CS549: Cryptography and Network Security

© by Xiang-Yang Li

Department of Computer Science, IIT

Notice©

This lecture note (Cryptography and Network Security) is prepared by Xiang-Yang Li. This lecture note has benefited from numerous textbooks and online materials. Especially the "Cryptography and Network Security" 2nd edition by William Stallings and the "Cryptography: Theory and Practice" by Douglas Stinson.

The author has used his best efforts in preparing this lecture note. The author makes no warranty of any kind, expressed or implied, with regard to the programs, protocols contained in this lecture note. The author shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these.

Cryptography and Network Security

Public key system

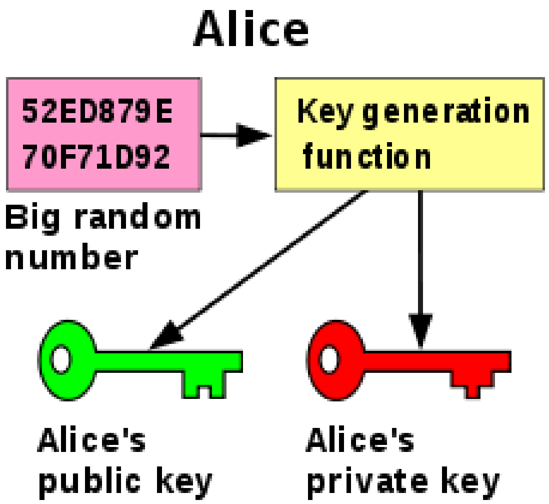
Xiang-Yang Li

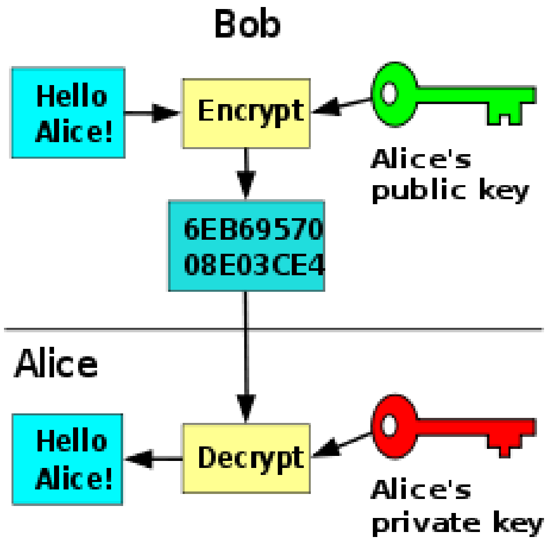
Public Key Encryption

- Two difficult problems
  - Key distribution under conventional encryption
  - Digital signature
- Diffie and Hellman, 1976
  - Astonishing breakthrough
  - One key for encryption and the other related key for decryption
  - It is computationally infeasible (under some assumptions) to determine the decryption key using only the encryption key and the algorithm

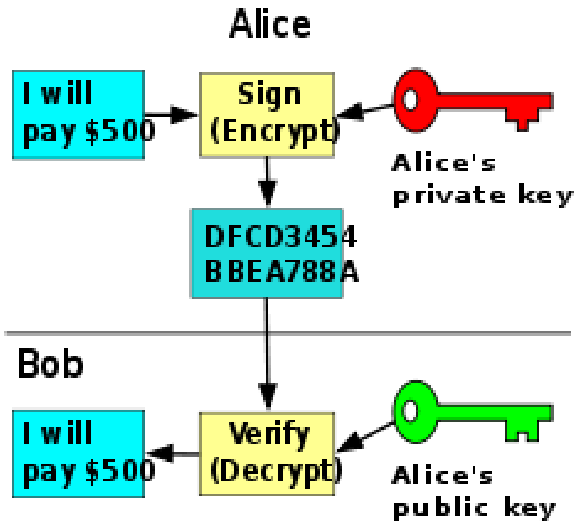
Public Key Cryptosystem

- Essential steps of public key cryptosystem
  - Each end generates a pair of keys
    - One for encryption and one for decryption
  - Each system publishes one key, called public key, and the companion key is kept secret
  - It A wants to send message to B
    - Encrypt it using B's public key
  - When B receives the encrypted message
    - It decrypt it using its own private key



