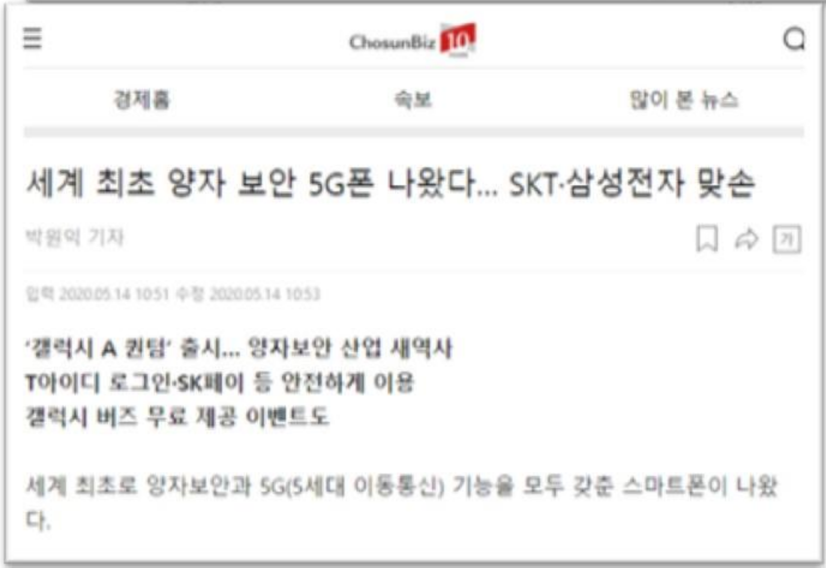
삼성 공동개발 '갤럭시A 퀀텀' 출시
세계 최초로 'QRNG 칩셋' 탑재
예측 불가·패턴 없는 난수 생성
자율주행·IoT 등 신사업 창출 기대



새끼손톱보다 작은 칩셋으로 가능해진 스마트폰 '양자보안'

2020-06-11 14:06

SK텔레콤, 국내 강소기업과 협력해 갤럭시A퀀텀에 적용..."IoT·자동차로 확대"



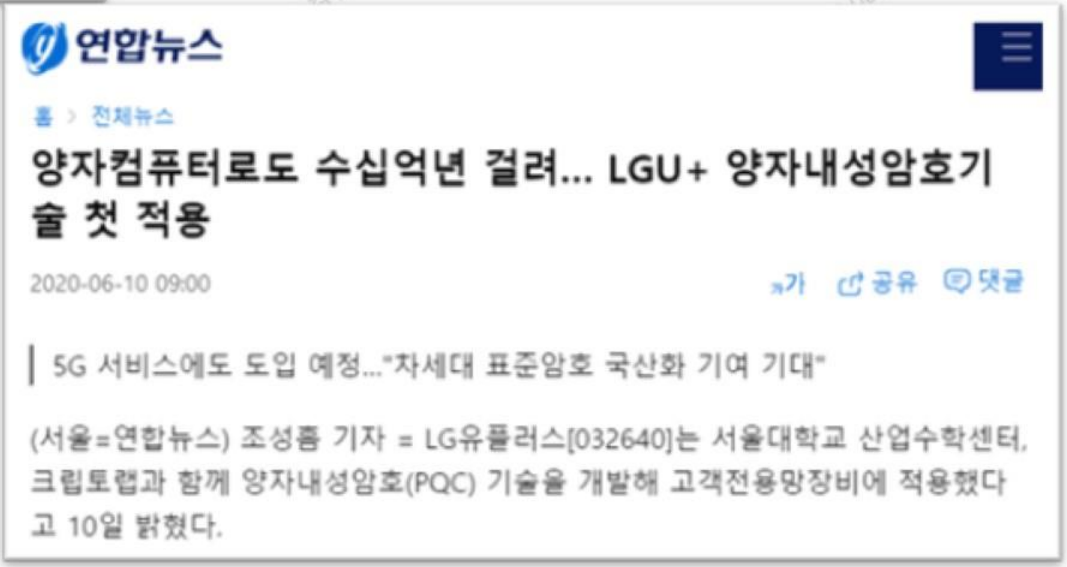
세계 최초 양자 보안 5G폰 나왔다... SKT·삼성전자 맞손

박원익 기자

입력 2020.05.14 10:51 수정 2020.05.14 10:53

'갤럭시 A 퀀텀' 출시... 양자보안 산업 새역사
T아이디 로그인·SK페이 등 안전하게 이용
갤럭시 버즈 무료 제공 이벤트도

세계 최초로 양자보안과 5G(5세대 이동통신) 기능을 모두 갖춘 스마트폰이 나왔다.



