

Introduction to Cryptography

Prepared by Dr. Mohammad Alkhatib

Associate Professor

Major: Information Security



Lecture #6: Digital Signatures



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

- Introduction to Digital Signatures
- Security Services
- RSA Digital Signature Scheme
- RSA Digital Signature Example
- Attacks Against RSA Digital Signature

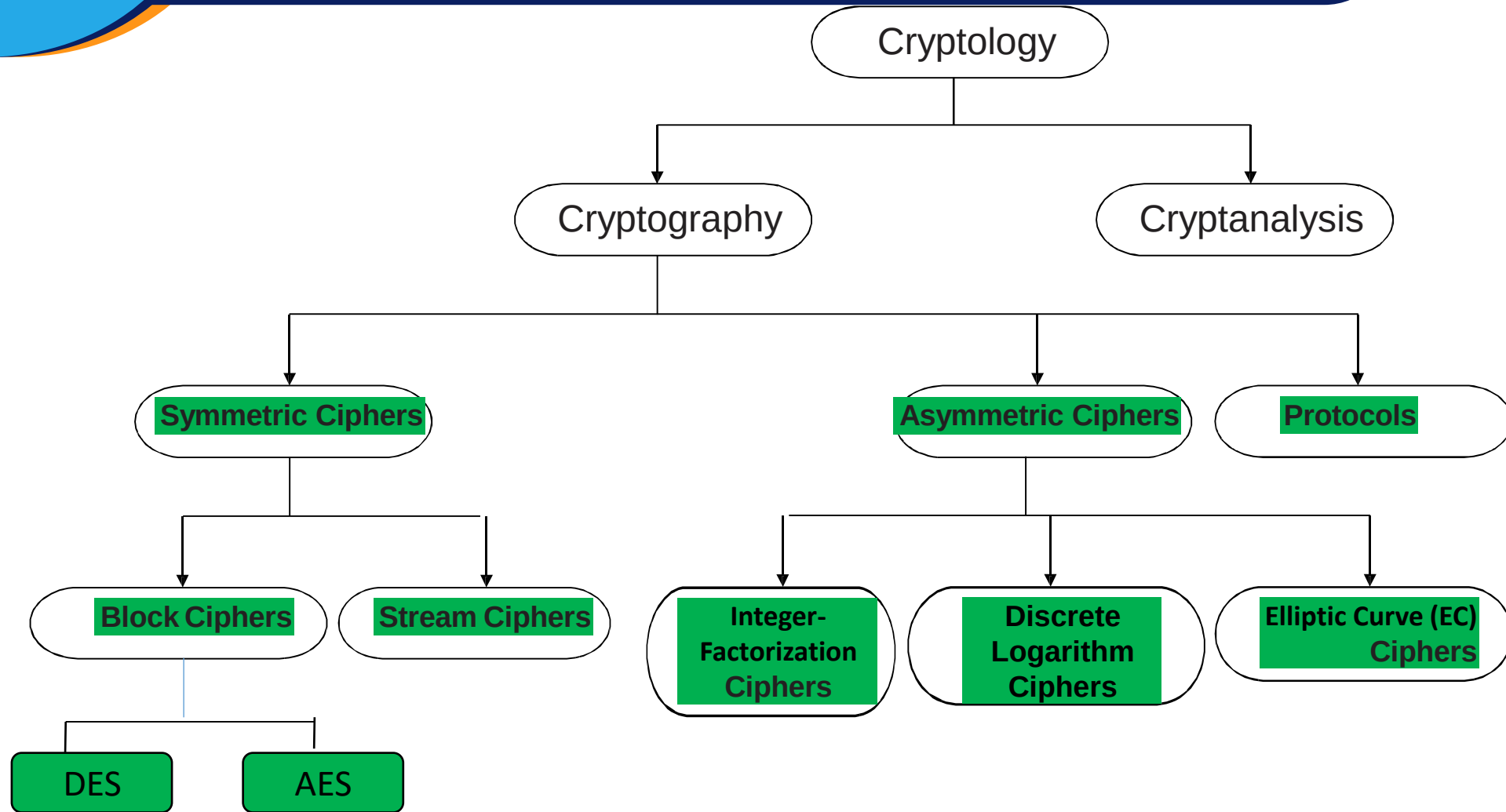
References:

