NORTHERN ARIZONA UNIVERSITY®

INF 638
Cryptography & Cryptosystems

Section 10: PKI & Digital signatures

Dr. Bertrand Cambou
Professor of Practice NAU, Cybersecurity
School of Informatics, Computing, and Cyber-Systems
Bertrand.cambou@nau.edu

NORTHERN ARIZONA UNIVERSITY

INF 638: Cryptography & Cryptosystems

- 1- Motivation & Definitions
- 2- Elements of Number theory
- 3- Early Cryptographic methods
- 4- Symmetrical Cryptography: DES
- 5- Symmetrical Cryptography: AES
- 6- Quantum Cryptography: Key distribution
- 7- Elements of Asymmetrical Cryptography
- 8- Asymmetrical Cryptography: RSA
- 9- ECC Key Distribution
- 10- PKI & Digital Signatures
- 11- Hash Functions
- 12- Smartcards

NORTHERN ARIZONA UNIVERSITY

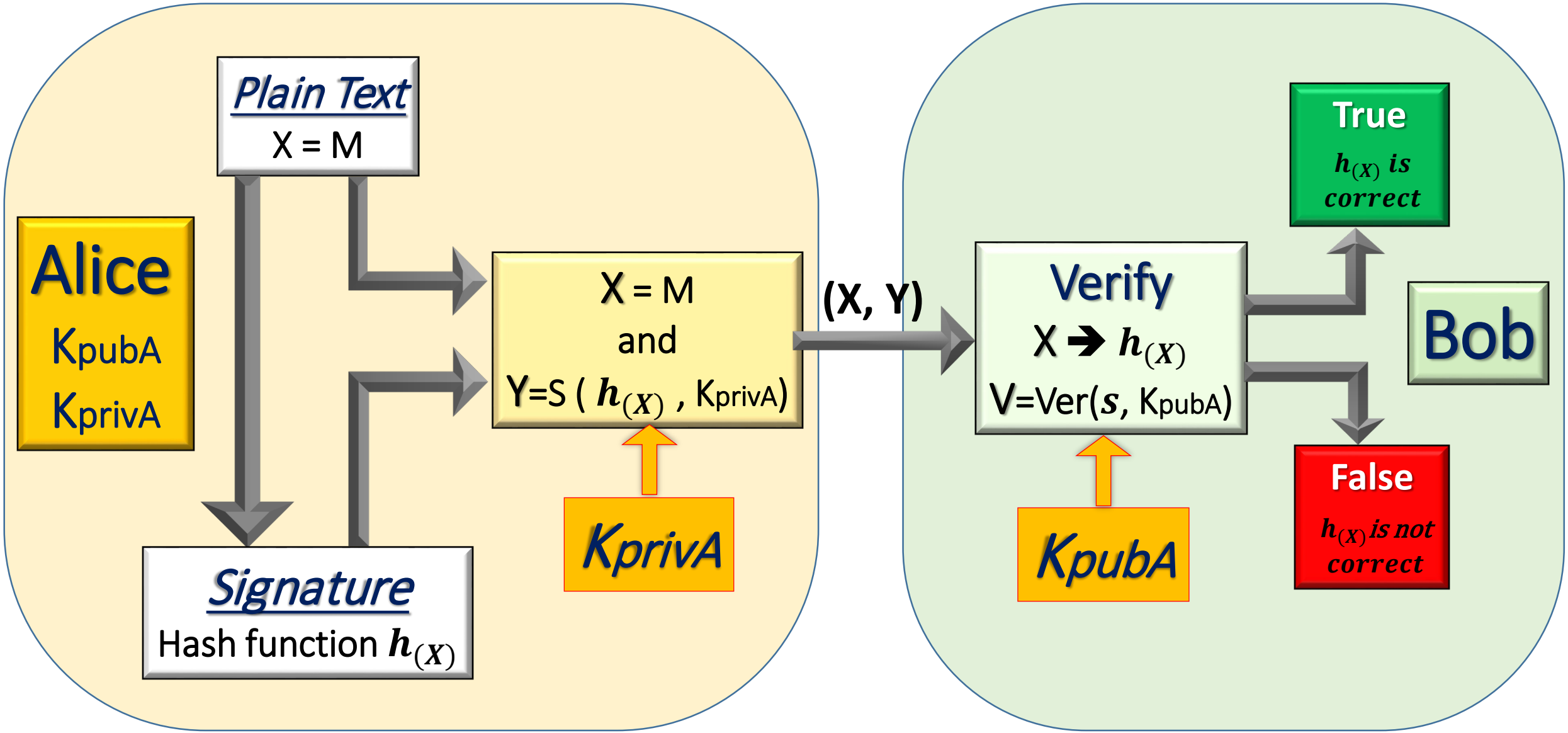
10- PKI & Digital signature

- 10-1 PKI for DSA
- 10-2 DSA with RSA
- 10-3 DSA with ECC
- 10-4 DSA with ElGamal
- 10-5 DSA – NIST
- 10-6 Schnorr DSA

Security services: Objectives of DSA

	Symmetrical	*Asymmetrical RSA encryption*	***DSA***	*Asymmetrical ECC or RSA double E*	*Asymmetrical with RN & Double E*
1- Confidentiality Information is kept secret from all but authorized parties	Yes Most of the time	Yes Most of the time	**No**	Yes Most of the time	Yes
2- Authentication The sender of the message is authentic	**No**	**No**	Yes	Yes	Yes
3- Integrity The message has not been modified during transmission	Yes	**No**	Yes	Yes	Yes
4- Non repudiation The sender cannot deny the creation of the message	**No**	**No**	Yes	Yes	Yes