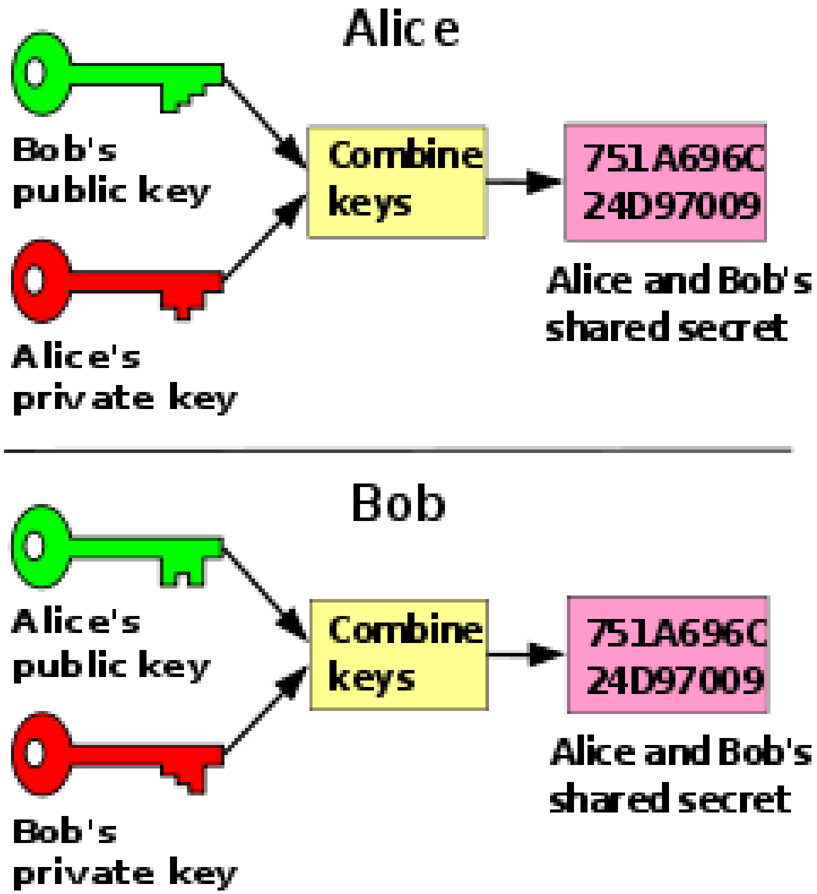


Encryption



Digital signature

Key distribution



Applications of PKC

- Encryption/Decryption
  - The sender encrypts the message using the receiver's public key
    - Q: Why not use the sender's secret key?
- Digital signature
  - The sender signs a message by encrypt the message or a transformation of the message using its own private key
- Key exchange
  - Two sides cooperate to exchange a session key, typically for conventional encryption

Conditions of PKC

- Computationally easy
  - To generate public and private key pair
  - To encrypt the message using encryption key
  - To decrypt the message using decryption key
- Computational infeasible
  - To compute the private key using public key
  - To recover the plaintext using ciphertext and public key
- The encryption and decryption can be applied in either order

One Way Function

- PKC boils down to one way function
  - Maps a domain into a range with unique inverse
  - The calculation of the function is easy
  - The calculation of the inverse is infeasible
- ***Easy***
  - The problem can be solved in polynomial time
- **Infeasible**
  - The effort to solve it grows faster than polynomial time
  - For example: 2^n
  - It requires infeasible for all inputs, not just worst case

Trapdoor One-way Function

- Trapdoor one way function
  - Maps a domain into a range with unique inverse
    - $Y=f_k(X)$
  - The calculation of the function is easy
  - The calculation of the inverse is infeasible if the key is not known
  - The calculation of the inverse is easy if the key is known