[1] Katz and Lindell, Introduction to Modern Cryptography. SECOND EDITION. Chapman & Hall/ CRC Press, 2014

[2] C. Paar and J. Pelzl. Understanding Cryptography. A textbook for Student and Practitioners. Springer 2010

[3] Trappe, Wade, and Lawrence C. Washington. 2006 (2nd Edition). Introduction to cryptography: with coding theory. Upper Saddle River, NJ: Prentice Hall

Classification of the field of Cryptology

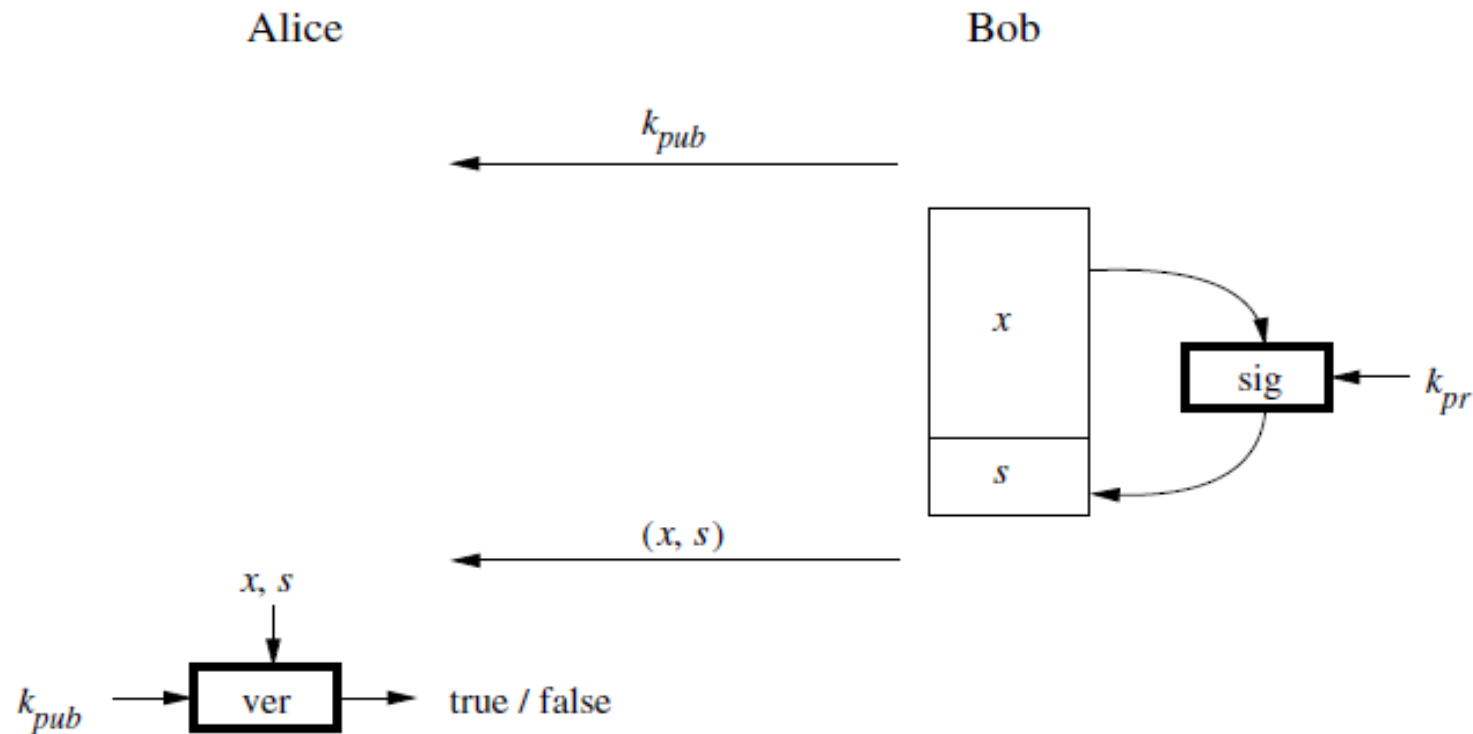


- ❑ The property of proving that a certain person generated a message is obviously also very important outside the digital domain.
- ❑ In the real, “analog” world, this is achieved by handwritten signatures on paper. For instance, if we sign a contract or sign a check, the receiver can prove to a judge that we actually signed the message.

- ❑ As with conventional handwritten signatures, only the person who creates a digital message must be capable of generating a valid signature.
- ❑ In order to achieve this with cryptographic primitives, we have to apply public-key cryptography.

Introduction to Digital Signature

- ❑ The basic idea is that the person who signs the message uses a private key, and the receiving party uses the matching public key.



- ❑ The process starts with Bob signing the message x
- ❑ The signature algorithm is a function of Bob's private key, k_{pr} .
- ❑ Hence, assuming he in fact keeps his private key private, only Bob can sign a message x *on his behalf*.
- ❑ In order to relate a signature to the message, x is also an input to the signature algorithm.

- ❑ After signing the message, the signature s is appended to the message x and the pair (x, s) is sent to Alice.
- ❑ Note that a digital signature without the message is the equivalent of a handwritten signature on a strip of paper without the contract or a check that is supposed to be signed.



- ❑ The digital signature itself is merely a (large) integer value, for instance, a string of 2048 bits.
- ❑ The signature is only useful to Alice if she has means to *verify whether* the signature is valid or not.