Digital Signature: basic communication



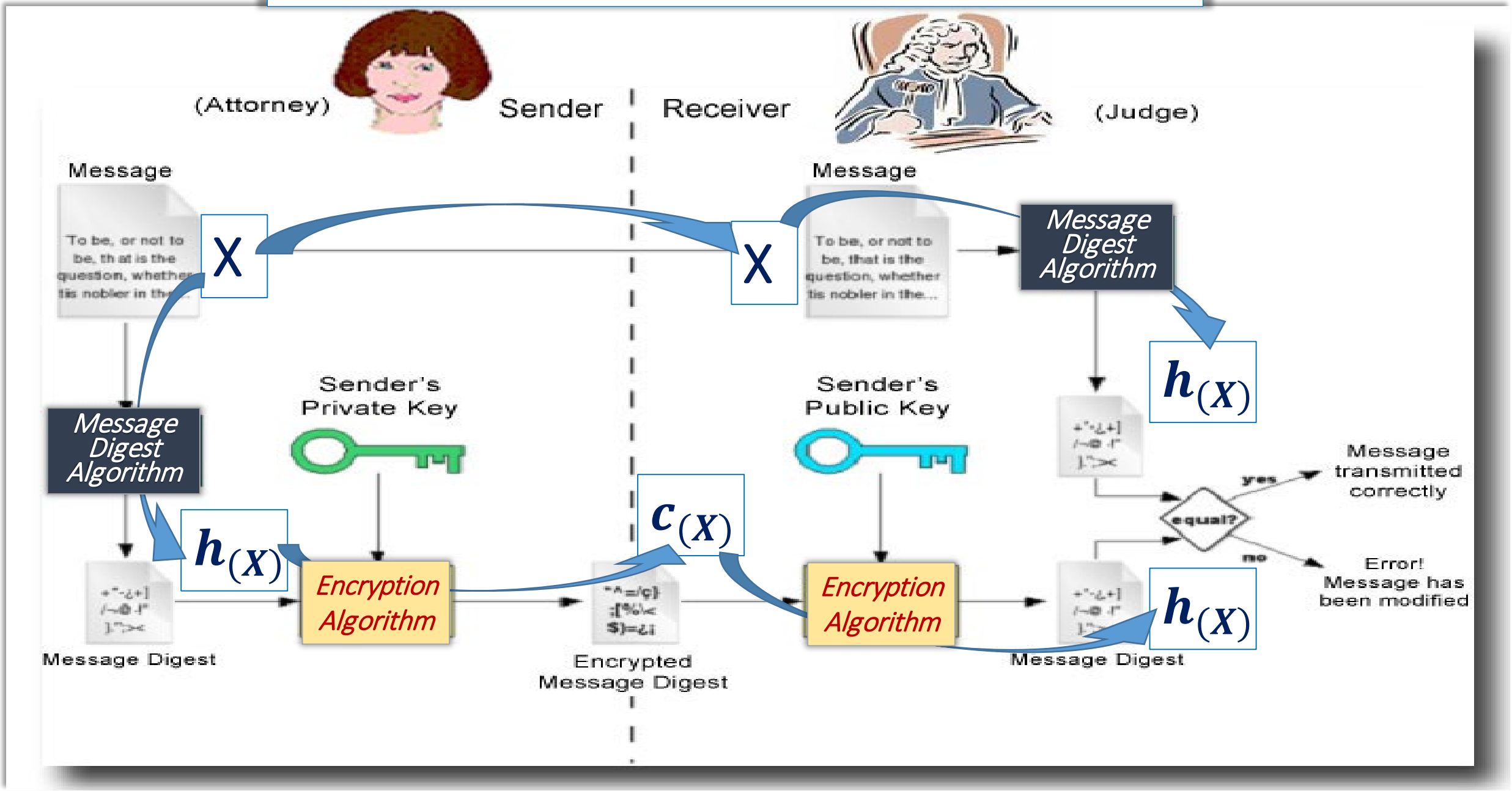
Digital Signature: basic communication

- Alice generate a ***signature*** Y = S based on X = M ,
  The hash $h_{(X)}$ is generated from X, and encrypted with private key ***KprivA***
- Alice ***transmit*** both X, and Y
- Bob ***verify*** that the message has been signed
  Use Alice public key ***KpubA***
- Ver(X, Y) has only two values:
  true if the same $h_{(X)}$ is generated from X
  false if $h_{(X)}$ not consistent

Improvement of the protection

- Alice encrypt (X, Y) with Bob's public key.
- Use of Hash functions and RNG

Digital Signature: Encryption algorithm



Key distribution for PKI and certificates

Two frameworks

- X.509: Centralized
- PGP (pretty good privacy): central authority

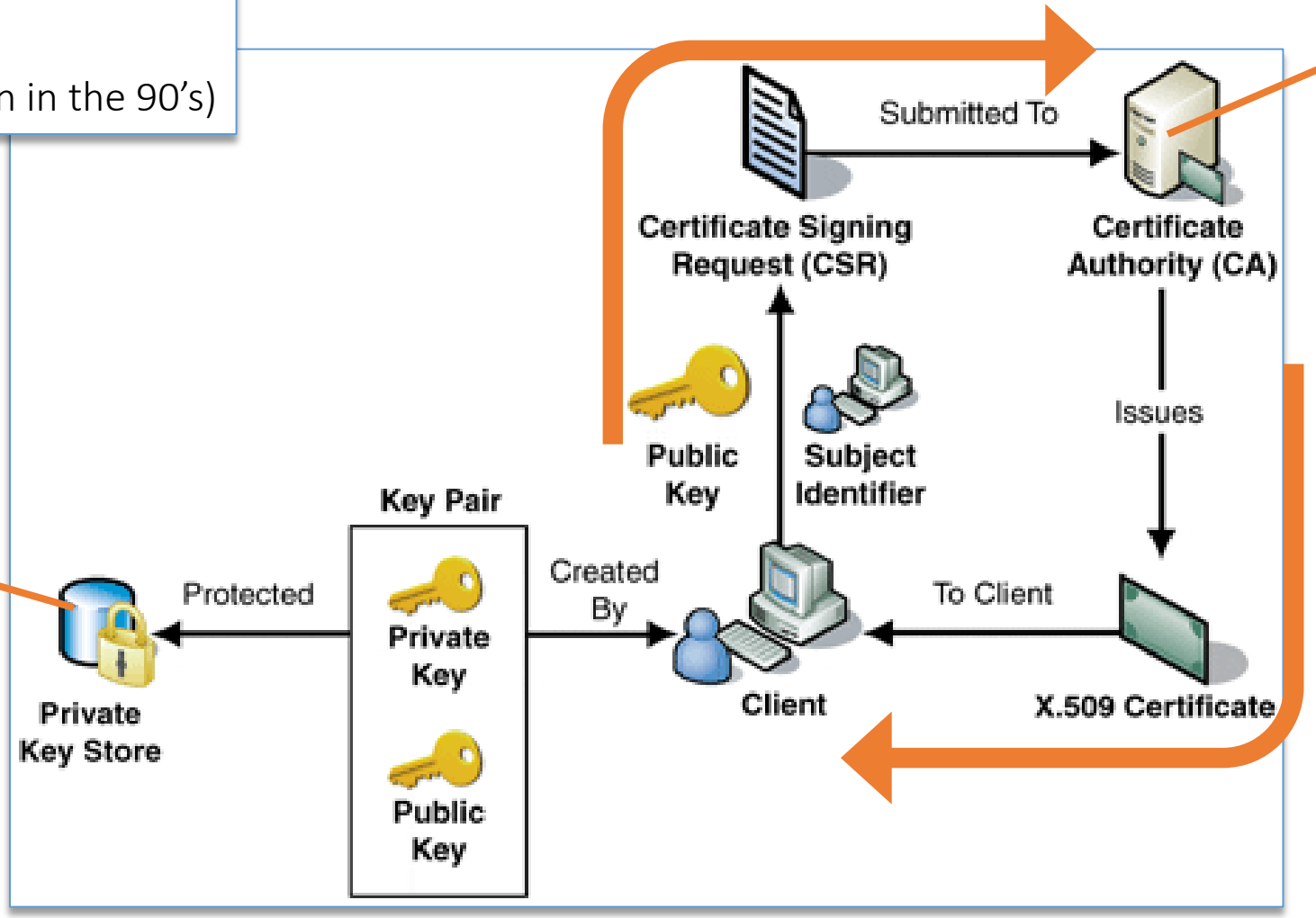
(invented by Philip Zimmermann in the 90's)