Possible Attacks

- Brute force
  - Use large keys
    - Trade-off: speed (not linearly depend on key size)
    - Confined to small data encryption: signature, key management
- Compute the private key from public key
  - Not proven that is not feasible for most protocols!
- Probable message attack
  - Encrypt all possible messages using encryption key
  - Compare with the ciphertext to find the matched one!
  - If data is small, feasible, regardless of key size of PKC

History

- In 1874, a book by William Stanley Jevons [1] described the relationship of one-way functions to cryptography and went on to discuss specifically the factorization problem used to create the trapdoor function in the RSA system.

History

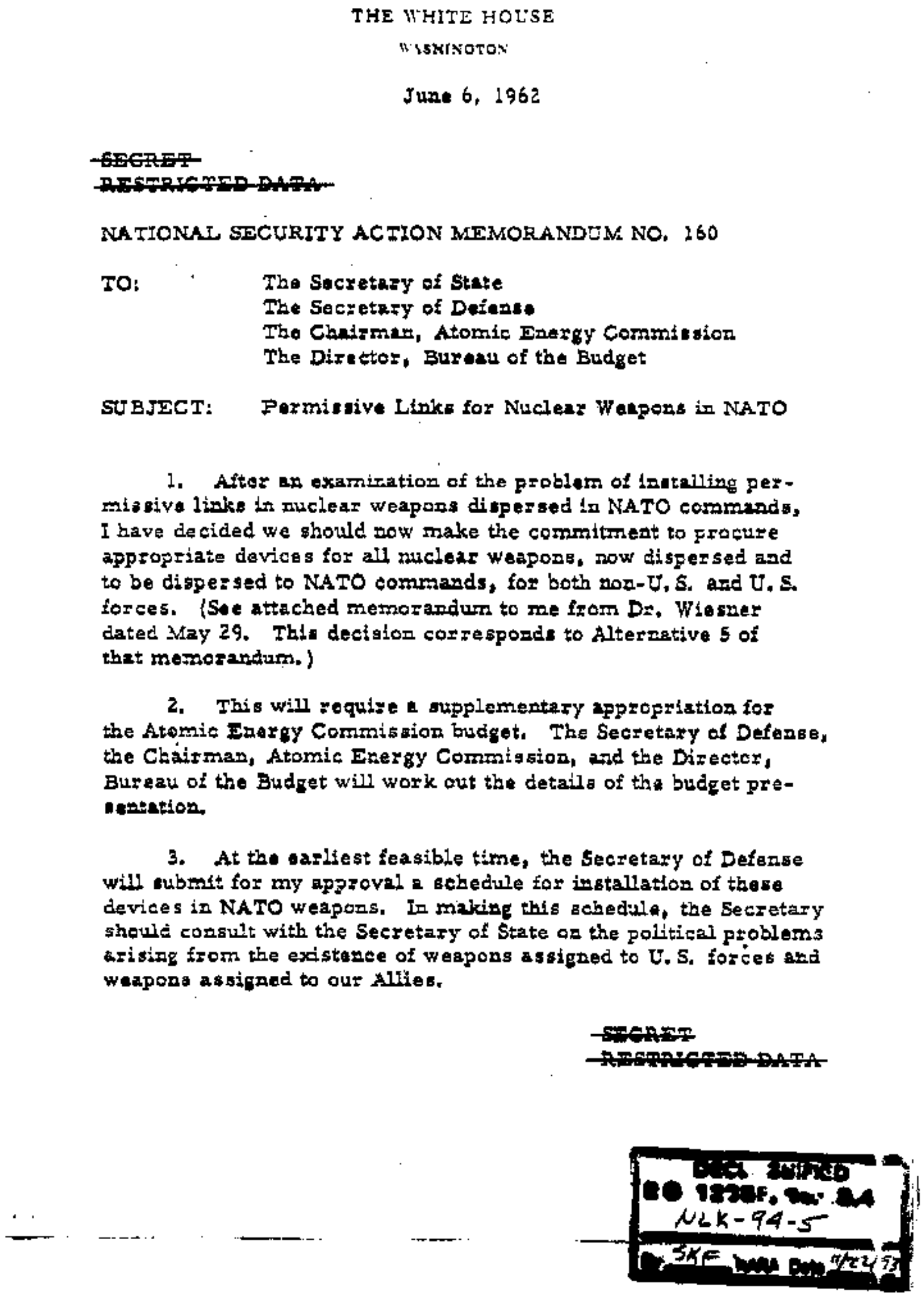
- 1976, Diffie-Hellman protocol was the first published practical method for establishing a shared secret-key over an authenticated (but not private) communications channel without using a prior shared secret.

