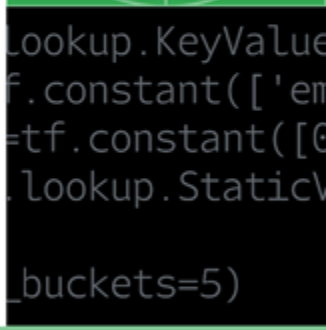
Google Developer Groups

Cryptography



건축학개론(2012)

RSA : Setting

p, q : Large primes

$n = p \times q$

RSA : Setting

p, q : Large primes

$n = p \times q$

e : encryption key

d : decryption key

$ed \equiv 1 \, mod \, (p-1)(q-1)$

RSA : Setting

p, q : Large primes

$n = p \times q$

public : n, e
private : p, q, d

e : encryption key

d : decryption key

$ed \equiv 1 \, mod \, (p-1)(q-1)$

e, p, q 가 정해지면 d는 확장 유클리드 알고리즘으로 구할 수 있다.

RSA : Encryption, Decryption

p, q : large primes, $n = p \times q$, $ed \equiv 1 \ mod \ \varphi(n)$

e : encryption key, d : decryption key

Encryption

Decryption

RSA : Encryption, Decryption

p, q : large primes, $n = p \times q$, $ed \equiv 1 \ mod \ \varphi(n)$

e : encryption key, d : decryption key

Encryption

$$C \equiv m^e \ mod \ n$$

Decryption

RSA : Encryption, Decryption

p, q : large primes, $n = p \times q$, $ed \equiv 1 \ mod \ \varphi(n)$

e : encryption key, d : decryption key

Encryption

$$C \equiv m^e \ mod \ n$$

Decryption

$$(m^e)^d \equiv m^{ed} \equiv m^{\varphi(n) \times k+1} \equiv m \ mod \ n$$

RSA : Encryption, Decryption

p, q : large primes, $n = p \times q$, $ed \equiv 1 \mod \varphi(n)$

e : encryption key, d : decryption key

Encryption

$$C \equiv m^e \mod n$$

Decryption

$$(m^e)^d \equiv m^{ed} \equiv \underline{m^{\varphi(n)\times k+1}} \equiv m \mod n$$

Euler's theorem

Why is RSA secure?

공개 정보 : $n(= p \times q),\ e$

비공개 정보 : $p,\ q,\ d$

Why is RSA secure?

공개 정보 : $n(= p \times q),\ e$

비공개 정보 : $p,\ q,\ d$

how to find d ?

1. p, q 를 각각 알아내기
2. $(p-1)(q-1)$ 알아내기

Why is RSA secure?

공개 정보 : $n(= p \times q),\ e$

비공개 정보 : $p,\ q,\ d$

how to find d ?

1. p, q 를 각각 알아내기
2. $(p-1)(q-1)$ 알아내기

소인수 분해는 NP ∩ co-NP 문제이다.

Complexity theory

P Problem

- the set of problems that can be solved in polynomial time by a **DTM**

NP Problem

- the set of problems that can be solved in polynomial time by a **NTM**

Complexity theory

P Problem

- the set of problems that can be solved in polynomial time by a **DTM**

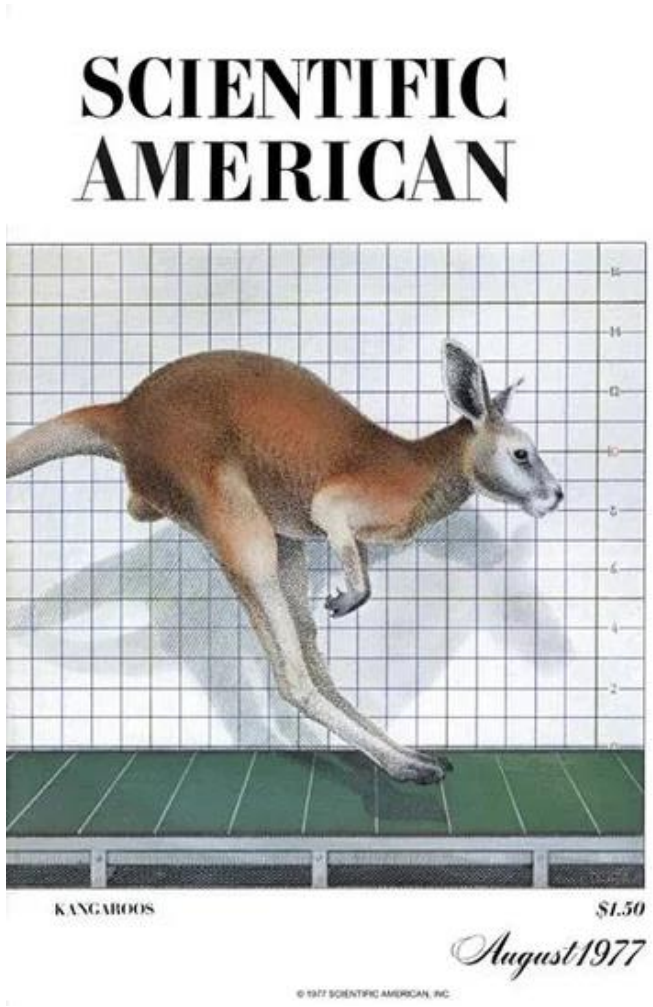
NP Problem

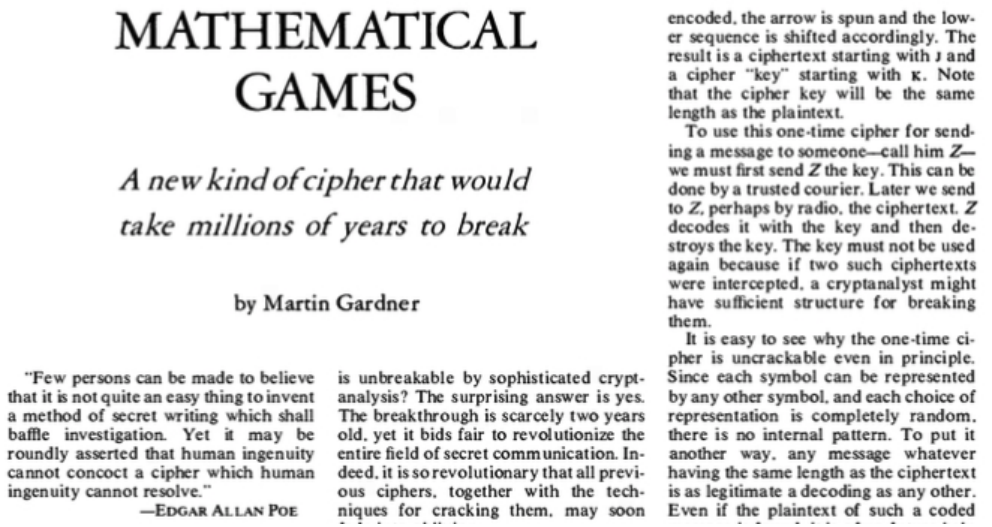
- the set of problems that can be solved in polynomial time by a **NTM**

