Introduction to Cryptology

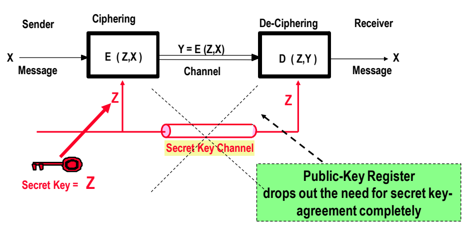
Lecture-11
Public-Key Cryptography
ElGamal Public-Key Crypto-System

15.05.2023, v48

Lecture Outlines

- Public-key Objective
- ElGamal Public-Key Encryption System
- ElGamal Public-Key Signature System
- ElGamal Security considerations
- Public Key Signature Standard

The Target of Public Key Cryptography



ElGamal Crypto-System, 1985

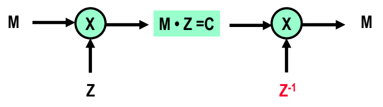
Taher ElGamal

Taher ElGamal: BsC, EE from Cairo University,
MsC and PhD Stanford University advisor M. Hellmann
ElGamal Cryptosystem became a NIST standard called DSA in 1994.

Basic idea: Multiplication instead of exponentiation

ElGamal Crypto-System 1985

Basic idea: Multiplication instead of exponentiation



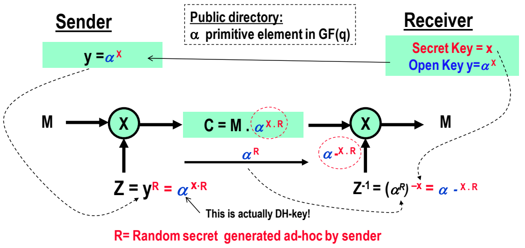
The inverse key Z^{-1} should not be computable if Z is known

Is that possible?: Yes,
By using groups arithmetic in in GF !

ElGamal Crypto-System 1985

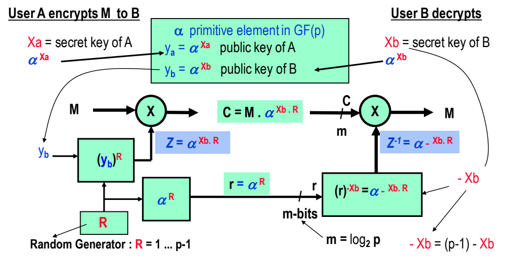
Basic idea: Multiplication instead of exponentiation for encryption

That is possible by using arithmetic in in GF(q) !



R= Random secret generated ad-hoc by sender

ElGamal Secrecy-System (1985)

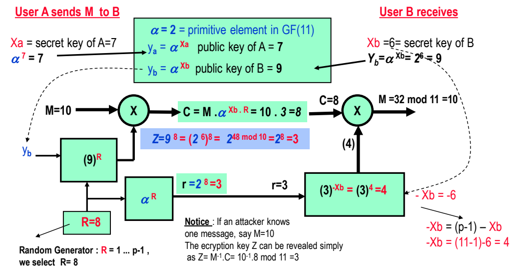


Notice 1: a new R is needed for every M! Otherwise, Z can be easily revealed if M is known!
Notice 2: The scheme applies similarly over $GF(2^m)$, with α as a primitive element in $GF(2^m)$.

Example 1: Setup ElGamal Encryption System using GF(11). Send the message M=10 from user A to B. The secret key of B is6and for A is 7

Solution : p=11=2 . 5+1, Possible orders = divisors of p-1=2x5, that is 1,2,5,10.

Computing order of α=2: $2^2=4 \neq 1$, $2^3=8$, $2^4=5$, $2^5=10 \neq 1$, => order of 2 is 10 => 2 is a primitive element !.