- Merkle's public-key-agreement technique became known as Merkle's Puzzles, and was invented in 1974 and published in 1978.

- RSA invented in 1977, and published 1978
- ElGamal system 1984
- **Elliptic curve cryptography (ECC)** , Neal Koblitz[1] and Victor S. Miller[2] in 1985.
- Digital Signature Algorithm (DSA) 1991-1993

History

- http://www.research.att.com/~smb/nsam-160/ British
- **National Security Action Memorandum 160**
  - Kennedy Nuclear Weapon
  - http://www.research.att.com/~smb/nsam-160/pg1.html
    - National Security Action Memorandum 160 (from June 6, 1962), entitled "Permissive Links for Nuclear Weapons in NATO". The claim was that this memo -- signed by President Kennedy and endorsing a memo from his science advisor, **Jerome Weisner** -- was the basis for the invention of public key cryptography by NSA.



THE WHITE HOUSE

WASHINGTON

June 6, 1962

~~SECRET~~
~~RESTRICTED DATA~~

NATIONAL SECURITY ACTION MEMORANDUM NO. 160

TO: The Secretary of State
The Secretary of Defense
The Chairman, Atomic Energy Commission
The Director, Bureau of the Budget

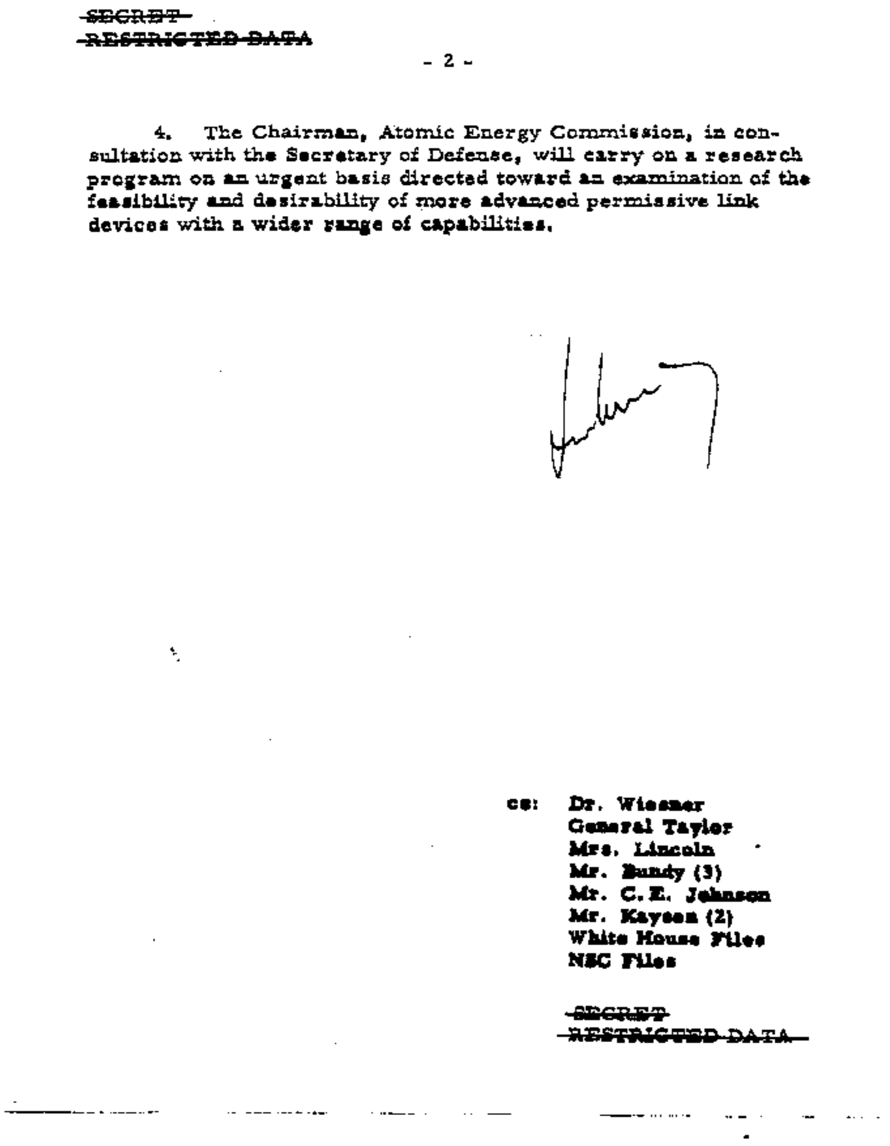
SUBJECT: Permissive Links for Nuclear Weapons in NATO

