MGF(Masked Generation Function) (Input：Seed, dLen → Output：Mask)

$(0,0,0,0)$ $(0,0,0,1)$ ・・・・・ $(0,0,0,m)$

$\lfloor dLen/20 \rfloor$

transform 4 integers to 4 byte string following the ascii code

AsciiStr AsciiStr ・・・・・ AsciiStr

Seed change a hexadecimal to a character.

For example, “\0\0\0\0”

Seed Chr | 4 byte

16 byte

SHA1

Hash Hash ・・・・・ Hash

T_0 (20 [byte]) T_1 (20 [byte]) T_m (20 [byte])

mask $dLen$ [byte] from the high-order bit

Mask

Total : $20 \times (m+1)$ [byte]

Padding of RSA-OAEP Encode

Input	$m, mLen, Seed$
Output	Em

empty by default

Hash

hash value of empty

If Hash is SHA1,
hLen is 20 byte.

$mLen$

m

$k - mLen - 2hLen - 2$ [byte]
padding a hexadecimal “00”

DB = Hash() | $00 \cdots 00$ | 01 | m

1 [byte]

$hLen$

from $hLen$ to $k - mLen - 2hLen - 2$

Seed

MGF

MGF

from $k - mLen - 2hLen - 2$ to $hLen$

1 [byte]

00

Em = 00 | Masked Seed | Masked DB

$k = \lceil \text{bitlen}(pq)/8 \rceil$ [byte]

Reverse Padding of RSA-OAEP Decode

lost the high-order 1 byte when decoding

$k-1$ byte?

Em = 00 | Masked Seed | Masked DB

r needs for giving hLen byte,

Masked Seed

mask by BitAnd

mask the high-order hLen byte

Masked DB

from $k - mLen - 2hLen - 2$ to $hLen$

MGF

Input	$Em, mLen, r$
Output	m

from $hLen$ to $k - mLen - 2hLen - 2$

Seed

MGF

same hash value?

Hash() $\stackrel{?}{=}$ HashD

HashD | $00\cdots00$ | 01 | m

mask the low-order $mLen$ byte

m

RSA-PSSの署名生成のパディング処理について

入力	m
出力	Em