Cryptography
Lecture 7

Dr. Panagiotis Rizomiliotis
Assistant Professor
Dep. Of Informatics and Telematics
Harokopio University of Athens

El Gamal

Agenda

- Public key main schemes
  - Integer Factorization: RSA
  - **Discrete Logarithm: El Gamal**
- Digital Signatures
- Key Agreement
- Encryption
- AKE
- Key Encapsulation

DISCRETE LOGARITHM

- $Z_n^*=\{1,2,3,\ldots,n-1\}$
- Definition. Let $b \in Z_n^*$. The order of b is the smallest positive integer satisfying $b^e \equiv 1 \pmod n$.
- $Z_p^* = <\alpha>$, i.e. $ord(\alpha) = p-1$. when n=p=prime integer
- **Example**

$Z_7^* = <3>$ $3^1=3, 3^2=2, 3^3=6, 3^4=4, 3^5=5, 3^6=1$

$Z_{13}^* = <2>$ $2^1=2, 2^2=4, 2^3=8, 2^4=3, 2^5=6, 2^6=12, 2^7=11, 2^8=9, 2^9=5, 2^{10}=10, 2^{11}=7, 2^{12}=1$

DISCRETE LOGARITHM

- If g is a generator of Z_n^*, then for all y there is a unique x (mod $\phi(n)$) such that

$$y = g^x \bmod n$$

- This is called the **discrete logarithm** of y and we use the notation

$$x = \log_g(y)$$

- The discrete logarithm is conjectured to be hard as factoring.

- **Example**

$Z_{13}^* = <2>$ $2^1=2, 2^2=4, 2^3=8, 2^4=3, 2^5=6, 2^6=12, 2^7=11, 2^8=9, 2^9=5, 2^{10}=10, 2^{11}=7, 2^{12}=1$

$Log_2(5) = 9$.

ELGAMAL PUBLIC-KEY CRYPTOSYSTEM

- **SetUp (Ring of integers)**

- Choose a prime number p (selected so that it is hard to solve the discrete log problem)
- All operations in the ring Z^*_p

1. Randomly select a generator g for G
2. Randomly select an element $a \in Z^*_p$
3. Compute $\beta = g^a \bmod p$

- Public Key: (g, β) and the prime p (some description of the ring)
- Private Key: a

ELGAMAL

- Invented in 1985
- Designed by Dr. Taher Elgamal
- Based on the difficulty of the discrete log
- problem
- No patents
- Digital signature and Key-exchange variants

- Works over various groups
- ✓ Z_p,
- ✓ Multiplicative group $GF(p^n)$,
- ✓ Elliptic Curves

ELGAMAL PUBLIC-KEY CRYPTOSYSTEM

- **Encryption**
- Encryption of the message m

- Randomly select an element $k \in Z_p$
- Compute the ciphertext:
- $C = (c_1, c_2)$
  $= (g^k, m * \beta^k)$
- Delete k!

- **Decryption of C**
- Decryption of the ciphertext C = ()
- Compute
- $c_2 * (c_1^a)^{-1} = (m * \beta^k) * (g^{ka})^{-1} = m * \beta^k * (\beta^k)^{-1} = m$

ELGAMAL: EXAMPLE

- **SetUp (Ring of integers)**
- Choose a prime number p=11.
  - g = 2
  - a = 8
  - Compute $\beta = 2^8 \pmod{11} = 3$
- Public key: (2,3), Z_{11}^*
- Private key: 8

- **Encryption:**
- For m=7, k=4, we compute $C = (2^4, 7 * 3^4) = (5, 6)$

- **Decryption:**
- $6 * (5^8)^{-1} = 6 * 4^{-1} = 6 * 3 \pmod{11} = 7$

ELLIPTIC CURVE CRYPTOGRAPHY (ECC)

- "Elliptic Curve Cryptography" is not a new cryptosystem
- Elliptic curves are a different way to do the math in public key system

- Elliptic curves may be more efficient
- Fewer bits needed for same security

- For equivalent key lengths computations are roughly equivalent
- Hence for similar security ECC offers significant computational advantages

- RFC690: Fundamental Elliptic Curve Cryptography Algorithms

RSA VS EL GAMAL

- A disadvantage of ElGamal encryption is that there is message expansion by a factor of 2. That is, the ciphertext is twice as long as the corresponding plaintext.
- El Gamal is by design probabilistic.
- RSA is more mature and has better marketing
- El Gamal can achieve much better performance.