양자컴퓨터로도 수십억년 걸려... LGU+ 양자내성암호기술 첫 적용

2020-06-10 09:00

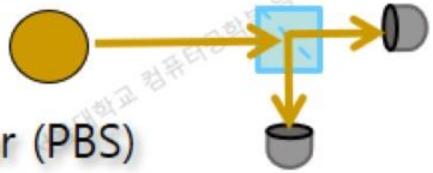
5G 서비스에도 도입 예정..."차세대 표준암호 국산화 기여 기대"

(서울=연합뉴스) 조성흠 기자 = LG유플러스[032640]는 서울대학교 산업수학센터, 크립토랩과 함께 양자내성암호(PQC) 기술을 개발해 고객전용망장비에 적용했다고 10일 밝혔다.

QRNG

- Quantum Random Number Generator (QRNG)
  - Basic concept: use the property of the quantum superposition to generate a true random number

$$|\psi\rangle = [|H\rangle - |H\rangle]/\sqrt{2}$$



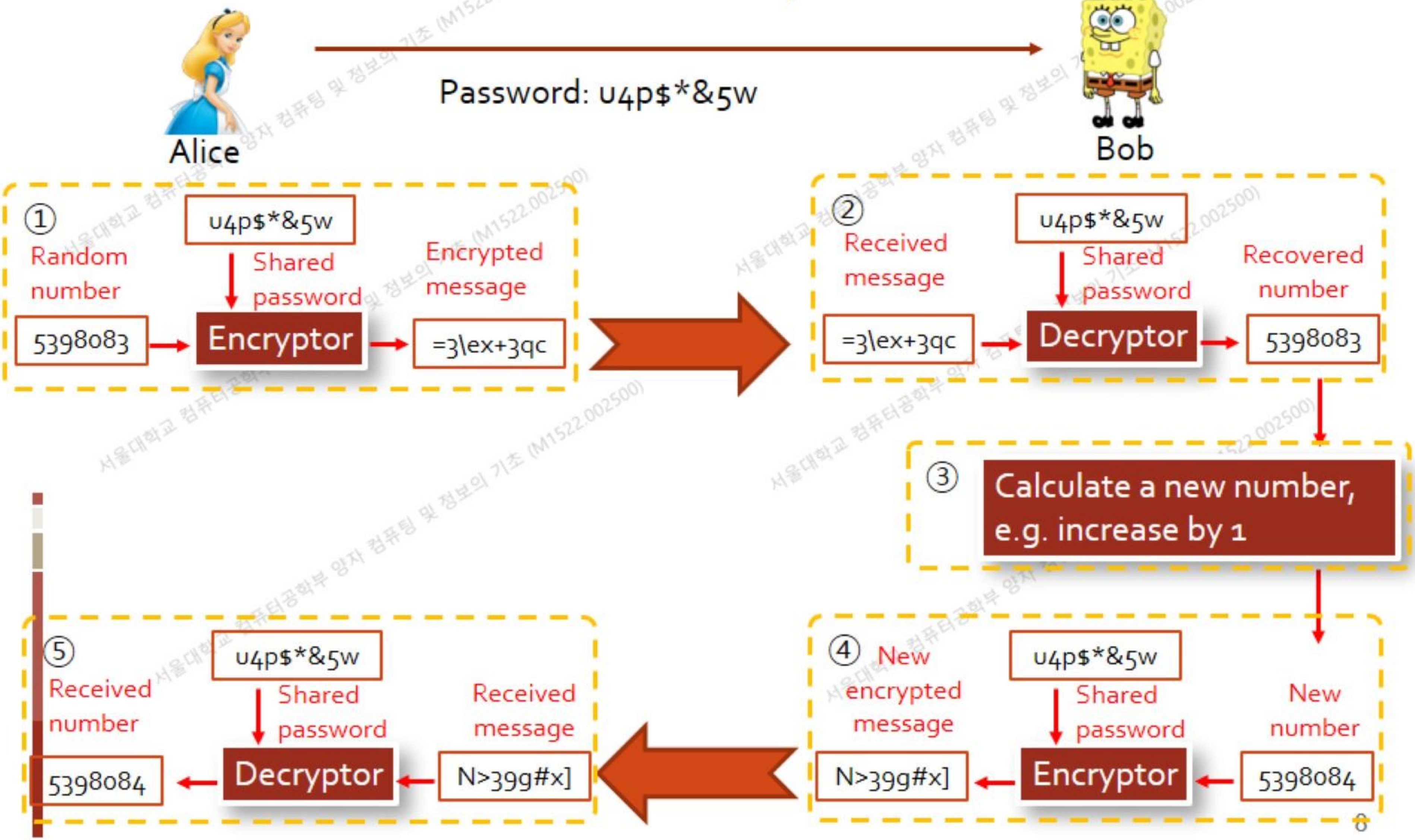
□ polarization beam splitter (PBS)