1. After an examination of the problem of installing permissive links in nuclear weapons dispersed in NATO commands, I have decided we should now make the commitment to procure appropriate devices for all nuclear weapons, now dispersed and to be dispersed to NATO commands, for both non-U.S. and U.S. forces. (See attached memorandum to me from Dr. Wiesner dated May 29. This decision corresponds to Alternative 5 of that memorandum.)

2. This will require a supplementary appropriation for the Atomic Energy Commission budget. The Secretary of Defense, the Chairman, Atomic Energy Commission, and the Director, Bureau of the Budget will work out the details of the budget presentation.

3. At the earliest feasible time, the Secretary of Defense will submit for my approval a schedule for installation of these devices in NATO weapons. In making this schedule, the Secretary should consult with the Secretary of State on the political problems arising from the existence of weapons assigned to U.S. forces and weapons assigned to our Allies.

~~SECRET~~
~~RESTRICTED DATA~~



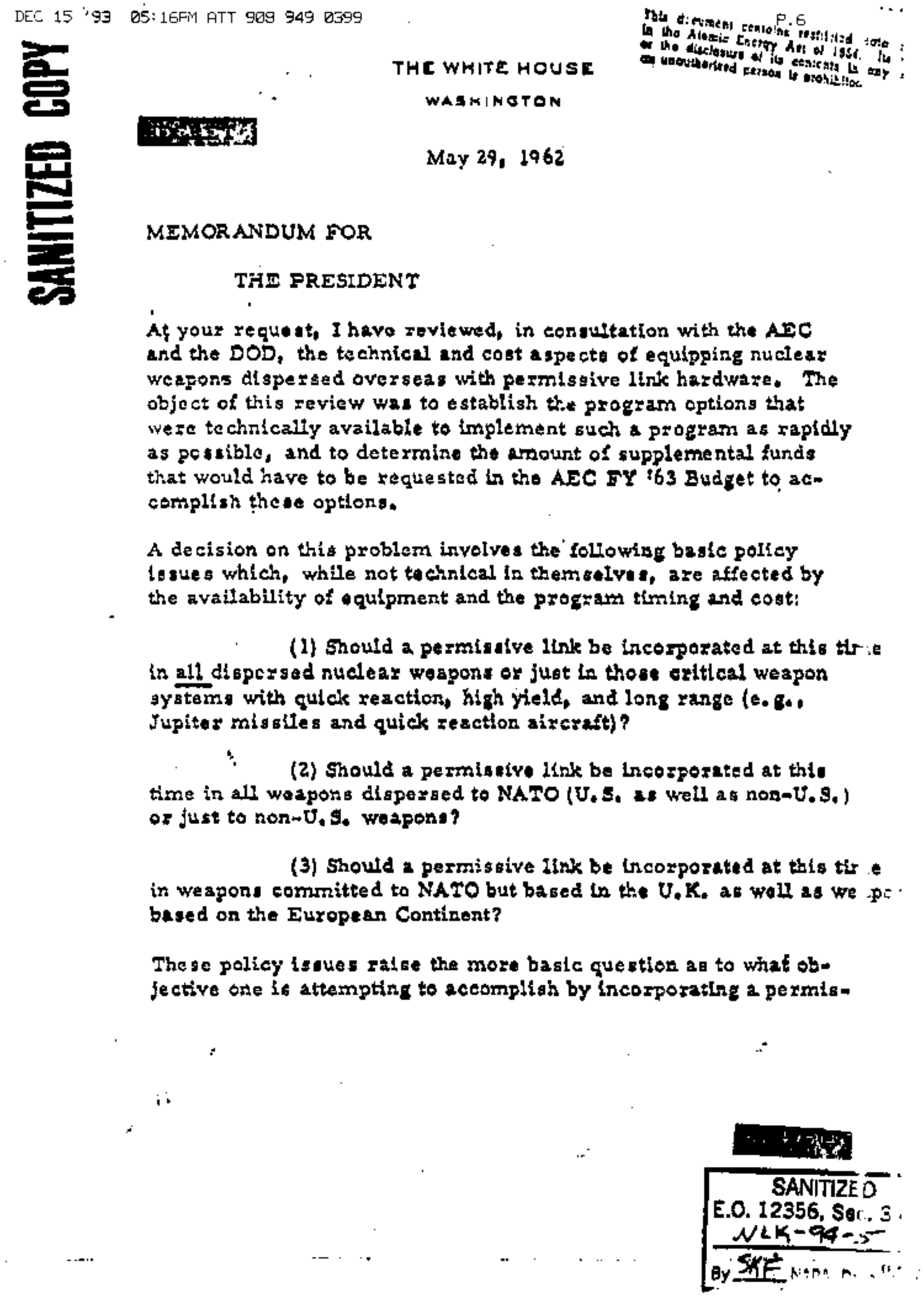
~~SECRET~~
~~RESTRICTED DATA~~

- 2 -

4. The Chairman, Atomic Energy Commission, in consultation with the Secretary of Defense, will carry on a research program on an urgent basis directed toward an examination of the feasibility and desirability of more advanced permissive link devices with a wider range of capabilities.