Millennium Prize Problems

P-NP problem

Scientific American





MATHEMATICAL GAMES

A new kind of cipher that would take millions of years to break

by Martin Gardner

"Few persons can be made to believe that it is not quite an easy thing to invent a method of secret writing which shall baffle investigation. Yet it may be roundly asserted that human ingenuity cannot concoct a cipher which human ingenuity cannot resolve."
—EDGAR ALLAN POE

is unbreakable by sophisticated cryptanalysis? The surprising answer is yes. The breakthrough is scarcely two years old, yet it bids fair to revolutionize the entire field of secret communication. Indeed, it is so revolutionary that all previous ciphers, together with the techniques for cracking them, may soon

encoded, the arrow is spun and the lower sequence is shifted accordingly. The result is a ciphertext starting with J and a cipher "key" starting with K. Note that the cipher key will be the same length as the plaintext.

To use this one-time cipher for sending a message to someone—call him Z—we must first send Z the key. This can be done by a trusted courier. Later we send to Z, perhaps by radio, the ciphertext. Z decodes it with the key and then destroys the key. The key must not be used again because if two such ciphertexts were intercepted, a cryptanalyst might have sufficient structure for breaking them.

It is easy to see why the one-time cipher is uncrackable even in principle. Since each symbol can be represented by any other symbol, and each choice of representation is completely random, there is no internal pattern. To put it another way, any message whatever having the same length as the ciphertext is as legitimate a decoding as any other. Even if the plaintext of such a coded

n = 114,381,625,757,888,867,669,235,779,976,146 ⋯
e = 9,007
p is 64 bits prime, q is 65 bits prime

The challenge message posed in it was successfully decoded as early as April **1994**

Google Developer Groups

RSA Signatures

p, q : large primes, $n = p \times q$, $ed \equiv 1 \ mod \ \varphi(n)$

e : verifying key, d : signature key

Signature for Message m

$$s(m) \equiv m^d \ mod \, n$$

Verifying Signature

$$v = (s(m))^e \ mod \, n$$
$$v == m \ ?$$

RSA Signatures

p, q : large primes, $n = p \times q$, $ed \equiv 1 \ mod \ \varphi(n)$

e : verifying key, d : signature key

Signature for Message m

$$s(m) \equiv m^d \ mod \ n$$

Only the owner of key

Verifying Signature

$$v = (s(m))^e \ mod \ n$$
$$v == m \ ?$$

Anyone

Cryptographic Hashing

- Signature가 문서 사이즈에 의존하지 않고, 적절한 고정된 길이를 사용하도록 하자
- 같은 해시값을 생성하는 두 개의 입력값을 찾는 게 매우 어려워야 함

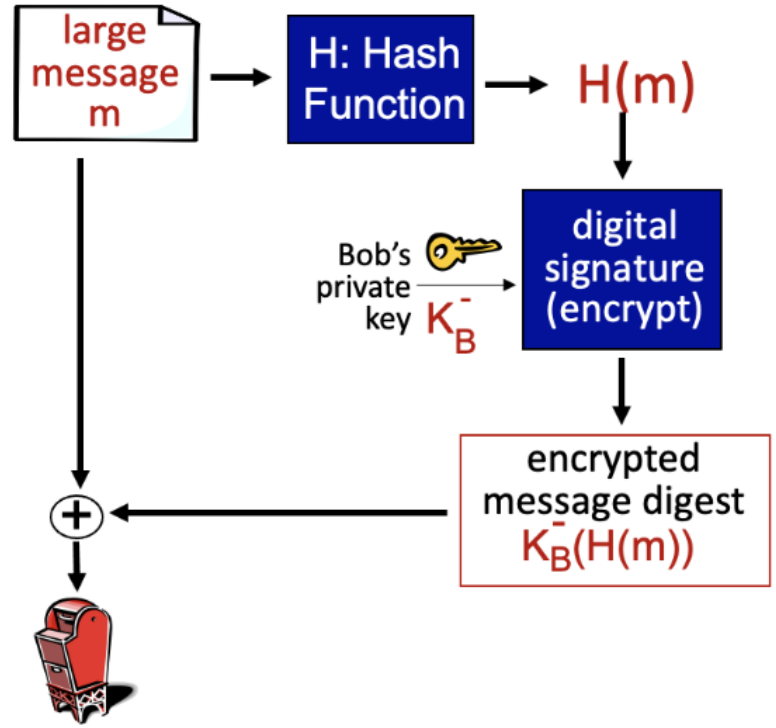
Cryptographic Hashing

- Signature가 문서 사이즈에 의존하지 않고, 적절한 고정된 길이를 사용하도록 하자
- 같은 해시값을 생성하는 두 개의 입력값을 찾는 게 매우 어려워야 함