  - Is this equivalent to QKD?
    - No! It has nothing to do with QKD.
- Pseudo random number generator (PRNG)
  - Random number is generated by complex calculation, starting with random seed number
  - If the same seed number is used, we can always predict the same random number
  - Seed number is generally supplied by non-repeating process such as current time or movement of computer mouse
  - However these seed numbers are relatively easily guessed ➔ We need perfect random number generator

서울대학교 컴퓨터공학부 양자 컴퓨팅 및 정보의 기초 (M1522.002500)

How to prove who you are?

- Authentication: don't want to send password on the network



Alice

Password: u4p$*&5w

Bob

① Random number: 5398083 → Encryptor (Shared password: u4p$*&5w) → Encrypted message: =3\ex+3qc

② Received message: =3\ex+3qc → Decryptor (Shared password: u4p$*&5w) → Recovered number: 5398083

③ Calculate a new number, e.g. increase by 1

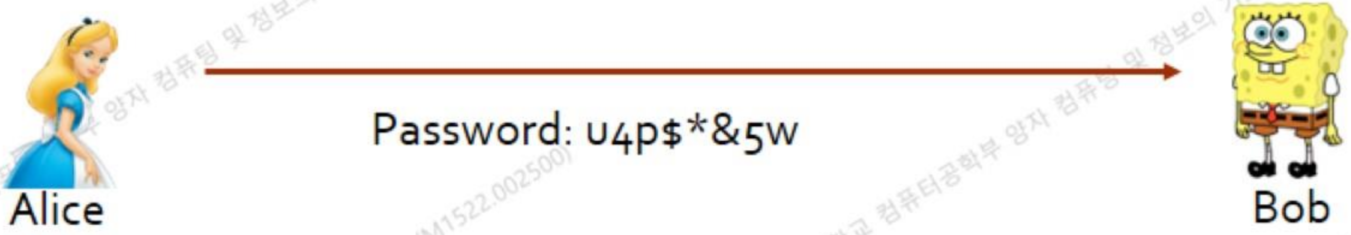
④ New number: 5398084 → Encryptor (Shared password: u4p$*&5w) → New encrypted message: N>39g#x]

⑤ Received message: N>39g#x] → Decryptor (Shared password: u4p$*&5w) → Received number: 5398084

서울대학교 컴퓨터공학부 양자 컴퓨팅 및 정보의 기초 (M1522.002500)