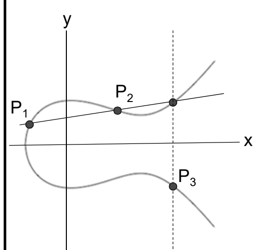
What is an Elliptic Curve?

- An elliptic curve E is the graph of an equation of the form

  $$y^2 = x^3 + ax + b$$

- Also includes a "point at infinity"
- What do elliptic curves look like?
- See the next slide!

Elliptic Curve Picture



- Consider elliptic curve
  E: $y^2 = x^3 - x + 1$
- If P_1 and P_2 are on E, we can define
  $P_3 = P_1 + P_2$
  as shown in picture
- Addition is all we need

Elliptic Curve Math

- Addition on: $y^2 = x^3 + ax + b \pmod p$

  $P_1=(x_1,y_1)$, $P_2=(x_2,y_2)$
  $P_1 + P_2 = P_3 = (x_3,y_3)$ where
  $x_3 = m^2 - x_1 - x_2 \pmod p$
  $y_3 = m(x_1 - x_3) - y_1 \pmod p$
  And $m = (y_2-y_1)*(x_2-x_1)^{-1} \bmod p$, if $P_1 \neq P_2$
  $m = (3x_1^2+a)*(2y_1)^{-1} \bmod p$, if $P_1 = P_2$
  Special cases: If m is infinite, $P_3 = \infty$, and
  $\infty + P = P$ for all P

Points on Elliptic Curve

- Consider $y^2 = x^3 + 2x + 3 \pmod 5$
  $x = 0 \Rightarrow y^2 = 3 \Rightarrow$ no solution (mod 5)
  $x = 1 \Rightarrow y^2 = 6 = 1 \Rightarrow y = 1,4 \pmod 5$
  $x = 2 \Rightarrow y^2 = 15 = 0 \Rightarrow y = 0 \pmod 5$
  $x = 3 \Rightarrow y^2 = 36 = 1 \Rightarrow y = 1,4 \pmod 5$
  $x = 4 \Rightarrow y^2 = 75 = 0 \Rightarrow y = 0 \pmod 5$
- Then points on the elliptic curve are
  (1,1) (1,4) (2,0) (3,1) (3,4) (4,0) and the point at infinity: ∞

Elliptic Curve Addition

- Consider $y^2 = x^3 + 2x + 3 \pmod 5$. Points on the curve are (1,1) (1,4) (2,0) (3,1) (3,4) (4,0) and ∞
- What is $(1,4) + (3,1) = P_3 = (x_3,y_3)$?
  $m = (1-4)*(3-1)^{-1} = -3*2^{-1}$
  $= 2(3) = 6 = 1 \pmod 5$
  $x_3 = 1 - 1 - 3 = 2 \pmod 5$
  $y_3 = 1(1-2) - 4 = 0 \pmod 5$
- On this curve, (1,4) + (3,1) = (2,0)

Finite Elliptic Curves

- Elliptic curve cryptography uses curves whose variables & coefficients are finite
- have two families commonly used:
  - prime curves $E_p(a,b)$ defined over Z_p
    - use integers modulo a prime
    - best in software
  - binary curves $E_{2m}(a,b)$ defined over $GF(2^n)$
    - use polynomials with binary coefficients
    - best in hardware

ECC Diffie-Hellman

- can do key exchange analogous to D-H
- users select a suitable curve $E_q(a,b)$
- select base point $G=(x_1,y_1)$
  - with large order n s.t. $nG=O$
- A & B select private keys $n_A<n$, $n_B<n$
- compute public keys: $P_A=n_AG$, $P_B=n_BG$
- compute shared key: $K=n_AP_B$, $K=n_BP_A$
  - same since $K=n_An_BG$
- attacker would need to find k, hard

Elliptic Curve Cryptography

- ECC addition is analog of modulo multiply
- ECC repeated addition is analog of modulo exponentiation
- need "hard" problem equiv to discrete log
  - $Q=kP$, where Q,P belong to a prime curve
  - is "easy" to compute Q given k,P
  - but "hard" to find k given Q,P
  - known as the elliptic curve logarithm problem
- Certicom example: $E_{23}(9,17)$