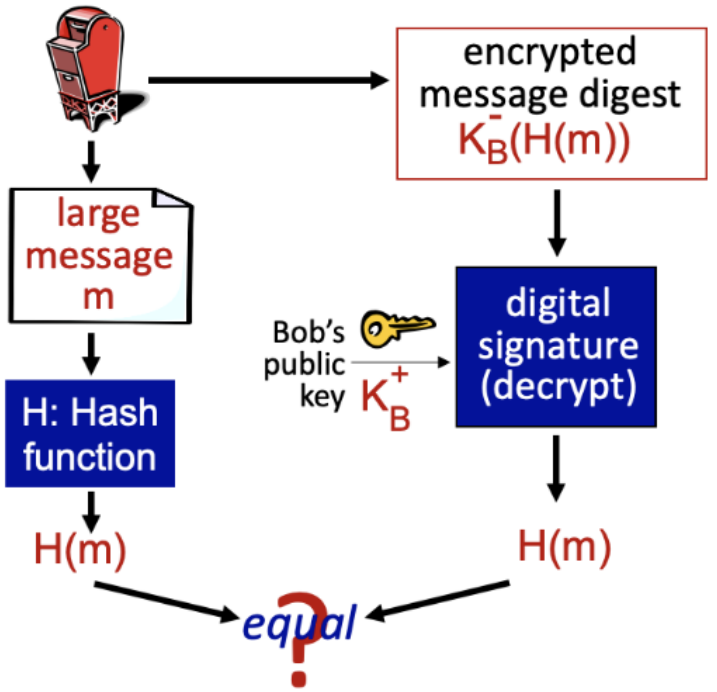
- MD5(Message-Digest algorithm 5, 128-bit)
- SHA-1 (Secure Hash Algorithm, 160-bit)
- SHA-2(224-bit, 256-bit, 384-bit, 512-bit)

Signatures

Bob sends digitally signed message:



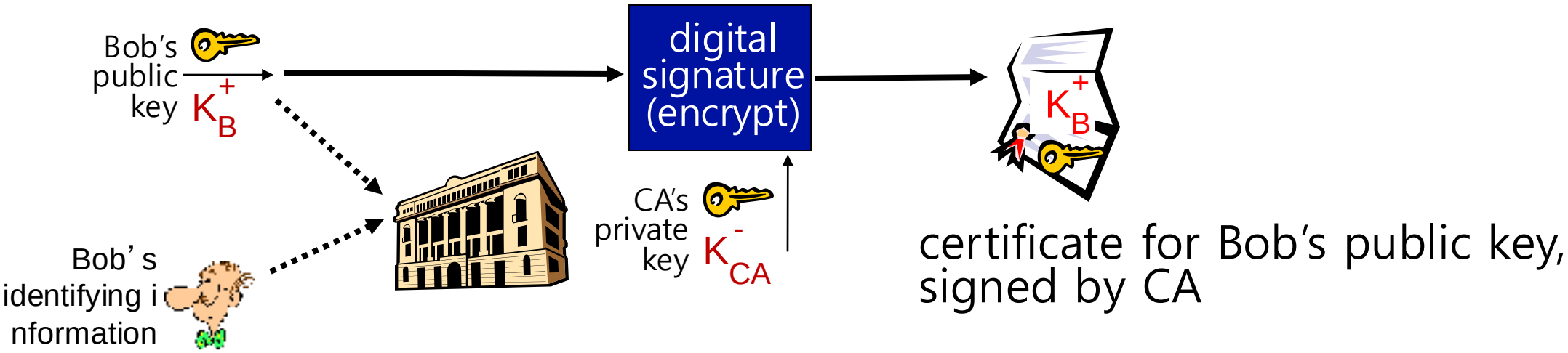
Alice verifies signature, integrity of digitally signed message:



문서 위변조 보장
작성자 보장

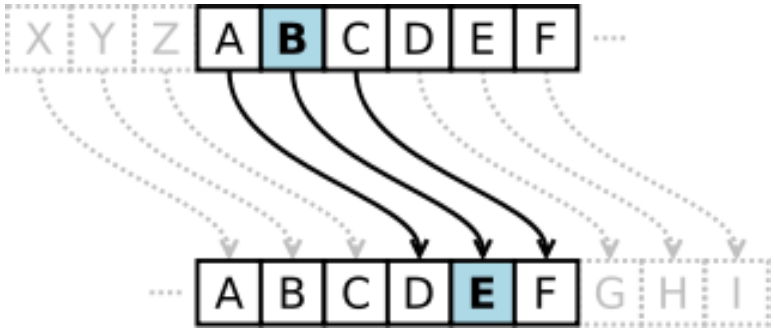
CA (Certificate Authority)

타인의 전자성명은 믿을 수 없다.