- ❑ For this purpose, a verification function is needed which takes both *x and the signature s as inputs. In order to link the signature to Bob, the function* also requires his public key.

- ❑ Even though the verification function has long inputs, its only output is the binary statement “true” or “false”.
- ❑ If x was *actually signed* with the private key that belongs to the public verification key, the output is true, otherwise it is false.

Basic Digital Signature Protocol

Alice

Bob

generate $k_{pr,B}, k_{pub,B}$

publish public key

sign message:

$$s = \text{sig}_{k_{pr}}(x)$$

send message + signature

$\xleftarrow{k_{pub,B}}$

$\xleftarrow{(x,s)}$

verify signature:

$$\text{ver}_{k_{pr,B}}(x, s) = \text{true/false}$$



➤ The core properties of digital signatures can be described as follows:

- ❑ A signed message can unambiguously be traced back to its originator since a valid signature can only be computed with the unique signer's private key.
- ❑ Only the signer has the ability to generate a signature on his behalf. Hence, we can prove that the signing party has actually generated the message.

➤ The core properties of digital signatures can be described as follows:

- ❑ Note that the basic protocol described in previous slide does not provide any confidentiality of the message since the message x is being sent over the communication network.
- ❑ Of course, the message can be kept confidential by also encrypting it, e.g., with AES or 3DES

❑ The most important security services that should be provided by security systems are:

1. **Confidentiality**: Information is kept secret from all but authorized parties.
2. **Integrity**: Messages have not been modified in transit.
3. **Message Authentication**: The sender of a message is authentic
4. **Nonrepudiation**: The sender of a message can not deny the creation of the message.

- ❑ **Confidentiality** can be achieved by primarily symmetric ciphers and less frequently asymmetric encryption.
- ❑ **Integrity and message authentication** are provided by digital signatures and message authentication codes which.
- ❑ **Nonrepudiation** can be achieved with digital signatures.