Centralized

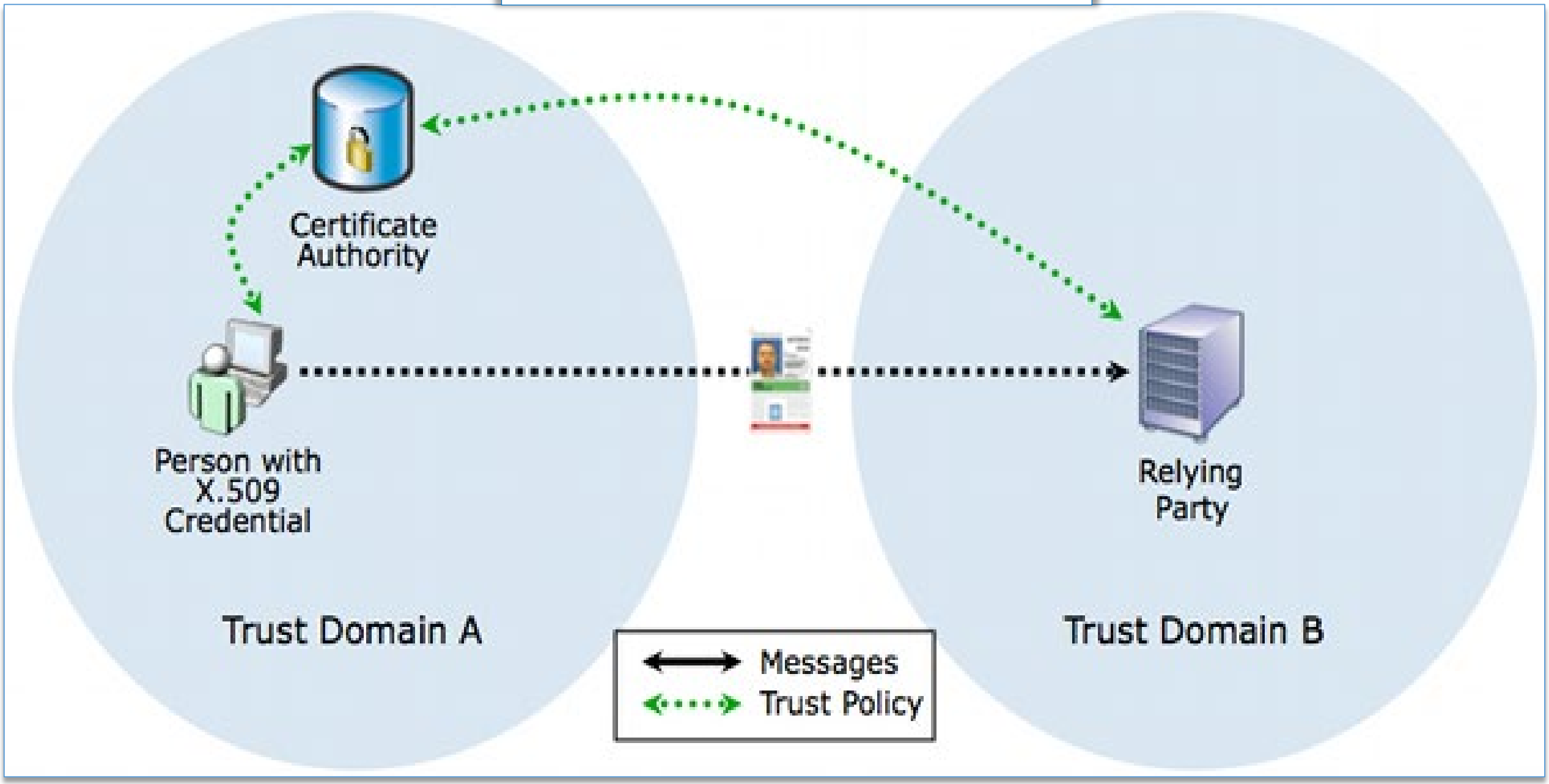
Certification

Key generation

Hardware Security Module (HSM)