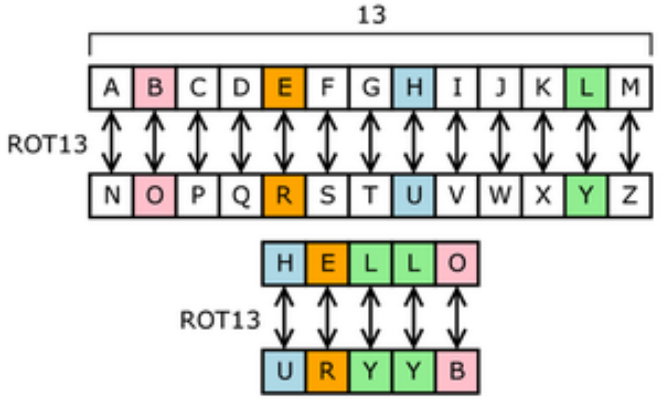


Symmetric key

Caesar cipher

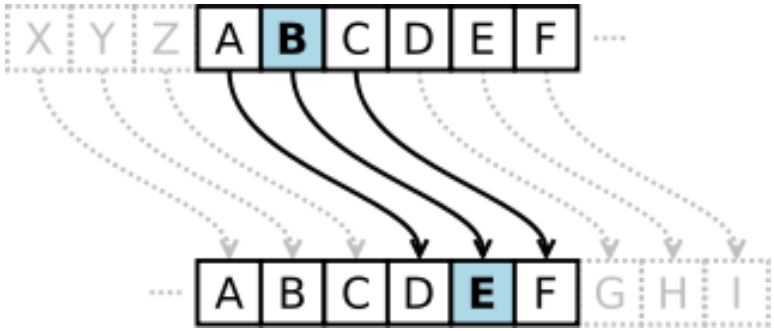


Monoalphabetic cipher

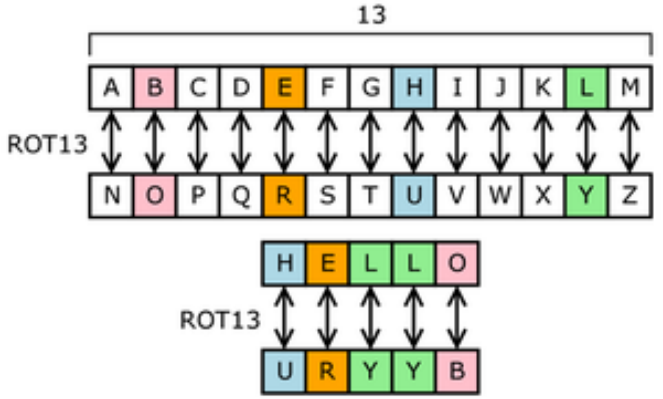


Symmetric key

Caesar cipher



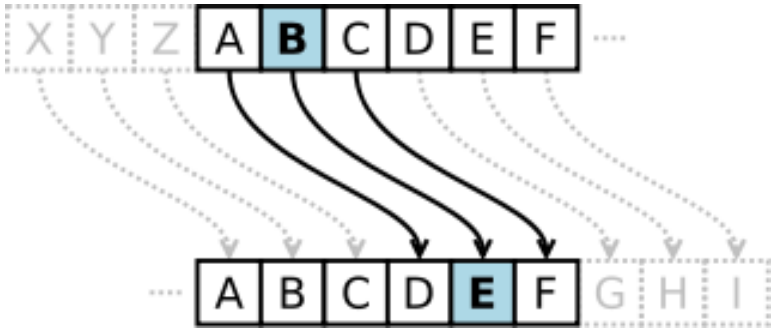
Monoalphabetic cipher



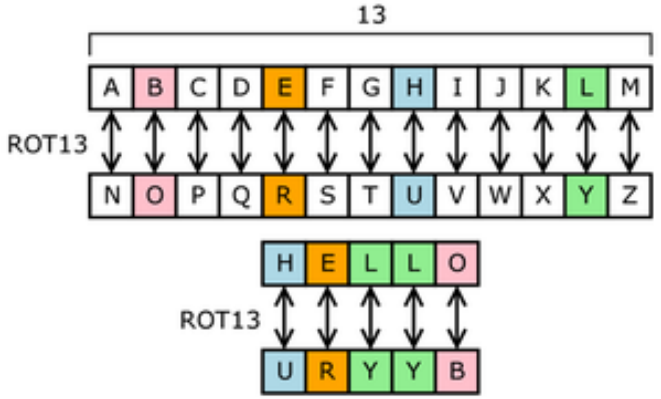
DES(Data Encryption Standard)
AES(Advanced Encryption Standard)

Symmetric key

Caesar cipher



Monoalphabetic cipher



DES(Data Encryption Standard)
AES(Advanced Encryption Standard)

56bit DES를 1초에 깰 수 있는 기계로 128bit AES를 깨려면 149조년이 걸림 [NIST]

DES는 RSA보다 소프트웨어로 구현하면 100배, 하드웨어로 구현하면 10,000배 빠름 [RSA Fast 2012]