Notice : If an attacker knows one message, say M=10
The ecryption key Z can be revealed simply as $Z= M^{-1}.C= 10^{-1}.8 \bmod 11 =3$

ElGamal Crypto-System (1985)

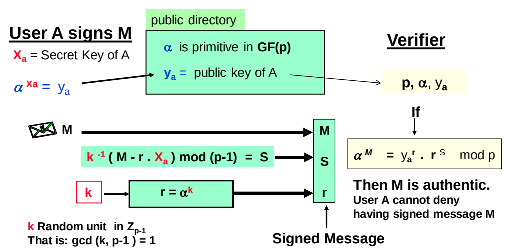
Advantages

- based on discrete logarithm problem
- Security is as that of DH system
- DL problem needs less key-bits than RSA for the same security
- Asymmetric workload: good for some applications
- ElGamal encryption is probabilistic, meaning that a single plaintext would be **encrypted** to many possible ciphertexts as a new random R is required for each encrypted block..

Disadvantages

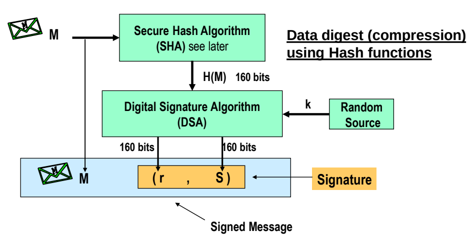
- The cryptogram needs more bits than the plaintext (double)
- A new random is needed for every encrypted message
- Asymmetric workload: bad for some applications

ElGamal Signature Scheme



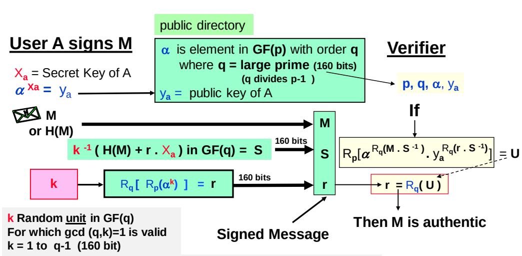
Digital Signature Standard DSS (1994)

Explicit true signature based on ElGamal Signature System (1985)



Digital Signature Algorithm DSA

Standardized (1994) Based on ElGamal Signature Scheme

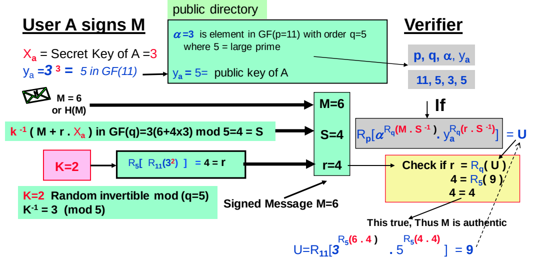


- p is a prime with 512 ... 1024 bits, q divides p-1 with a size of 160 bits,
- fresh k is required for every message!

Example 2: 1. Sign the message M=6 by using the Digital Signature Algorithm DSA, Use GF(p)=GF(11).
2. Check the resulting electronic signature

Solution: P=11=2 . 5+1, Possible orders = divisors of p-1=2x5, that is 1,2,5,10. Select q=5

Computing order of α=3: $3^2=9\neq1$, $3^3=5$, $3^4=4$, $3^5=1$ => order of 3 is 5



ElGamal Signature System (1985), DSS (1994)

Advantages

- Computations on Signer site are less complex than verifier site
- Security is based on the discrete logarithm problem which is still seen as computationally infeasible.