Trusted transaction - PKI

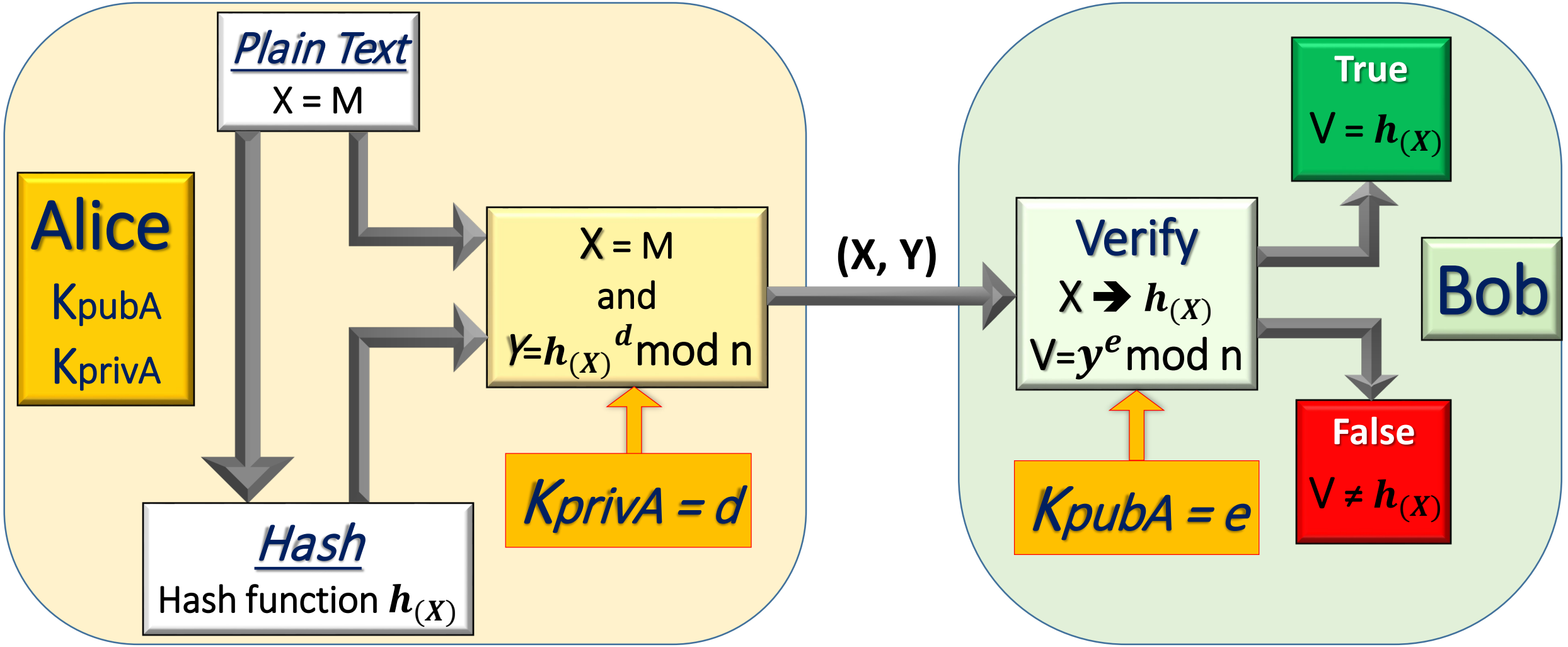


NORTHERN ARIZONA UNIVERSITY

10- PKI & Digital signature

- 10-1 PKI for DSA
- 10-2 DSA with RSA
- 10-3 DSA with ECC
- 10-4 DSA with ElGamal
- 10-5 DSA – NIST
- 10-6 Schnorr DSA

Digital Signature with RSA



Proof of correctness:

$$V \equiv Y^{e} \mod n \equiv h_{(X)}^{d\,e} \mod n = h_{(X)}^{1+k\,\phi(n)} \mod n = h_{(X)}$$

NORTHERN ARIZONA UNIVERSITY

10- PKI & Digital signature

- 10-1 PKI for DSA
- 10-2 DSA with RSA
- 10-3 DSA with ECC
- 10-4 DSA with ElGamal
- 10-5 DSA – NIST
- 10-6 Schnorr DSA

Digital signature with ECC: EC-DSA

Step-1: Key generation

- Use Elliptic Curve E ($y^2 \equiv x^3 + a\,.\,x + b \bmod m$) ; primitive $A\,(x_A, y_A)$

- The private key d is a random integer d: $0 < d < q$
  (q is the number of elements of the cyclic group)

- Compute $B = d\,.\,A$ ➔ $B\,(x_B, y_B)$

Public: (m, a, b, q, A, B)

Private Key: d

Digital signature with ECC: EC-DSA

Step-2: Signature

Message X ➔ Hash $h_{(X)}$

- Choose a random ephemeral integer k_E : $0 < k_E < q$
- Compute $R = k_E \,.\, A$ ➔ $R\,(x_R, y_R)$ ➔ $r = x_R$
- Compute $s \equiv (h_{(X)} + d \,.\, r)\, k_E^{-1} \bmod q$

[$\mathbf{k_E^{-1}}$ is the inverse of $\mathbf{k_E}$ modq]

Send *digital signature: (r, s)*

Digital signature with ECC: EC-DSA

Step-3: Verification

- Compute: $w \equiv s^{-1} \bmod q$
- Compute: $u_1 \equiv w.\ h_{(X)} \bmod q$
- Compute: $u_2 \equiv w.\ r \bmod q$
- Compute: $P = u_1.A + u_2\ B$ ➔ $P\ (x_P, y_P)$

If : $x_P \equiv r \bmod q$ ➔ *valid signature*

If : $x_P \neq r \bmod q$ ➔ *invalid signature*

Digital signature with ECC: Proof

$w \equiv s^{-1} \bmod q$

$u_1 \equiv w.h_{(X)} \bmod q$

$u_2 \equiv w.\ r \bmod q$

$P = u_1 \ . A + u_2 \ . B$

Proof:

$s \equiv (h_{(X)} + d \ . \ r\ )\ k_E^{-1} \bmod q$

$k_E \equiv (h_{(X)} + d \ . \ r\ )\ s^{-1} \bmod q \quad ;$

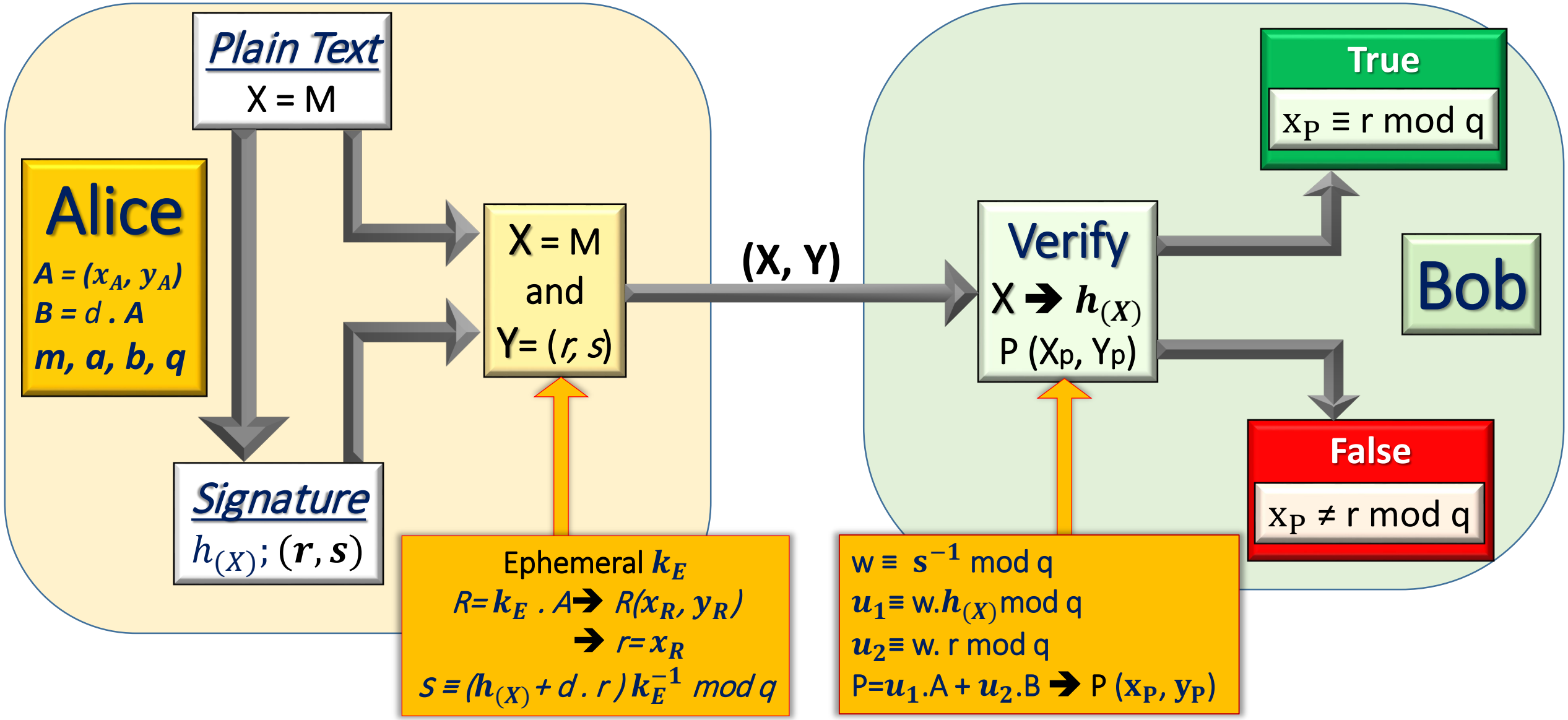
$k_E \equiv (h_{(X)} + d \ . \ r\ )\ w \bmod q$

$P = u_1.A + u_2\ B \equiv w.h_{(X)}A + w.r.d\ A \bmod q$

$\equiv (h_{(X)}\ A + r.d\ )\ w\ A \bmod q \equiv k_E \ . A \bmod q$

$k_E \ . A = R \rightarrow R(x_R, y_R) \rightarrow \quad r = x_R$

Summary: Digital Signature with ECC



Example: DSA with ECC

- **Key generation:**
  - Alice Choose: E with m=17; a=2; b=2; A = (5,1); q= 19; d =7
  - Compute: B= d . A = 7 . (5, 1) = (0, 6)
- **Signature:**
  - Sign hash message h(x) = 26
  - Pick $\mathbf{k_E} = 10$ ➔ R = $\mathbf{k_E}$. A = 10 . (5, 1) = (7, 11) ➔ r = 7
  - Compute: s ≡ (h(x) + d . r) $\mathbf{k_E^{-1}}$ mod 19 = (26 + (7 . 7)) . 2 = 17

$$\underline{(r, s) = 7, 17}$$

- **Verify:**
  - W= 17^{-1} ≡ 9 mod 19
  - u_1= 9 . 26 ≡ 6 mod 19
  - u_2= 9 . 7 ≡ 6 mod 19
  - P = 6 . (5, 1) + 6 (0, 6) = (7, 11)

$$\underline{\mathbf{x_P} = 7 \equiv r}$$

Homework 9

A- Can you propose a DSA scheme with ECC and the extended Galois field?

B- Can you propose a simple example demonstrating a DSA with $GF2^4$?

- Key generation
- Signature
- Verification

NORTHERN ARIZONA UNIVERSITY

10- PKI & Digital signature

- 10-1 PKI for DSA
- 10-2 DSA with RSA
- 10-3 DSA with ECC
- 10-4 DSA with ElGamal
- 10-5 DSA – NIST
- 10-6 Schnorr DSA

Digital signature with El Gamal

Step-1: Key generation

1-Alice chooses a large prime **p** and a primitive root **α**

2-She chooses a secret integer **z** and calculates:

$$\beta \equiv \alpha^z \bmod p$$

3-The values of **p**, **α** and **β** are public; **z** is kept private

Digital signature with ElGamal

Step-2: Signature

In order to sign the message M ➔ $h_{(X)}$ Alice follows:

1-She selects a secret ephemeral integer **k** such that

$$\gcd(\mathbf{k}, \mathbf{p} - 1) = 1$$

2- She computes: $\mathbf{r} \equiv \boldsymbol{\alpha}^{k} \bmod p$

3- She computes: $\mathbf{s} \equiv k^{-1} (h_{(X)} - z.\ r) \bmod (p - 1)$

[k^{-1} is the inverse of k mod (p-1)]

Send ***digital signature***: (**r**, **s**).