Typical usage

Typical usage

public: n_A, e_A

Plaintext m

sender

A

private: d_A

Random k

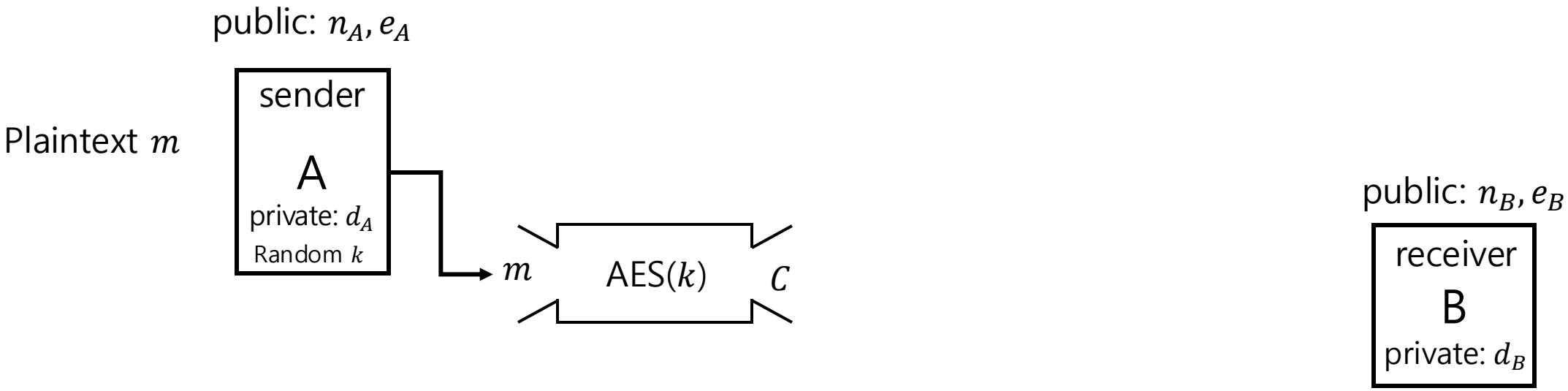
public: n_B, e_B

receiver

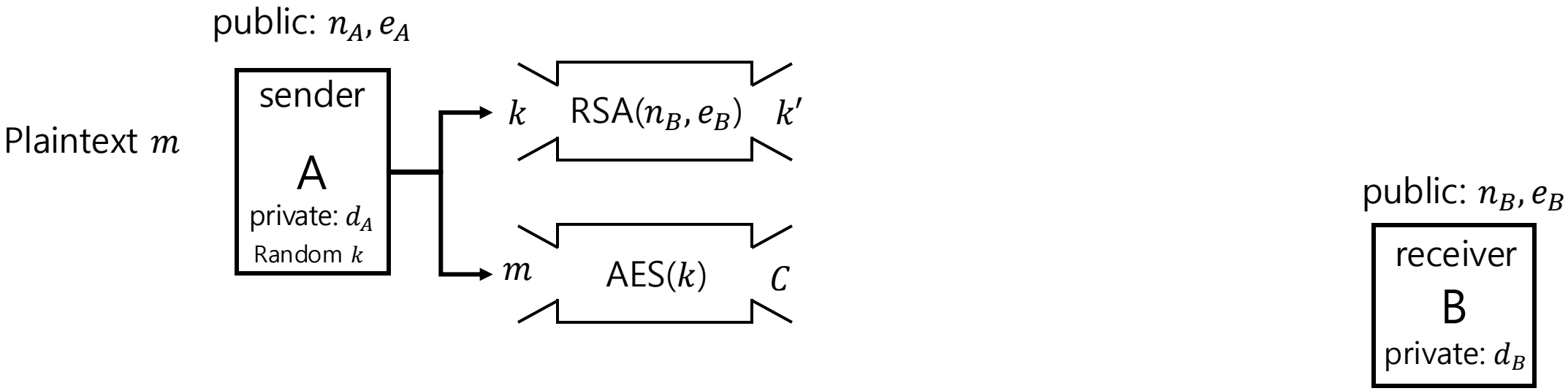
B

private: d_B