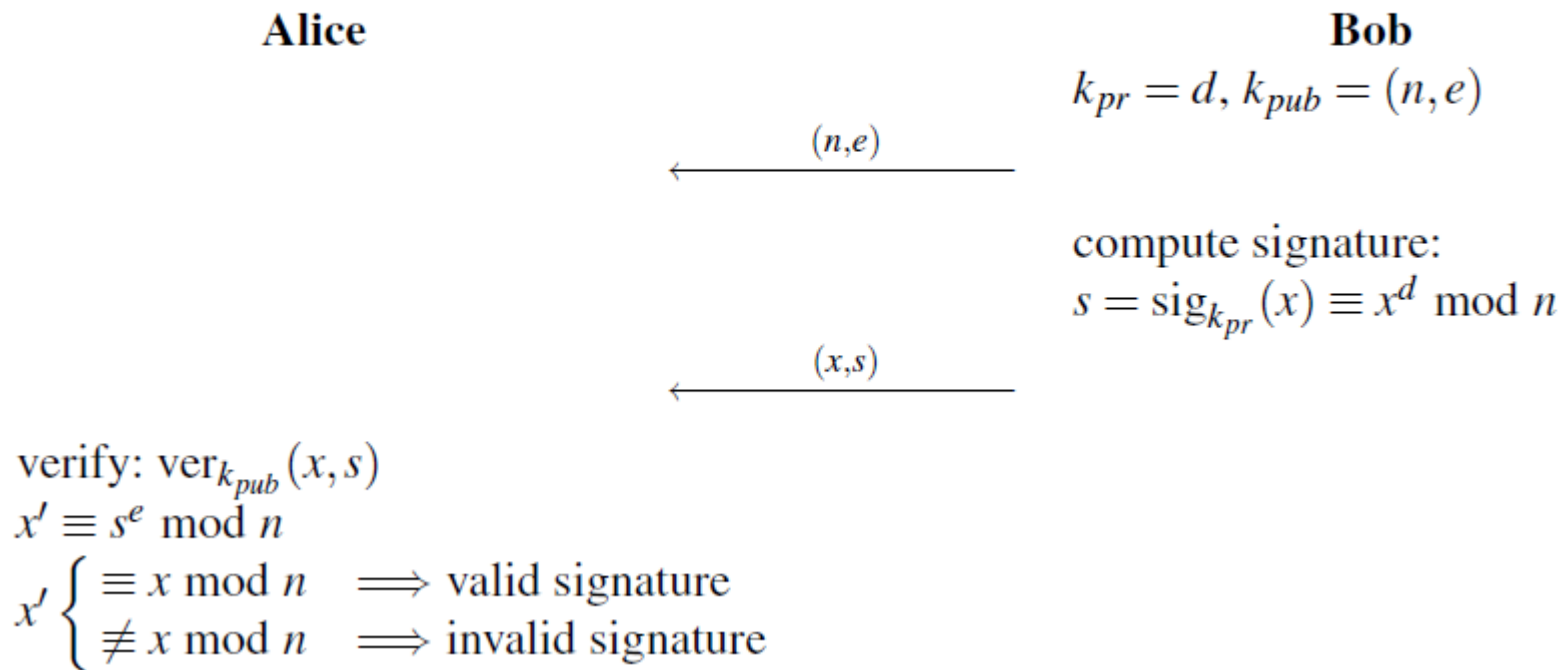
- ❑ The RSA signature scheme is based on RSA encryption.
- ❑ Therefore, its security relies on the difficulty of factoring a product of two large primes (the integer factorization problem).
- ❑ Suppose Bob wants to send a signed message x to Alice. He generates the same RSA keys that were used for RSA encryption.
- ❑ At the end of the set-up he has the following parameters:

RSA Keys

- Bob's private key: $k_{pr} = (d)$
- Bob's public key: $k_{pub} = (n, e)$

- ❑ The actual signature protocol is shown in the following figure. The message x that is being signed is in the range $(1, 2, \dots, n-1)$

Basic RSA Digital Signature Protocol



- ❑ It can be seen from the protocol in previous slide that:
- ❑ Bob computes the signature s for a message x by RSA-encrypting x with his private key k_{pr} .
- ❑ Bob is the only party who can apply k_{pr} , and hence the ownership of k_{pr} authenticates him as the author of the signed message.
- ❑ Bob appends the signature s to the message x and sends both to Alice.
- ❑ Alice receives the signed message and RSA-decrypts s using Bob's public key k_{pub} , yielding $\tilde{x}$.