ECC Encryption/Decryption

- several alternatives, will consider simplest
- must first encode any message M as a point on the elliptic curve P_m
- select suitable curve & point G as in D-H
- each user chooses private key $n_A<n$
- and computes public key $P_A=n_AG$
- to encrypt P_m : $C_m=\{kG, P_m+kP_A\}$, k random
- decrypt C_m compute:

  $P_m+kP_A-n_A(kG) = P_m+k(n_AG)-n_A(kG) = P_m$

ECC Security

- relies on elliptic curve logarithm problem
- fastest method is "Pollard rho method"
- compared to factoring, can use much smaller key sizes than with RSA etc
- for equivalent key lengths computations are roughly equivalent
- hence for similar security ECC offers significant computational advantages

ELLIPTIC CURVE CRYPTOGRAPHY (ECC)

- ✓ RFC690: Fundamental Elliptic Curve Cryptography Algorithms
- **https://tools.ietf.org/html/rfc6090**

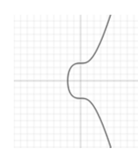
- ✓ FIPS PUB 186-4

- Several discrete logarithm-based protocols have been adapted to elliptic curves (replacing the group)

Comparable Key Sizes for Equivalent Security

Symmetric scheme (key size in bits)	ECC-based scheme (size of *n* in bits)	RSA/DSA (modulus size in bits)
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

ECC - EXAMPLE: BITCOIN



- Secp256k1 (with the ECDSA algorithm)
- Parameters (p,a,b,G,n,h)

- The curve $E: y^2 = x^3+ax+b$ over F_p is defined by:
- a = 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
- b = 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007
- $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$

- The base point G in compressed form is:
- G = 02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798

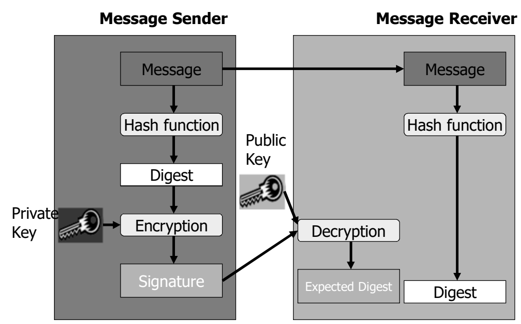
- and in uncompressed form is:
- G = 04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8

- Finally the order n of G and the cofactor are:
- n = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141
- h = 01

STATE OF THE ART

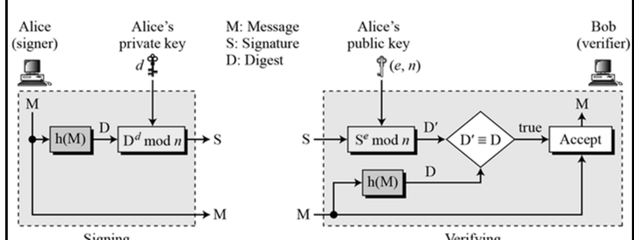
Primitive	Parameters	Legacy System Minimum	Future System Minimum
RSA Problem	N, e, d	$\ell(n) \geq 1024$, $e \geq 3$ or 65537, $d \geq N^{1/2}$	$\ell(n) \geq 3072$ $e \geq 65537$, $d \geq N^{1/2}$
Finite Field DLP	p, q, n	$\ell(p^n) \geq 1024$ $\ell(p), \ell(q) > 160$	$\ell(p^n) \geq 3072$ $\ell(p), \ell(q) > 256$
ECDLP	p, q, n	$\ell(q) \geq 160, \star$	$\ell(q) > 256, \star$
Pairing	p, q, n, d, k	$\ell(p^{k \cdot n}) \geq 1024$ $\ell(p), \ell(q) > 160$	$\ell(p^{k \cdot n}) \geq 3072$ $\ell(p), \ell(q) > 256$

DIGITAL SIGNATURES



Digital signatures

RSA + HASH



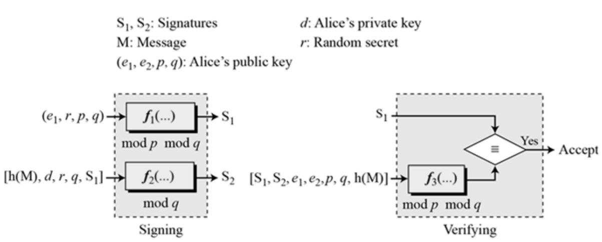
FROM ELGAMAL TO DSA

- The Digital Signature Algorithm (DSA) is a modification of ElGamal digital signature scheme.
- It was proposed in August 1991 and adopted in December 1994 by the National Institute of Standards and Technology.
- Digital Signature Standard (DSS)
  - ✓ Computation of DSS signatures is faster than computation of RSA signatures when using the same p.
  - ✓ DSS signatures are smaller than ElGamal signatures because q is smaller than p.