Digital signature with ElGamal

Step-3: Verification

The verification process can be performed by Bob using public information: **p**, **α** , **β** , $h_{(X)}$ and (**r**, **s**)

1- Bob computes:

$$v_1 \equiv \beta^{r} . r^{s} \bmod p$$

$$v_2 \equiv \alpha^{h(X)} \bmod p$$

2- The signature is declared valid if:

$$v_1 \equiv v_2 \bmod p$$

Digital signature with ElGamal

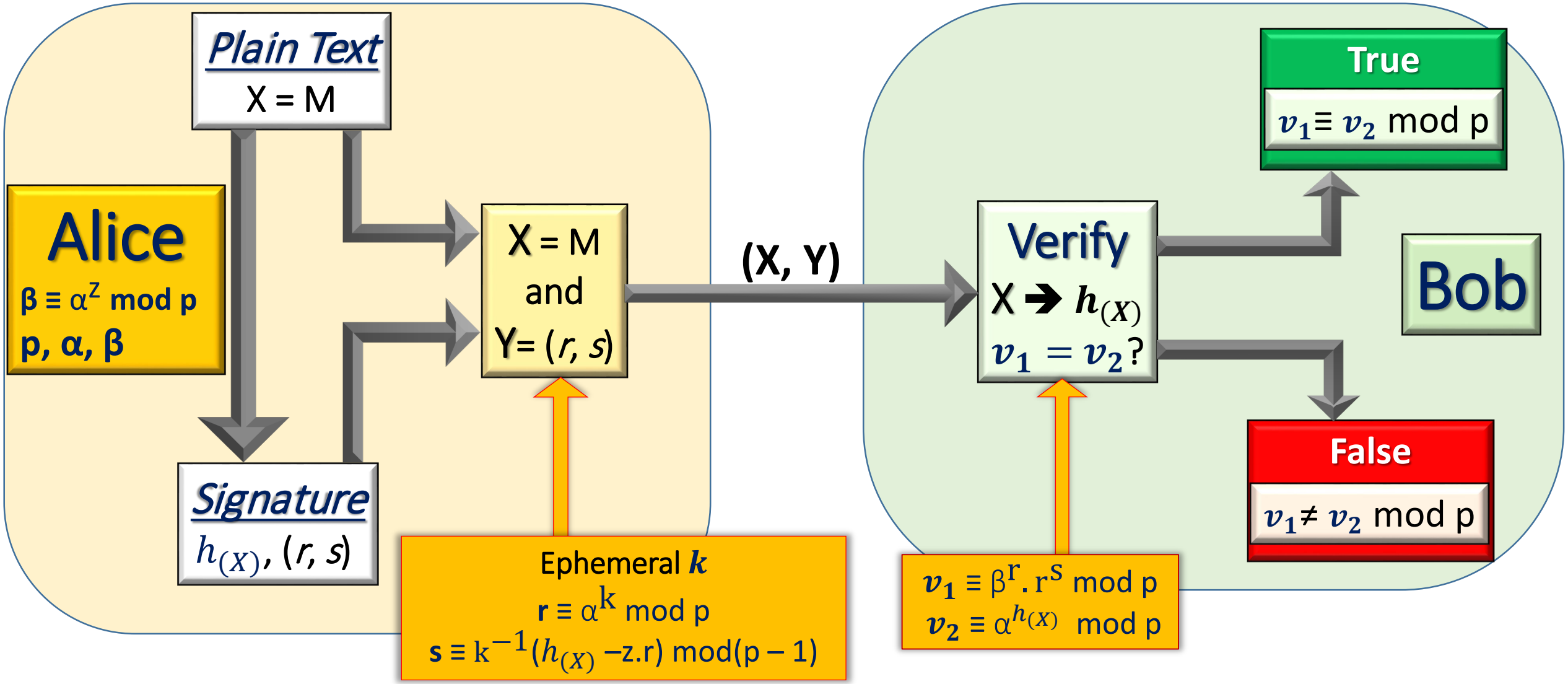
Proof of correctness

- We assume that the signature is valid.

$$s \equiv k^{-1}(h_{(X)} - z.r) \bmod (p-1)$$
$$s.k \equiv (h_{(X)} - z.r) \bmod (p-1)$$
$$h_{(X)} \equiv (s.k + z.r) \bmod (p-1)$$

$$v_2 \equiv \alpha^{h_{(X)}} \bmod p$$
$$\equiv \alpha^{s.k+z.r+L(p-1)} \bmod p$$
$$\equiv \alpha^{s.k+z.r} \bmod p \quad \text{(Fermat)}$$
$$\equiv \alpha^{k.s}\ \alpha^{z.r} \bmod p$$
$$\equiv r^{s}\ \beta^{r} \bmod p \equiv v_1$$

Summary: DSA with ElGamal



Example: DSA with ElGamal

- **Key generation:**
  - Alice Choose prime p=71; primitive root α =7; secret integer z=16:

$$\beta \equiv \alpha^{z} \bmod p \equiv 7^{16} \bmod 71 = 19$$

- **Signature:**
  - Sign hash message h(x) = 15
  - Pick $\mathbf{k} = 31$ ➔ $r \equiv \alpha^{k} \bmod p \equiv 7^{31} \bmod 71 = 11$
  - Compute: $s \equiv (h - z \,.\, r)\, k^{-1} \bmod p-1 = (15 - (16 \,.\, 11)) \,.\, 61 \bmod 70 = 49$

$$\underline{(r, s) = 11, 49}$$

- **Verify:**
  - $v_1 \equiv \beta^{r} . r^{s} \bmod p \equiv 19^{11} . 11^{49} \bmod 71 \equiv 64 \,.\, 17 \bmod 71 = 23$
  - $v_2 \equiv \alpha^{h} \bmod p \equiv 7^{15} \bmod 71 = 23$

NORTHERN ARIZONA UNIVERSITY

10- PKI & Digital signature

- 10-1 PKI for DSA
- 10-2 DSA with RSA
- 10-3 DSA with ECC
- 10-4 DSA with ElGamal
- 10-5 DSA – NIST
- 10-6 Schnorr DSA

DSA – NIST 1991

Step-1: Key generation

1- Alice finds a 160-bit prime **q**

2- She chooses a prime **p** such that **p – 1** is divisible by **q**

3- Alice chooses a primitive root ***g*** mod p:

$$\alpha \equiv g^{(p-1)/q} \bmod p$$

$$\alpha^{q} \equiv g^{(p-1)} \bmod p \equiv 1 \bmod p \quad \text{(Fermat)}$$

4- Alice chooses a secret **z** $\in (1, \ldots, q-1)$:

$$\beta \equiv \alpha^{z} \bmod p$$

5- Alice publishes (**p**, **q**, **α**, **β**); **z** is secret.

DSA – NIST 1991

Step-2: Signature

In order to sign the message M ➔ $h_{(X)}$, Alice follows:

1- She selects a secret ephemeral integer **k** ∈ (1, q – 1)

2- She computes: **r** ≡ (α^{k} mod p) mod q

3- She computes: **s** ≡ $k^{-1}(h_{(X)}$ +z.r) mod q

Send ***digital signature***: ($h_{(X)}$, **r**, **s**).