- ❑ If $x^{\sim}$ and x match, Alice knows two important things:
 - i. The author of the message was in possession of Bob's secret key, If only Bob has had access to the key, it was in fact Bob who signed the message. This is called message authentication.
 - ii. The message has not been changed in transit, so that message integrity is guaranteed.

□ We now prove that the scheme is correct, i.e., that the verification process yields a “true” statement if the message and signature have not been altered during transmission.

□ We start from the verification operation $s^e \bmod n$:

$$s^e = (x^d)^e = x^{de} \equiv x \bmod n$$

□ Due to the mathematical relationship between the private and the public key, namely that:

$$d.e \equiv 1 \bmod \varphi(n),$$

□ raising any integer $x \in \mathbb{Z}_n$ to the $(d.e)$ th power yields the integer itself again

- ❑ The role of the public and the private keys are swapped compared to the RSA encryption scheme.
- ❑ Whereas RSA encryption applies the public key to the message x , the signature scheme applies the private key k_{pr} .
- ❑ On the other side of the communication channel, RSA encryption requires the use of the private key by the receiver, while the digital signature scheme applies the public key for verification.

- ❑ Suppose Bob wants to send a signed message ($x = 4$) to Alice.

Alice

Bob

1. choose $p = 3$ and $q = 11$
2. $n = p \cdot q = 33$
3. $\Phi(n) = (3 - 1)(11 - 1) = 20$
4. choose $e = 3$
5. $d \equiv e^{-1} \equiv 7 \pmod{20}$

$(n,e)=(33,3)$



compute signature for message

$x = 4$:

$$s = x^d \equiv 4^7 \equiv 16 \pmod{33}$$

$(x,s)=(4,16)$



verify:

$$x' = s^e \equiv 16^3 \equiv 4 \pmod{33}$$

$$x' \equiv x \pmod{33} \implies \text{valid signature}$$

- ❑ The first steps are exactly the same as it is done for an RSA encryption:
- ❑ Bob computes his RSA parameters and sends the public key to Alice.
- ❑ In contrast to the encryption scheme, now the private key is used for signing while the public key is needed to verify the signature.
- ❑ Alice can conclude from the valid signature that Bob generated the message and that it was not altered in transit, i.e., message authentication and message integrity are given.

- ❑ It should be noted that we introduced a digital signature scheme only.
- ❑ In particular, the message itself is not encrypted and, thus, there is no confidentiality.
- ❑ If this security service is required, the message together with the signature should be encrypted, e.g., using a symmetric algorithm like AES.

❑ **Algorithmic Attacks:**

- ❑ This type of attacks attempts to break the underlying RSA scheme by computing the private key d .
- ❑ The most general type of these attacks tries to factor the modulus n into the primes p and q .
- ❑ If an attacker succeeds with this, she/he can compute the private key d from e .
- ❑ In order to prevent factoring attacks the modulus must be sufficiently large, as discussed previously. In practice, 1024 bit or more are recommended.

RSA Digital Signature- Existential Forgery Attack

❑ Existential Forgery:

This attack allows an attacker to generate a valid signature for a random message x . The attack works as follows:

Existential Forgery Attack Against RSA Digital Signature

Alice

Oscar

Bob

$$k_{pr} = d$$

$$k_{pub} = (n, e)$$

$\leftarrow (n, e)$

$\leftarrow (n, e)$

1. choose signature:
 $s \in \mathbb{Z}_n$
2. compute message:
 $x \equiv s^e \pmod n$

$\leftarrow (x, s)$

verification:

$$s^e \equiv x' \pmod n$$

since $x' = x$
 $\implies$ valid signature!