DIGITAL SIGNATURE

	Classification	
Scheme	Legacy	Future
RSA-PSS	✓	✓
ISO-9796-2 RSA-DS2	✓	✓
PV Signatures	✓	✓
(EC)Schnorr	✓	✓
(EC)KDSA	✓	✓
RSA-PKCS# 1 v1.5	✓	✗
RSA-FDH	✓	✗
ISO-9796-2 RSA-DS3	✓	✗
(EC)DSA,(EC)GDSA	✓	✗
(EC)RDSA	✓	✗
ISO-9796-2 RSA-DS1	✗	✗

DIGITAL SIGNATURE STANDARD (DSS)



STD SOLUTIONS

- **RSA-PKCS# 1 v1.5**

Has no security proof,
Nor any advantages over other RSA
it is widely deployed.
Not propose be used beyond legacy systems.

- **RSA-PSS**

UF-CMA secure in the random oracle model
It is used in a number of places including e-passports.

- **RSA-FDH**

The RSA-FDH scheme hashes the message to the group Z/NZ and then applies the RSA function to the output.
The scheme has strong provable security guarantees
Difficult to defining a suitably strong hash function with codomain the group Z=NZ.
The scheme is not practically deployable.

STD SOLUTIONS

- **ISO 9796-2 RSA Based Mechanisms**

3 different RSA signature padding schemes called Digital Signature 1,Digital Signature 2 and Digital Signature 3 (DS1, DS2 and DS3).
Variant DS1 essentially RSA encrypts a padded version of the message along with a hash of the message.This variant should no longer be considered secure.
Variant DS2 is a standardized version of RSA-PSS, but in a variant which allows partial message recovery.

  - Variant DS3 is defined by taking DS2 and reducing the randomisation parameter to length zero. Not to use for future applications

- (EC)DSA

Widely standardized

- German DSA (GDSA),
- Korean DSA (KDSA)
- Russian DSA (RDSA) [133,162].

All (EC)DSA variants (bar KDSA) have weak provable security guarantees
The KDSA is suitable for future use.

MORE ON SIGNATURES

- *Blind Signatures*
- Sometimes we have a document that we want to get signed without revealing the contents of the document to the signer.
- *Time Stamped Signatures*
- Sometimes a signed document needs to be time stamped to prevent it from being replayed by an adversary. This is called time-stamped digital signature scheme.
- *Group Signatures*
- Protect privacy. Part of a group. Not the same secret key.
- *Proxy Signatures*
- Delegate signature to a server.

STD SOLUTIONS

- **PV Signatures**

ISO 14888-3
A variant of DSA signatures (exactly the same signing equation as for DSA)
Due to Pointcheval and Vaudeney
The PV signature scheme can be shown to be provably secure in the random oracle model
PV signatures suffer from issues related to poor randomness in the ephemeral secret key.

- **(EC)Schnorr**

Like (EC)DSA signatures
Schnorr signatures can be proved UF-CMA secure in the random oracle model [280].
Also a proof in the generic group model
Signature size can be made shorter than that of DSA.
Schnorr signatures are to be preferred over DSA style signatures for future applications.
Defences proposed for (EC)DSA signatures should also be applied to Schnorr signatures

Key agreement

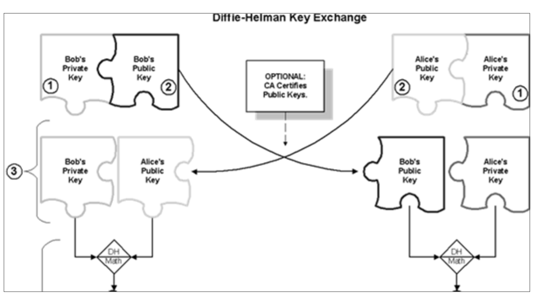
KEY AGREEMENT

- Two entities agree upon a common secret over a public channel
- No pre-shared keys.
- 1976: "New directions in Cryptography"
- Based on the discrete logarithm problem