Example Protocol for QRNG

- Authentication: don't want to send password on the network

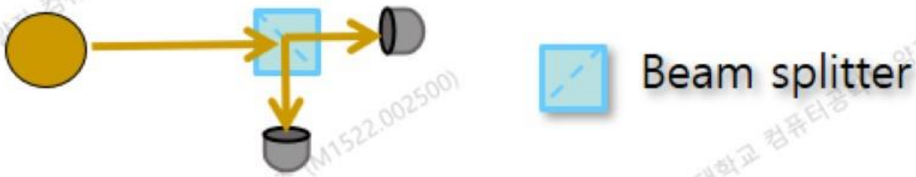


- Challenge-response
  1. Alice encrypts a random number with shared password and sends the encrypted message
  2. Bob decrypts the encrypted message using the mutually shared password
  3. Bob calculates new value based on the delivered random number
  4. Bob send back this new value encrypted again by the same password
  5. Alice decrypts the return message & verify Bob has the same password as well

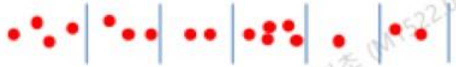
서울대학교 컴퓨터공학부 양자 컴퓨팅 및 정보의 기초 (M1522.002500)

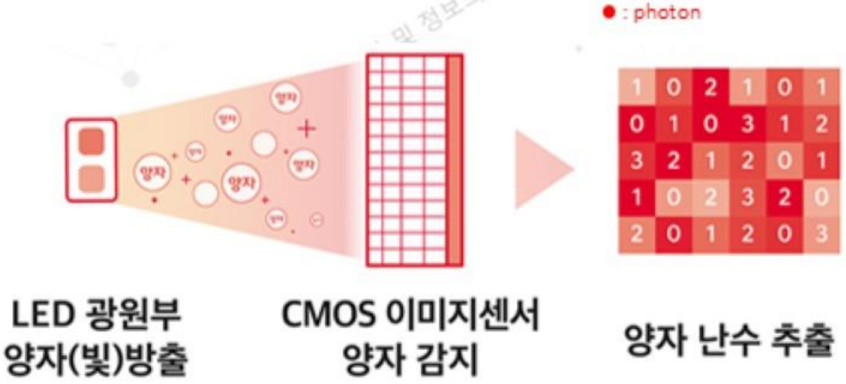
Different Approaches to Implement QRNG Chip

- What type of superposition can we use?
  - Superposition of paths



  - Superposition of photon numbers

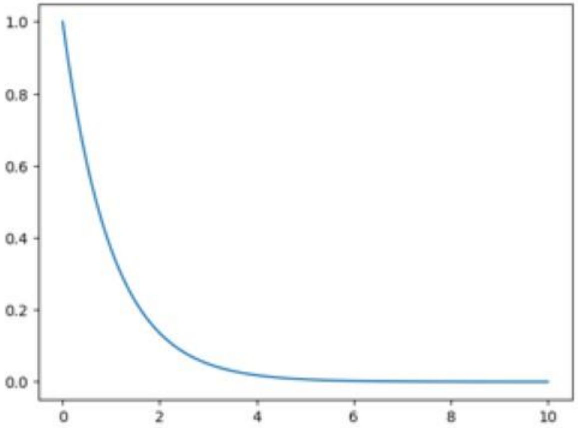




1	0	2	1	0	1
0	1	0	3	1	2
3	2	1	2	0	1
1	0	2	3	2	0
2	0	1	2	0	3

Image from https://www.sktinsight.com/123142

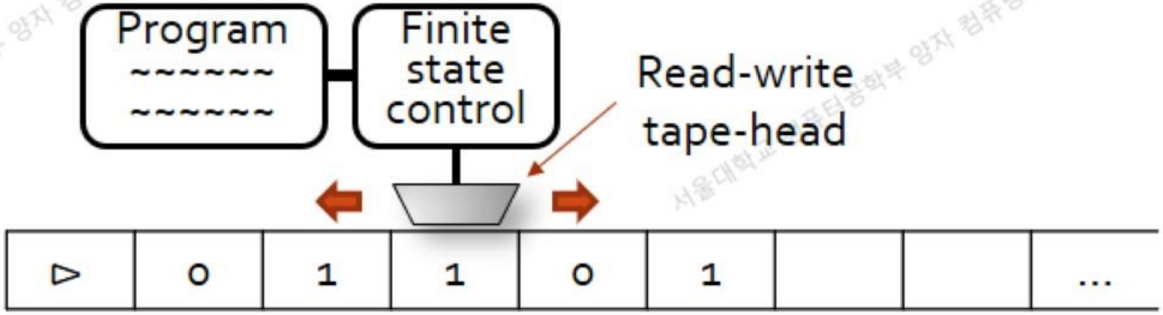
  - Superposition of times
    - Radioactive decay