DSA – NIST 1991

Step-3: Verification

The verification process can be performed by Bob using public information: **p, q, α , β, M** and (**r**, **s**)

1- Bob computes:

$$v_1 \equiv s^{-1}.h_{(X)} \bmod q$$

$$v_2 \equiv s^{-1}.\, r \bmod q$$

$$v \equiv (\alpha^{v_1}.\beta^{v_2} \bmod p) \bmod q$$

2- The signature is declared valid if: **v = r**

DSA – NIST 1991

Proof of correctness

- We assume that the signature is valid.

$\mathbf{s} \equiv k^{-1}(h_{(X)} + z.\ r) \bmod \mathbf{q}$

$\mathbf{k} \equiv s^{-1}(h_{(X)} + z.\ r) \bmod \mathbf{q} \equiv (s^{-1}h_{(X)} + s^{-1}z.\ r) \bmod \mathbf{q}$

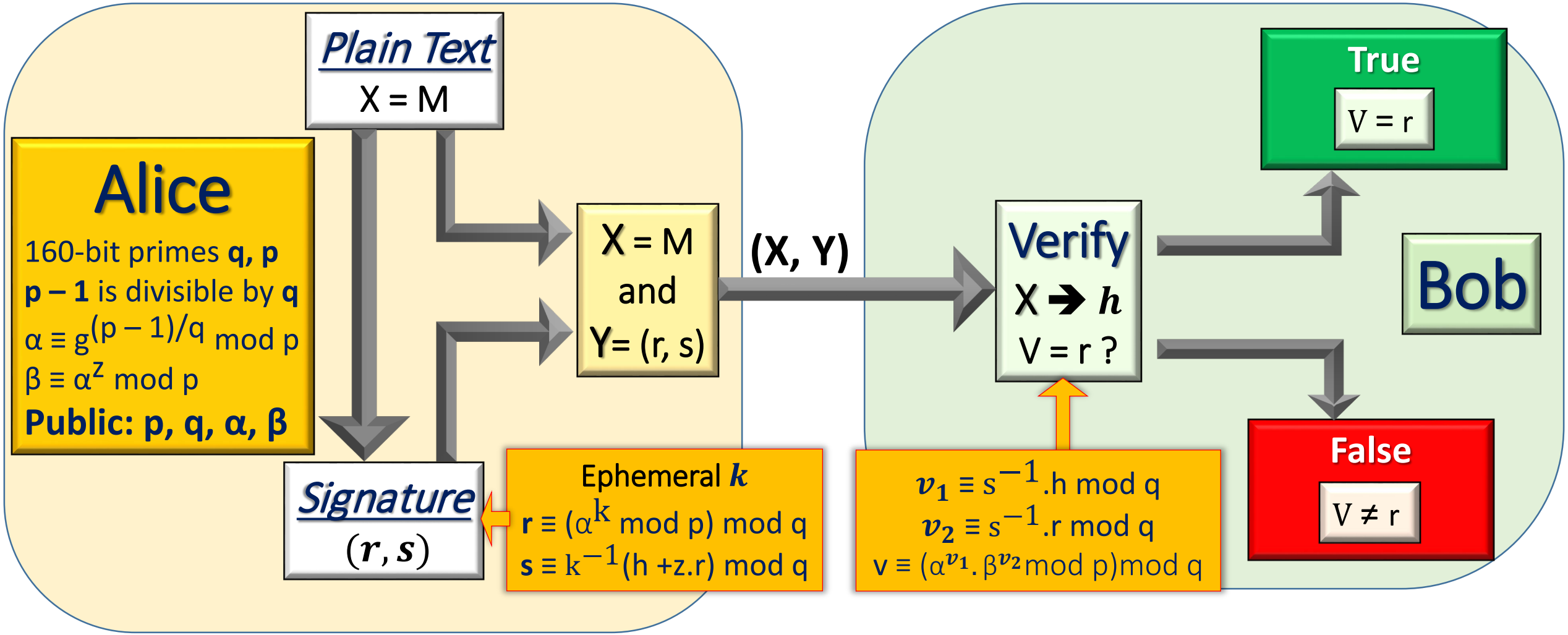
$\mathbf{r} \equiv (\alpha^{k} \bmod \mathbf{p}) \bmod \mathbf{q} \equiv (\alpha^{s^{-1}h_{(X)} + s^{-1}z.\ r + fq} \bmod \mathbf{p}) \bmod \mathbf{q}$

$\equiv (\alpha^{s^{-1}h_{(X)}}\ \alpha^{z.s^{-1}r} \bmod \mathbf{p}) \bmod \mathbf{q}$

$\equiv \alpha^{v_1}\ \beta^{v_2} \bmod \mathbf{p}) \bmod \mathbf{q} = \mathbf{v}$

$[\alpha^{q} \equiv 1 \bmod p\ ;\ v_1 = s^{-1}.h_{(X)} + dq\ ;\ v_2 = s^{-1}.r + cq\ ;\ \beta \equiv \alpha^{z} \bmod p]$

Summary: DSA – NIST 1991



Example: DSA – NIST 1991

- **Key generation:**
  - Alice Choose prime p=131; q=13 primitive root g =2 mod 13; z=6

$$\alpha \equiv g^{(p-1)/q} \bmod p \equiv 2^{10} \bmod 131 = 107$$

$$\beta \equiv \alpha^{z} \bmod p \equiv 107^{6} \bmod 131 = 45$$

- **Signature:**
  - Sign hash message h(x) = 27
  - Pick $\mathbf{k} = 4$
  - **r** ≡ (α^{k} mod p) mod q ≡ (107^{4} mod 131) mod 13 ≡ 84 mod 13 = 6
  - **s** ≡ k^{-1}(h+z.r) mod q ≡ 10 (27+6x6) mod 13 =6

$$\underline{(r, s) = 6, 6}$$

- **Verify:**
  - $v_1 \equiv s^{-1}.h \bmod q \equiv 11 \times 27 \bmod 13 = 11$
  - $v_2 \equiv s^{-1}.r \bmod q \equiv 11 \times 6 \bmod 13 = 1$
  - $v \equiv (\alpha^{v_1}.\beta^{v_2} \bmod p) \bmod q \equiv (107^{11}.45^{1} \bmod 131) \bmod 13 = 6$

NORTHERN ARIZONA UNIVERSITY

10- PKI & Digital signature

- 10-1 PKI for DSA
- 10-2 DSA with RSA
- 10-3 DSA with ECC
- 10-4 DSA with ElGamal
- 10-5 DSA – NIST
- 10-6 Schnorr DSA

DSA – Schnorr

Step-1: Key generation

1- Alice finds a 1024-bit prime **p**

2- She chooses **q** a prime factor of **p – 1**

q is about 160-bit long $q \in (1, q-1)$

3- She chooses a such as: $a^q \equiv 1 \bmod p$

$a \in (1, q-1)$:

4- She chooses a secret key $s < q$

5- She calculate $v \equiv a^{-s} \bmod p$

Alice publishes (p, q, a, v); s is secret.