IMPLEMENTATION

- p and g are both publicly available numbers
- Users pick private values a and b
- Compute public values
- $x = g^a \bmod p$
- $y = g^b \bmod p$
- Public values x and y are exchanged

THE MAIN IDEA - DH



IMPLEMENTATION

- **Compute shared, private key**

$k_a = y^a \bmod p$

$k_b = x^b \bmod p$

- **Algebraically it can be shown that $k_a = k_b$**

Users now have a symmetric secret key to encrypt

TOY EXAMPLE

- **Alice and Bob get public numbers**

$p = 23,\ g = 9$

- **Alice and Bob compute public values**

$X = 9^4 \bmod 23 = 6561 \bmod 23 = 6$
$Y = 9^3 \bmod 23 = 729 \bmod 23 = 16$

- **Alice and Bob exchange public numbers**
- **Alice and Bob compute symmetric keys**

$k_a = y^a \bmod p = 16^4 \bmod 23 = 9$
$k_b = x^b \bmod p = 6^3 \bmod 23 = 9$

- **Alice and Bob now can talk securely!**

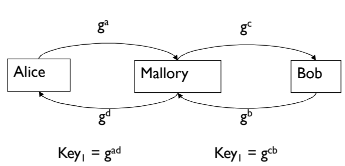
SOLUTION

- AKE protocols (authentication and key establishment protocols)
- Authenticate before key establishment
- Literally hundreds of AKE protocols

- Authentication:
  - Use public key encryption (and usually certificates)
  - Use pre-shared keys (like passwords)

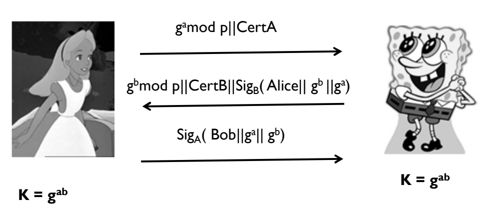
- Two main types of key establishment:
  - Key agreement
  - Key distribution

PERSON-IN-THE-MIDDLE ATTACK



Mallory gets to listen to everything.

AKE BASED ON DH: STATION-TO-STATION PROTOCOL



Key agreement/distribution

Quantum cryptography

QUANTUM KEY DISTRIBUTION

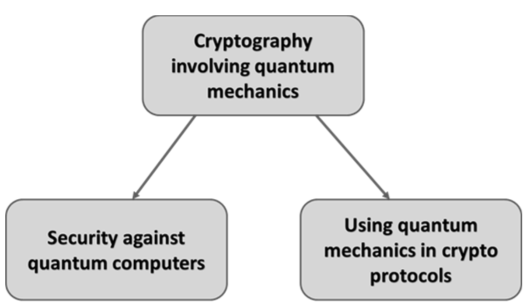
- Symmetric key
- It is based on quantum mechanics
- Two physically separated parties can create and share random secret keys

Allows them to verify that the key has not been intercepted.

- Establish an unconditionally secure communication channel

1. Quantum Key distribution
2. Switch to one-time-pad

QUANTUM CRYPTOGRAPHY



BASIC IDEA

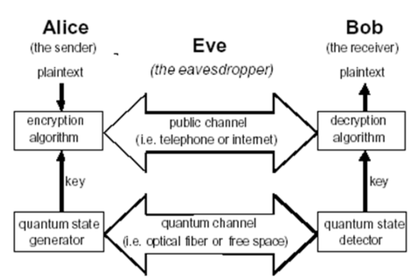
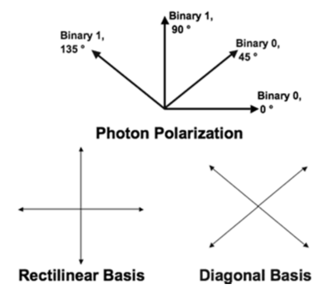


Figure 1. Quantum Key Distribution.