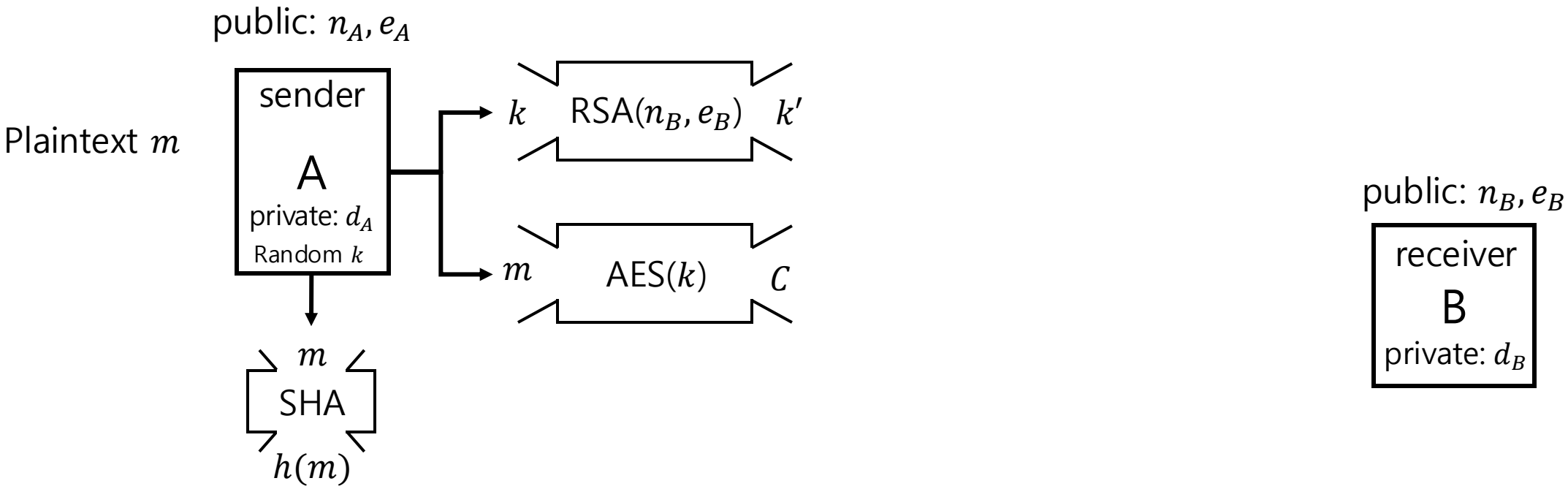
Typical usage



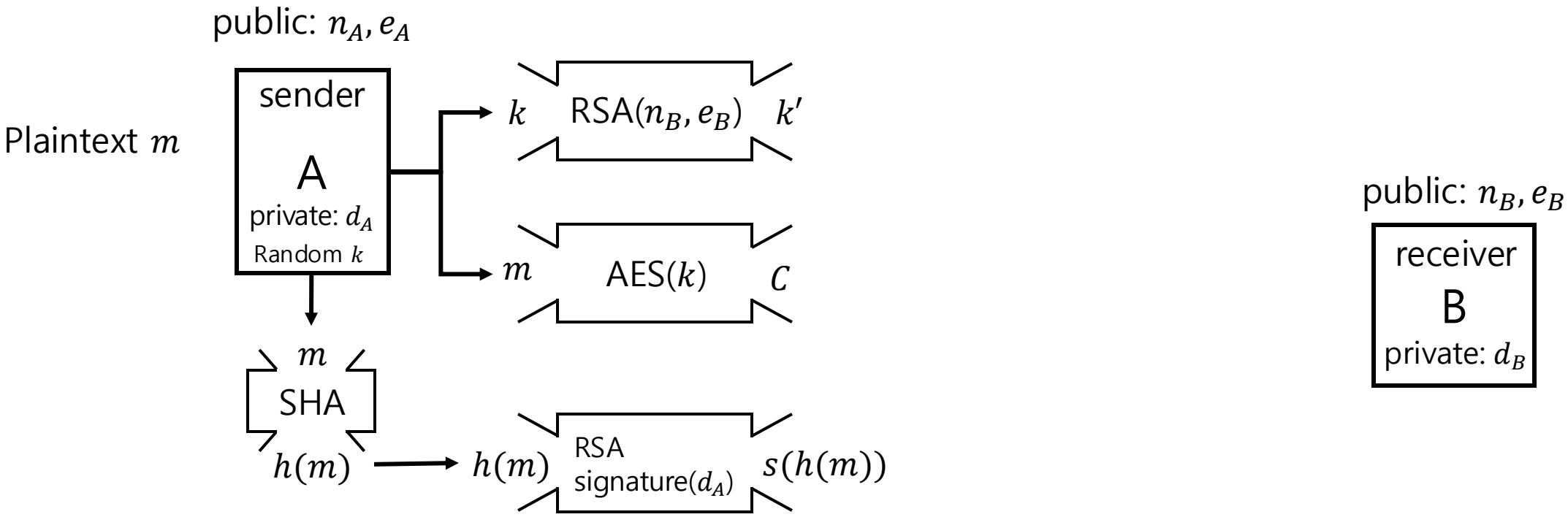
Typical usage



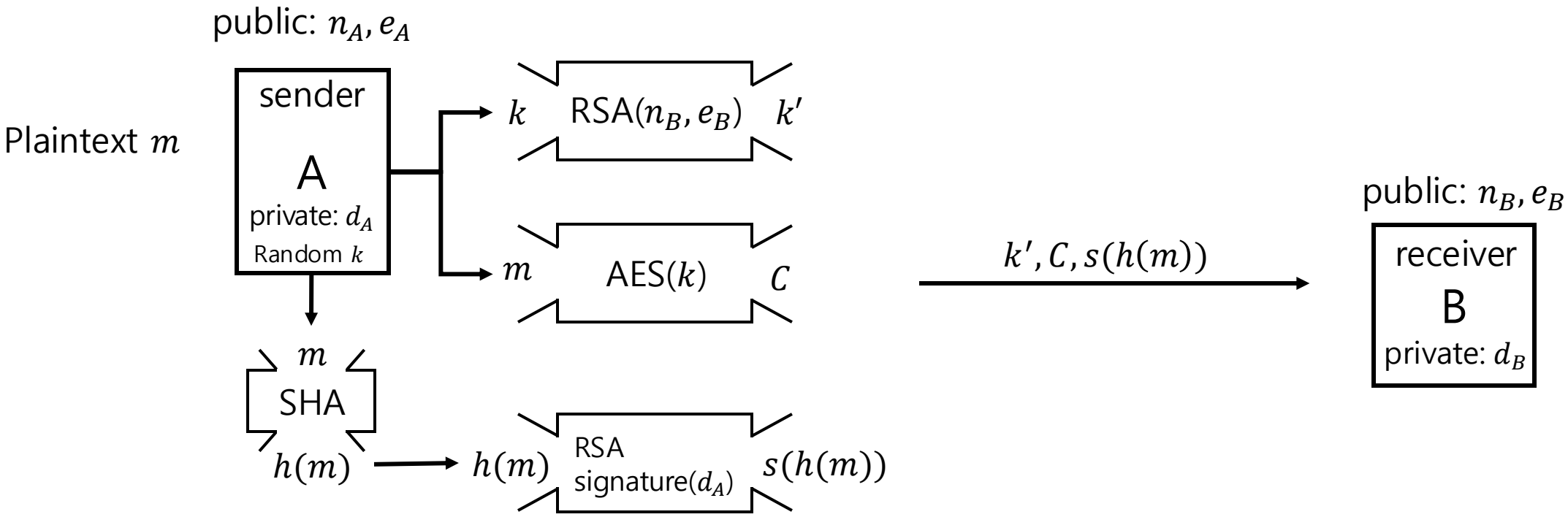
Typical usage



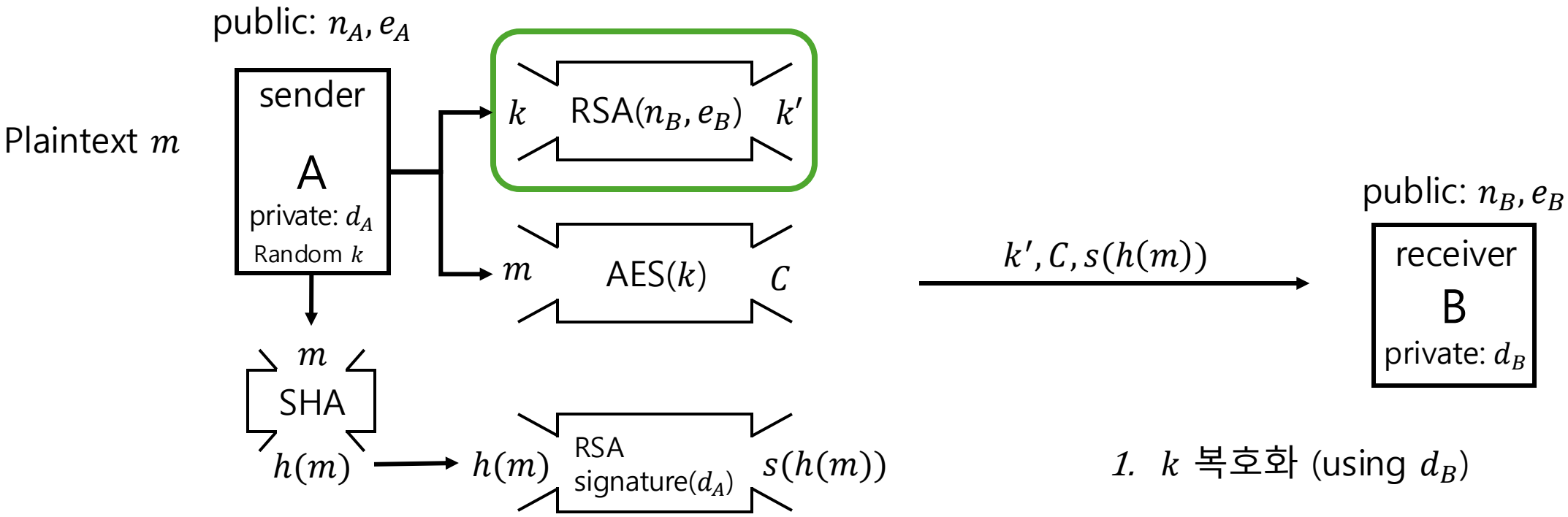
Typical usage



