Elgamal Encryption Scheme

Saturday, 7 December 2019 3:13 PM

ENCRYPTION WITH DLP:

SERVICE	Algorithmic Family		
	Integer Factor	DL	
		Z_p^*	Elliptic Curve
Key exchange	RSA	D-H	ECDH
Digital Signature	RSA	Elgamal	ECDSA
Encryption	RSA	Elgamal	ECXX

Goal: Develop Encryption scheme from D-H Key Exchange

D-H

ALICE — $\alpha, p \rightarrow$ — BOB

(2) $A \equiv \alpha^a \bmod p$

$\leftarrow B$ (1)

$A \rightarrow$ (3)

(3) $B^a \equiv \alpha^{ab} \equiv K_{AB} \bmod p$

(6) $A^b \equiv \alpha^{ab} \equiv K_{AB} \bmod p$

encryption

message x

(4) $y \equiv x \cdot K_{AB} \bmod p$

y (5)

(7) $x \equiv y \cdot K_{AB}^{-1} \bmod p$

ELGAMAL ENCRYPTION

→ invented around 1985

→ very similar to D-H but reordering of steps

→ also spelled as ElGamal.

ALICE — BOB

BOB: choose p, primitive element α; $k_{pr} = d \in \{2, 3, \dots, (p-2)\}$; $k_{pub} = \beta = \alpha^d \bmod p$

(1) (β, p, α) →

Alice chooses $i \in \{2, \dots, p-2\}$

(2) $k_E \equiv \alpha^i \bmod p$ (Ephemeral key)

(3) $k_m = \beta^i \bmod p$ (masking key)

(4) $y \equiv x \cdot k_m \bmod p$

(5) (y, k_E) →

(6) $k_m = (k_E)^d \bmod p$

(7) $x \equiv y \cdot k_m^{-1} \bmod p$

proof of correctness:

Bob computes: $(\quad)^{-1} \bmod p$

Remarks:

(1) Advantage of Elgamal over direct D-H encryption:
Bob's public key is fixed and α, p are chosen by him.

(2) k_E must be different for every plaintext. $k_{pub} = \beta$ (1) randomiser

(3) $x_1 \rightarrow$ e() $\rightarrow y_1$; $y_2 = x_1 \rightarrow$ e() $\rightarrow y_2$

is $y_2 \overset{?}{=} y_1$

No, because i changes every time $\Rightarrow k_E$ changes $\Rightarrow y_2 \neq y_1$

∴ Elgamal is a "PROBABILISTIC ENCRYPTION" scheme unlike schoolbook RSA

COMPUTATIONAL ASPECTS:

Alice and Bob have to compute

$\beta = \alpha^d$, $k_E = \alpha^i$, $k_m = \beta^i$ } SQUARE AND MULTIPLY ALGORITHM

Q: (1) $k_m \equiv k_E^d \bmod p$

(2) k_m^{-1} (extended euclidean)

$x \equiv y k_m^{-1}$

we can merge (1) and (2) by Fermat's little theorem

$k_E \in Z_p^* : k_E^{p-1} \equiv 1 \bmod p$

$k_m^{-1} \equiv (k_E^d)^{-1} \cdot 1$

$= k_E^{-d} \cdot k_E^{p-1}$

$\equiv k_E^{p-1-d} \bmod p$

step (7) becomes $x \equiv y k_m^{-1} \equiv y k_E^{p-1-d} \bmod p$

ATTACKS

parameters that oscar doesn't know are i and d

(1) compute DLP

$d = \log_\alpha \beta$

& then he can do same as Bob $[\rightarrow k_m = k_E^d ; x \equiv y k_m^{-1}]$

Or

$i = \log_\alpha k_E$

$[\rightarrow k_m = \beta^i , x \equiv y k_m^{-1}]$

∴ DLP needs to be a computational hard problem

proof of correctness:

Bob computes:

$$y \cdot K_M^{-1} \equiv y \cdot \left((K_E)^d\right)^{-1} \mod p$$

$$\equiv x \cdot K_M \cdot K_E^{-d} \mod p$$

$$\equiv x \cdot \beta^i \cdot (\alpha^i)^{-d} \mod p$$

$$\equiv x (\alpha^d)^i \cdot (\alpha^i)^{-d} \mod p$$

$$\equiv x \cdot \alpha^{id} \cdot \alpha^{-id} \mod p$$

$$\equiv x \cdot \alpha^{id-id} \mod p$$

$$\equiv x \cdot 1 \mod p$$

$$\equiv x \mod p$$

$\therefore$ DLP needs to be a computational hard problem

$\therefore p$ must be large $p \geq 2^{1024}$

(2) Attack Re-use of secret exponent i

$K_E = \alpha^i$, $K_M = \beta^i$

$y_1 = x_1 \cdot K_M$ (y_1, K_E)

$y_2 = x_2 \cdot K_M$ (y_2, K_E)

assume: Oscar knows x_1 (is a KNOWN PLAINTEXT)

$K_M = y_1 \cdot x_1^{-1} = y_2 \cdot x_2^{-1}$

$x_2 \equiv y_2 y_1^{-1} x_1 \mod p$