FUNDAMENTALS

- Measurement causes perturbation
- **No Cloning Theorem**
- An unknown quantum state CANNOT be cloned. Therefore, eavesdropper, Eve, cannot have the same information as Bob.
- Single-photon signals are secure.
- Thus, measuring the qubit in the wrong basis destroys the information

TWO BASIS



QUANTUM COMMUNICATIONS

Liner States

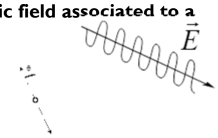
- **Transmitting information with a single-photon**
- **Use a quantum property to carry information**

 = "0" = |0>

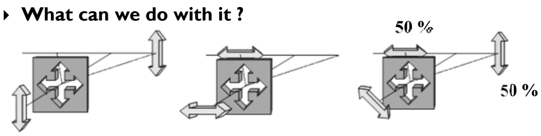
 = "1" = |1>

POLARIZATION OF PHOTONS

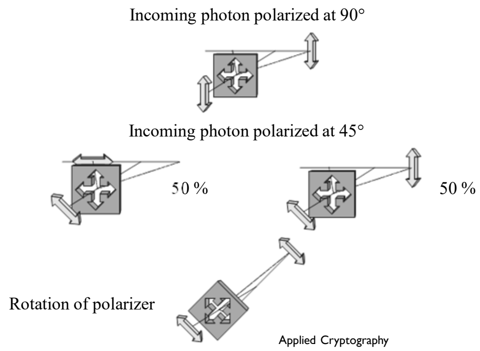
- **Direction of oscillation of the electric field associated to a lightwave**



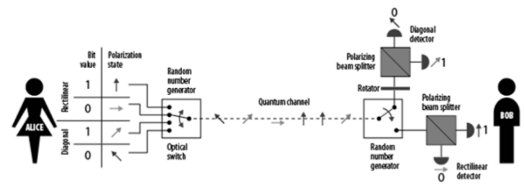
- **Polarization states**
- **What can we do with it ?**



IRREVERSIBILITY OF MEASUREMENTS



BB84



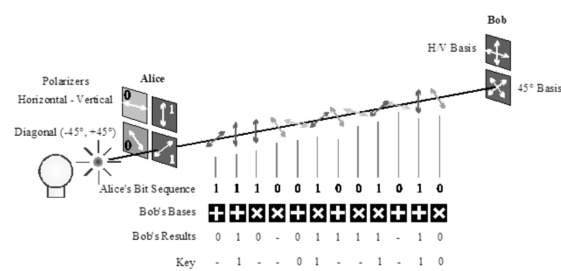
BB84 - SET-UP

- **Paper by Charles Bennett and Gilles Brassard in 1984 is the basis for QKD protocol BB84. Prototype developed in 1991.**

- Alice
- Has the ability to create qubits in two orthogonal bases

- Bob
- Has the ability to measure qubits in those two bases.

BB84



EXAMPLE

Alice's bit	0	1	1	0	1	0	0	1
Alice's basis	+	+	X	+	X	X	X	+
Alice's polarization	↑	→	↖	↑	↖	↗	↗	→
Bob's basis	+	X	X	X	+	X	+	+
Bob's measurement	↑	↗	↖	↗	→	↗	→	→
Public discussion								
Shared Secret key	0		1			0		1

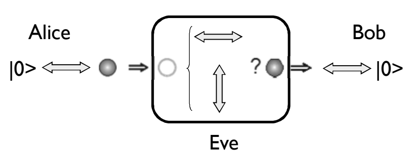
ASSSUMPTIONS

- Source: Emits perfect single photons. (No multi-photons)
- Channel: noisy but lossless. (No absorption in channel)
- Detectors: Perfect detection efficiency. (100 %)
- Basis Alignment: Perfect. (Angle between X and Z basis is exactly 45 degrees.)

- Conclusion: QKD is secure in theory.
- (Assumptions lead to security proofs)

EAVESDROPPING

- Communication interception

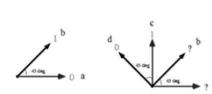


- Use quantum physics to force spy to introduce errors in the communication
- The errors are detected

OTHER SCHEMES

- EPR
- Uses entangled qubits sent from a central source
- Alice and Bob measure qubits randomly and independently
- After measuring, they compare measurement bases and proceed as in BB84
- Advantage over BB84 is that Eve can now be detected using rejected qubits

- B92
- Uses only two non-orthogonal states
- Each bit is either successfully
- received or an "erasure"