DSA – Schnorr

Step-2: Signature

In order to sign the message **M**, Alice:

1- She selects a secret ephemeral integer $\mathbf{r} \in (1, q - 1)$

2- She computes: $\mathbf{x} \equiv \alpha^{r} \bmod \mathbf{p}$

4- Message **M** ➔ $\mathbf{e} = H(\mathbf{M}||\mathbf{x})$

3- She computes: $\mathbf{y} \equiv (\mathbf{r} + \mathbf{s}.\mathbf{e}) \bmod \mathbf{q}$

Send ***digital signature***: (**e**, **y**)

DSA – Schnoor

Step-3: Verification

The verification process can be performed by Bob using public information: **p, q**, **a** , **v, M** , and(**e**, **y)**

1- Bob computes:

$$x' \equiv a^{y} . v^{e} \bmod p$$

$$e' = H(M||x')$$

2- The signature is declared valid if: $e' = e$

DSA – Schnoor

Proof of correctness

$$x' \equiv a^{y}.v^{e} \bmod p$$

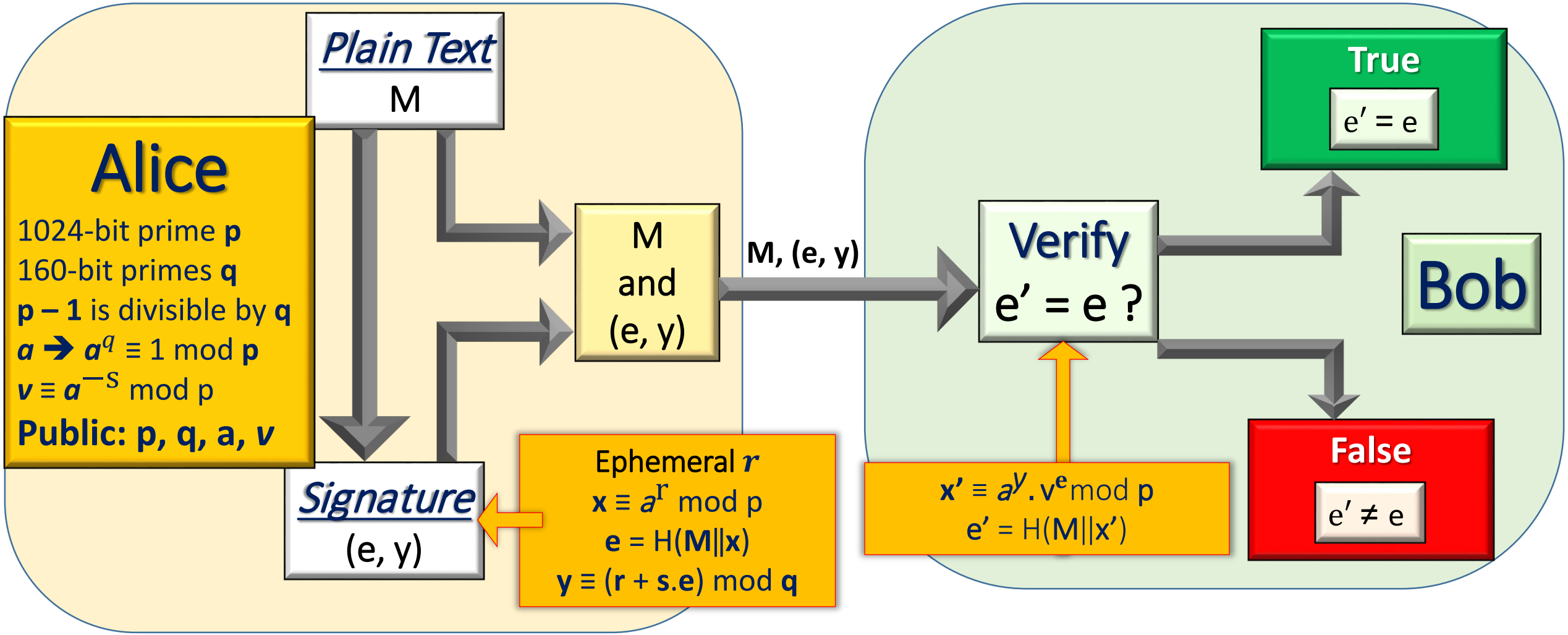
$$\equiv (a^{(r+s.e+\alpha q)})(a^{-s.e} + \beta p) \bmod p$$

$$\equiv (a^{(r+s.e+\alpha q)})(a^{-s.e} + \beta p) \bmod p$$

$$\equiv a^{(r+s.e)}\ a^{\alpha q}\ a^{(-s.e)} \bmod p \qquad [\ a^{q} \equiv 1 \bmod p\ ]$$

$$\equiv a^{r} \bmod p = x$$

Summary: DSA – Schnoor



Homework 10

Can you propose suggest a simple example demonstrating the DSA scheme with Schnorr scheme?

- Key generation
- Signature
- Verification

NORTHERN ARIZONA UNIVERSITY

All DSA schemes presented in section 10 are not quantum computing resistant!!!

10- PKI & Digital signature

- 10-1 PKI for DSA
- 10-2 DSA with RSA
- 10-3 DSA with ECC
- 10-4 DSA with ElGamal
- 10-5 DSA - NIST

Hash-Based DSA → See section 11

NORTHERN ARIZONA UNIVERSITY®

QUESTIONS ?

Dr. Bertrand Cambou
Professor of Practice NAU, Cybersecurity
School of Informatics, Computing, and Cyber-Systems
Bertrand.cambou@nau.edu