Disadvantages

- A new random is required to sign every message
- more computations than RSA are needed
- DSS may be less secure than RSA as the security in GF(q) with the order of about 160 Bits

Security of ElGamal Public Key Crypto-System (Equivalent to DH system)

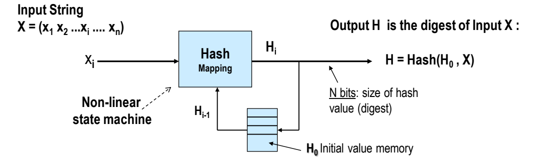
Security considerations and known facts:

1. Based on the assumption/claim that the **discrete logarithm** is still not efficiently computable according to the public literature
2. A primitive element α from GF(p) or GF(2^m) is used to make exhaustive search algorithms infeasible. If $y=\alpha^t$, only y and α are known. To break the system, we need to find t. To get t, α is repeatedly multiplied by itself i times when $\alpha^i=y$, then t=i.
   The order of α (as a primitive element) is p-1 in GF(p) or 2^m-1 in GF(2^m).
   Therefore, p is selected as 1000 to 4000 bits prime or m> 1000.
3. Caution: There is no evidence that no efficient algorithms can be found to break the system.
4. **p-1** should have large prime factor to make the discrete logarithm computation infeasible (p is called a strong prime).

Hash Functions

Hash functions are needed to generate message digest

Iterated Hash Function: generates a digest of the data after being sequentially processed through the so-called Hash function. In general as follows:



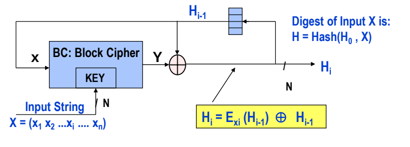
Example: SHA (Secure Hash Algorithm) proposed as a standard with DSA with N=160 bits (exposed to many attacks !) not more recommended !!!

Few Recommended "Practical Hash Functions" by deploying "Block Ciphers"

Hash Functions

Based on block ciphers
DM Scheme (Davis and Meyer)

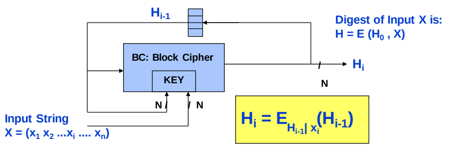
Cipher key length = Hash Block length = N



Hash Functions

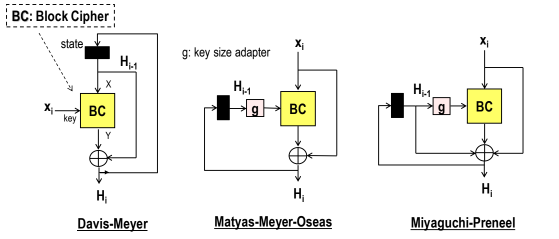
Based on block ciphers
LM Scheme (Lai and Massey 1992)

Cipher key length = 2 x Cipher block length N

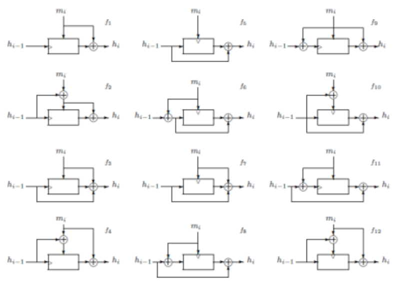


Traditional Hash Functions

Based on deploying block ciphers (BC)
(well known constellations. See NIST standards)

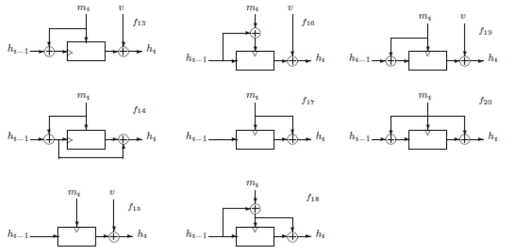


Block-Cipher based Hash Functions alternatives 1/2



Source: Handbook of Applied Cryptography
by Alfred J. Menezes, Paul C. Van Oorschot, Scott A. Vanstone, CRC Press (October 16, 1996) (available free of charge on the WEB)

Block-Cipher based Hash Functions alternatives 2/2



Source: Handbook of Applied Cryptography
by Alfred J. Menezes, Paul C. Van Oorschot, Scott A. Vanstone, CRC Press (October 16, 1996) (available free of charge on the WEB)