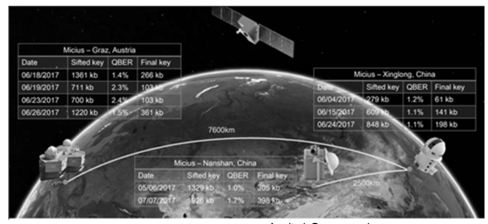
CURRENT STATE OF AFFAIRS

- **Commercial quantum key distribution products exist**

- **Current fiber-based distance**
- **record: 200 km**

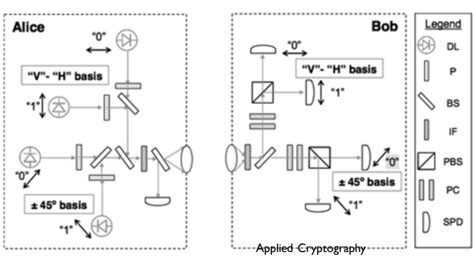
SATELLITE-TO-GROUND QUANTUM KEY DISTRIBUTION

- Micius satellite
- Use QKD and symmetric encyrption
- ESA signed a contract with SES Techcom S.A. (LU) to develop the Quantum Cryptography Telecommunication System (QUARTZ)



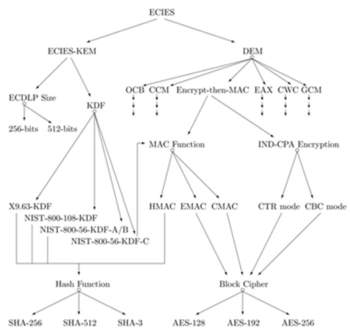
CURRENT STATE OF AFFAIRS

- **Demonstrated free-space link: 10 km**



Public key hybrid schemes

OVERVIEW



*** Algorithms, key size and parameters report. ENISA– 2014**

PROTECTING DATA CONFIDENTIALITY

	Classification	
Scheme	Legacy	Future
RSA-OAEP	✓	✓
RSA-KEM	✓	✓
PSEC-KEM	✓	✓
ECIES-KEM	✓	✓
RSA-PKCS# 1 v1.5	✗	✗

PROTECTING DATA CONFIDENTIALITY

- Public key encryption and decryption are expensive computations.
- It is not secure to encrypt long-plaintext
- Rarely used for plaintext confidentiality protection.

Main schemes used in practice:
1. KEM: Key Encapsulation Mechanism
2. Non-KEM
   - RSA-PKCS# 1 v1.5
   - RSA-OAEP

DEM: Data Encryption Mechanism

NON-KEM

- **RSA-PKCS# 1 v1.5**
- No modern security proof
- Used in SSL/TLS protocol extensively (until v1.2)
- The weak form of padding
- Attacks on various cryptographic devices

- **RSA-OAEP**
- the preferred method of using the RSA primitive to encrypt a small message
- Provably secure in the random oracle model
- The hash functions used can be SHA-1 for legacy applications and SHA-2/SHA-3 for future applications

KEY ENCAPSULATION MECHANISM (KEM)

- Combine a public key encryption with key derivation functions (KDF)

- **RSA-KEM**
- Takes a random element m and encrypts it using the RSA
- The output key is computed by applying a KDF to m
- Secure in the random oracle model

- **PSEC-KEM**
- It is based on elliptic curves.
- Provable secure
- Based on the hardness of the (computational) DH problem
- More secure than ECIES-KEM, less efficient

- **ECIES-KEM**
- Discrete logarithm based encryption scheme
- Very popular

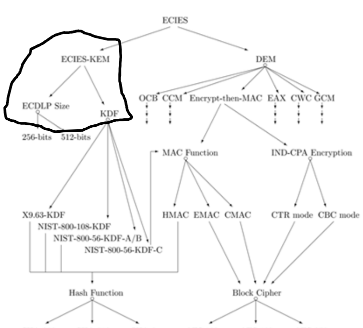
CONTEMPORARY COMMUNICATION PROTOCOL

First Phase: Authentication (sometimes mutual)
- Public Key
- Symmetric Key

Second Phase: Key Establishment
- Key agreement
- Key distribution

Third Phase: Data Encryption
- Symmetric key encryption

OVERVIEW

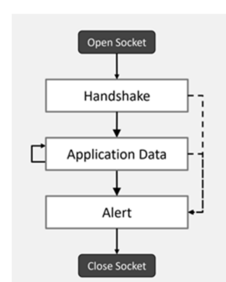


*** Algorithms, key size and parameters report. ENISA– 2014**

TLS 1.3 (EXAMPLE)

- **Handshake**
- – Agree a cipher suite.
- – Agree a master secret.
- – Authentication using certificate(s).

- **Application Data**
- – Symmetric key encryption.
- – AEAD cipher modes.
- – Typically HTTP



(OWASP presentation)

Assistant Professor, Harokopio University of Athens, Greece