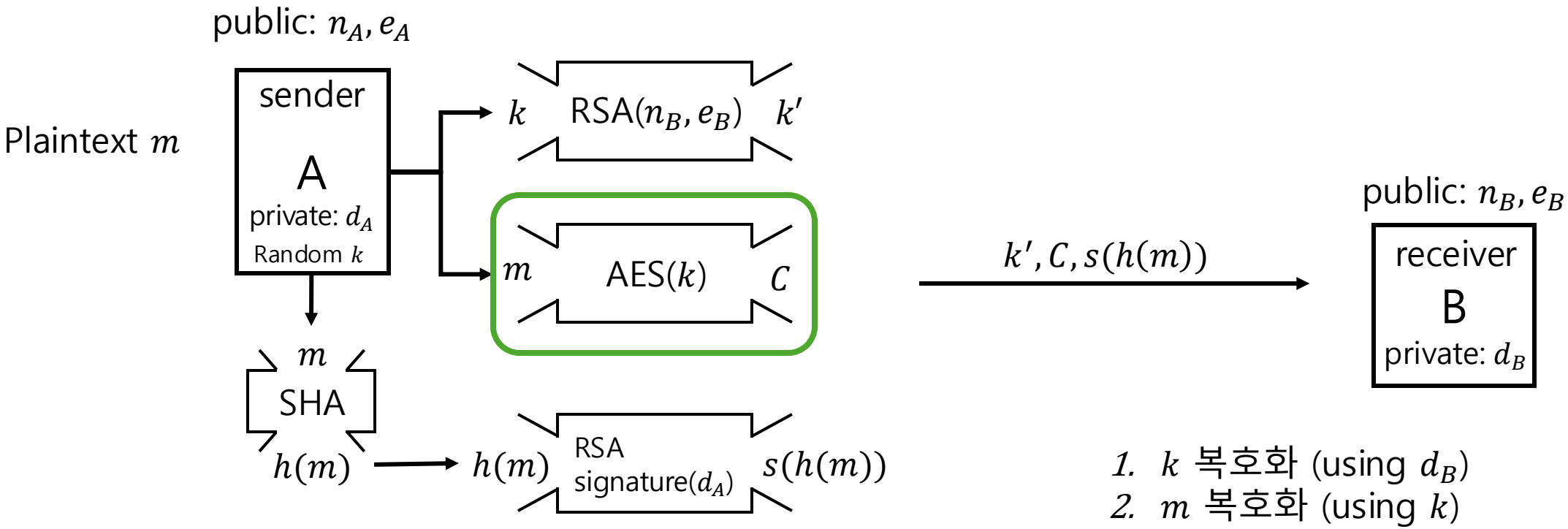
Typical usage



Typical usage

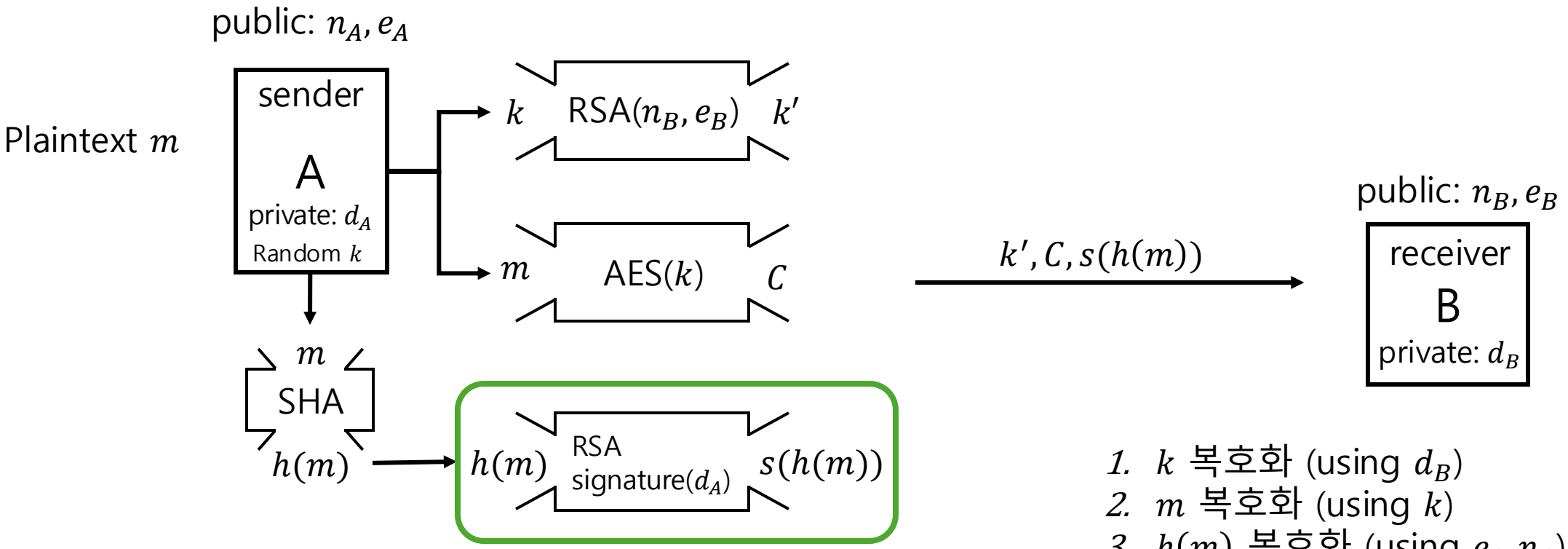


Typical usage



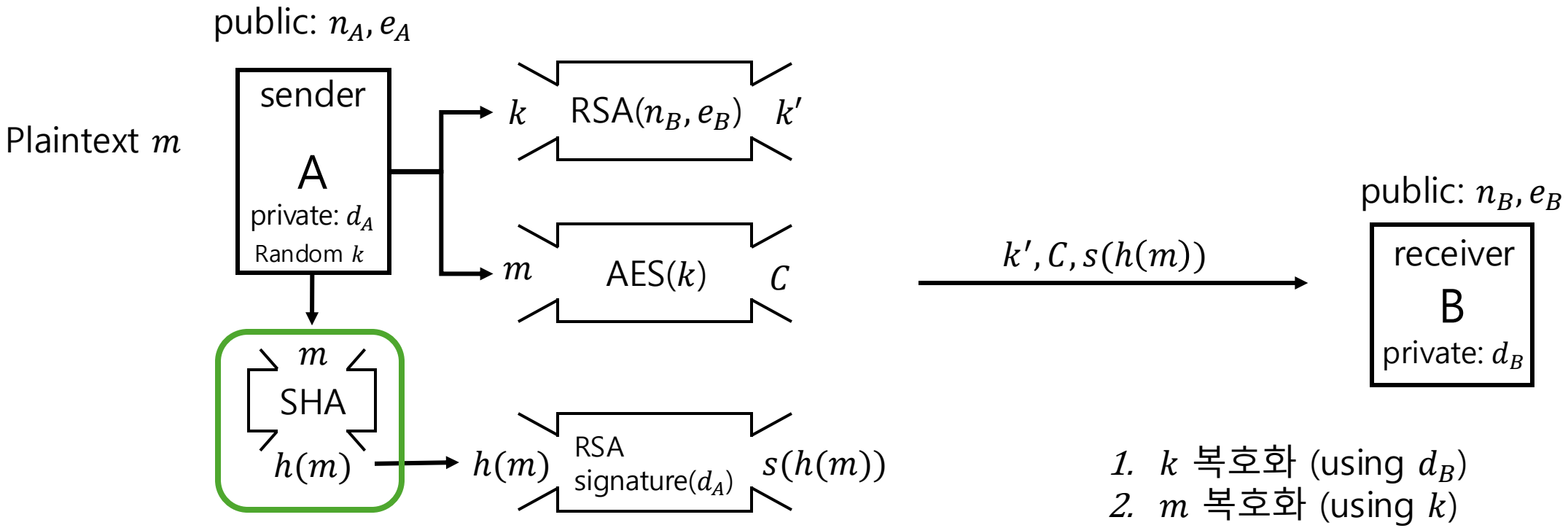
1. k 복호화 (using d_B)
2. m 복호화 (using k)