서울대학교 컴퓨터공학부 양자 컴퓨팅 및 정보의 기초 (M1522.002500)

Models for computation

- Elements of Turing machine (section 3.1.1)
  - Program
  - Finite state control
  - Tape (like a computer memory)
  - Read-write tape-head
- Finite state control consists of a finite set of internal states, $q_1, \ldots, q_m$. q_s is a starting state and q_h is a halting state.
- Each square in the tape contains one symbol drawn from some alphabet Γ, a finite set of distinct symbols. ▷ indicates the left edge of the tape, and b means blank.
- The read-write tape-head starts from ▷ and after reading the symbol written in current square, update the value and move the head according to the program.


Program
~~~~~~~
~~~~~~~
Finite state control
Read-write tape-head
▷
0
1
1
0
1
...

서울대학교 컴퓨터공학부 양자 컴퓨팅 및 정보의 기초 (M1522.002500)

Turing machine

- A program for a Turing machine is a finite ordered list of program lines of the form $\langle q, x, q', x', s\rangle$. q is a current state and x is a value in the current square. q' is the next state and x' is the value to be written in the square. s is the direction to move the head.
- Example program

  1: $\langle q_{\mathrm{s}}, \rhd, q_1, \rhd, +1\rangle$

  2: $\langle q_1, 0, q_1, \mathrm{b}, +1\rangle$

  3: $\langle q_1, 1, q_1, \mathrm{b}, +1\rangle$

  4: $\langle q_1, \mathrm{b}, q_2, \mathrm{b}, -1\rangle$

  5: $\langle q_2, \mathrm{b}, q_2, \mathrm{b}, -1\rangle$

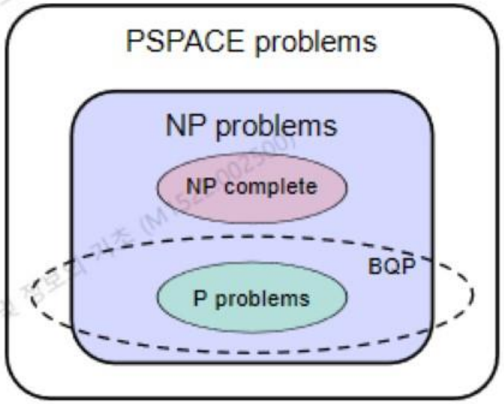
  6: $\langle q_2, \rhd, q_3, \rhd, +1\rangle$

  7: $\langle q_3, \mathrm{b}, q_{\mathrm{h}}, 1{,}0\rangle$

서울대학교 컴퓨터공학부 양자 컴퓨팅 및 정보의 기초 (M1522.002500)

Church-Turing thesis

- Church-Turing thesis
  - The class of functions computable by a Turing machine corresponds exactly to the class of functions which we would naturally regard as being computable by an algorithm
- Feasibility thesis
  - Complexity-theoretic (or extended) Church-Turing thesis
  - A probabilistic Turing machine can efficiently simulate any realistic model of computation, where "efficiently" means polynomial-time reduction
  - BPP (bounded-error probabilistic polynomial time) is the class of decision problems solvable by a probabilistic Turing machine in polynomial time with an error probability bounded away from 1/3 for all instances.
  - BQP (bounded-error quantum polynomial time) is the class of decision problems solvable by a quantum computer in polynomial time, with an error probability of at most 1/3 for all instances.
- Quantum complexity-theoretic Church-Turing thesis
  - A quantum Turing machine can efficiently simulate any realistic model of computation



The suspected relationship of BQP to other problem spaces (image from https://en.wikipedia.org/wiki/BQP)

서울대학교 컴퓨터공학부 양자 컴퓨팅 및 정보의 기초 (M1522.002500)