- ❑ The attacker impersonates Bob, i.e., Oscar claims to Alice that he is in fact Bob.
- ❑ Because Alice performs exactly the same computations as Oscar, she will verify the signature as correct.
- ❑ However, by closely looking at Steps 1 and 2 that Oscar performs, one sees that the attack is somewhat odd.

- ❑ The attacker chooses the signature first and then *computes the message*. As a consequence, he *cannot control the semantics* of the message x .
- ❑ For instance, Oscar cannot generate a message such as “Transfer \$1000 into Oscar’s account”.
- ❑ Nevertheless, the fact that an automated verification process does not recognize the forgery is certainly not a desirable feature.
- ❑ For this reason, schoolbook RSA signature is rarely used in practice, and padding schemes are applied in order to prevent this and other attacks

❑ **Question:** What kinds of countermeasures can be implemented to secure RSA digital signature against Existential Forgery Attack?

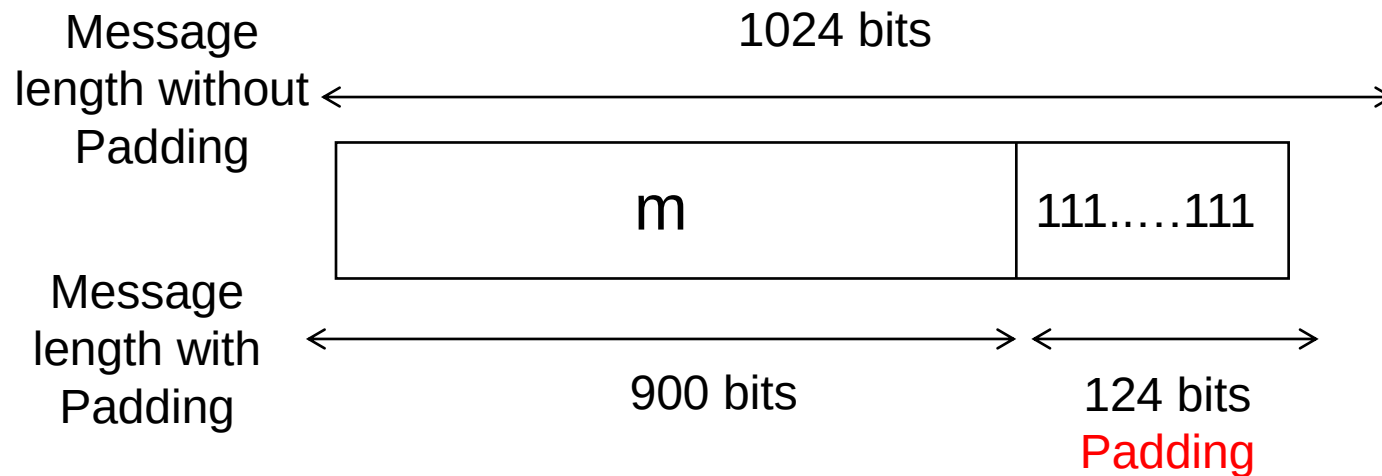
❑ **Answer:** Impose formatting rules on the message x , which can be checked by Alice

❑ The technique that can be implemented to achieve this: ***RSA Padding***.

❑ The existential forgery attack can be prevented by allowing only certain message formats.

❑ The formatting imposes a rule which allows the verifier, Alice in our examples, to distinguish between valid and invalid messages; this is called ***padding***.

- ❑ The formatting imposes a rule which allows the verifier to distinguish between valid and invalid messages; this is called **padding**.



- ❑ Oscar computes:

- ❑ $x \equiv s^e \pmod{n}$

- ❑ He has a chance of 2^{-124} to have a message x with 124 trailing bits with values 1s