Typical usage



1. k 복호화 (using d_B)
2. m 복호화 (using k)
3. $h(m)$ 복호화 (using e_A, n_A)

Typical usage



1. k 복호화 (using d_B)
2. m 복호화 (using k)
3. $h(m)$ 복호화 (using e_A, n_A)
4. m을 SHA 적용 후 3번 값과 같은지 비교

Discrete Logarithm Problem

n, g, x가 주어질 때,
$g^k = x \mod n$을 만족하는 k를 찾아라

Discrete Logarithm Problem

n, g, x가 주어질 때,
$g^k = x \mod n$을 만족하는 k를 찾아라

DL problem is NP ∩ co-NP

Diffie-Hellman Key Exchange

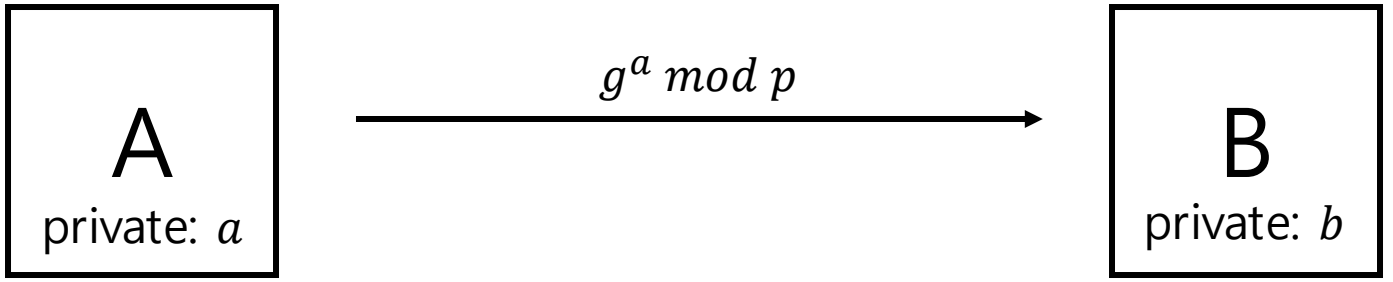
public: p, g

A
private: a

B
private: b

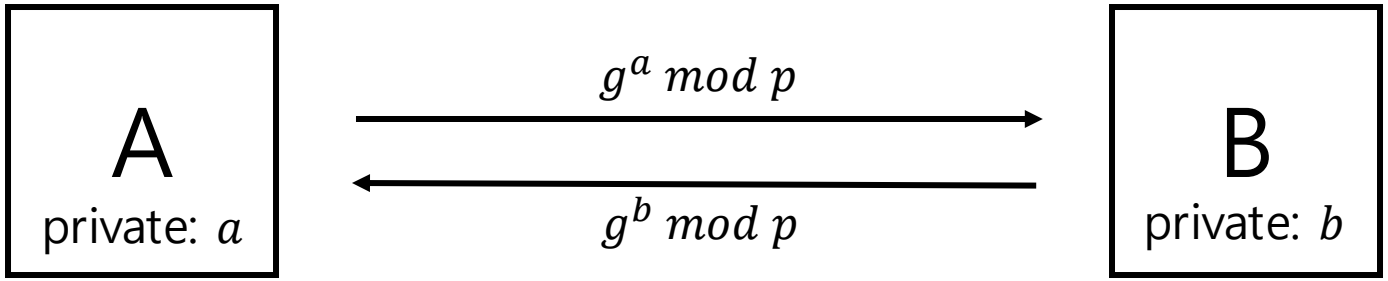
Diffie-Hellman Key Exchange

public: p, g



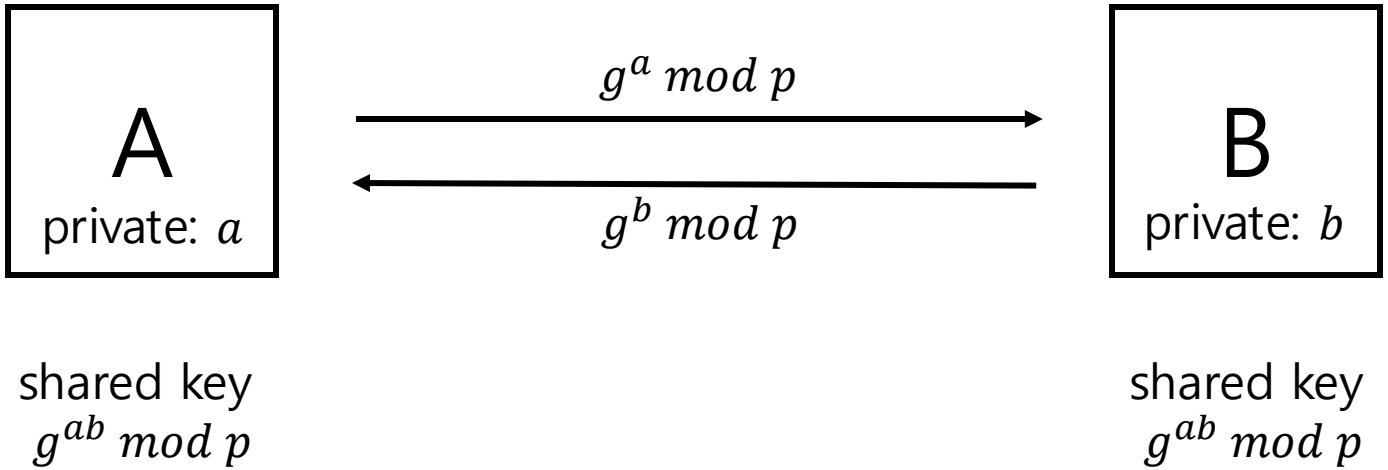
Diffie-Hellman Key Exchange

public: p, g



Diffie-Hellman Key Exchange

public: p, g



건축학개론(2012)