cc: Dr. Wiesner
General Taylor
Mrs. Lincoln
Mr. Bundy (3)
Mr. C. E. Johnson
Mr. Kaysen (2)
White House Files
NSC Files

~~SECRET~~
~~RESTRICTED DATA~~



DEC 15 '93 05:16PM ATT 908 949 0399

P.6

This document contains restricted data
in the Atomic Energy Act of 1954. Its
or the disclosure of its contents in any
an unauthorized person is prohibited.

SANITIZED COPY

THE WHITE HOUSE

WASHINGTON

[redacted]

May 29, 1962

MEMORANDUM FOR

THE PRESIDENT

At your request, I have reviewed, in consultation with the AEC and the DOD, the technical and cost aspects of equipping nuclear weapons dispersed overseas with permissive link hardware. The object of this review was to establish the program options that were technically available to implement such a program as rapidly as possible, and to determine the amount of supplemental funds that would have to be requested in the AEC FY '63 Budget to accomplish these options.

A decision on this problem involves the following basic policy issues which, while not technical in themselves, are affected by the availability of equipment and the program timing and cost:

(1) Should a permissive link be incorporated at this time in all dispersed nuclear weapons or just in those critical weapon systems with quick reaction, high yield, and long range (e.g., Jupiter missiles and quick reaction aircraft)?

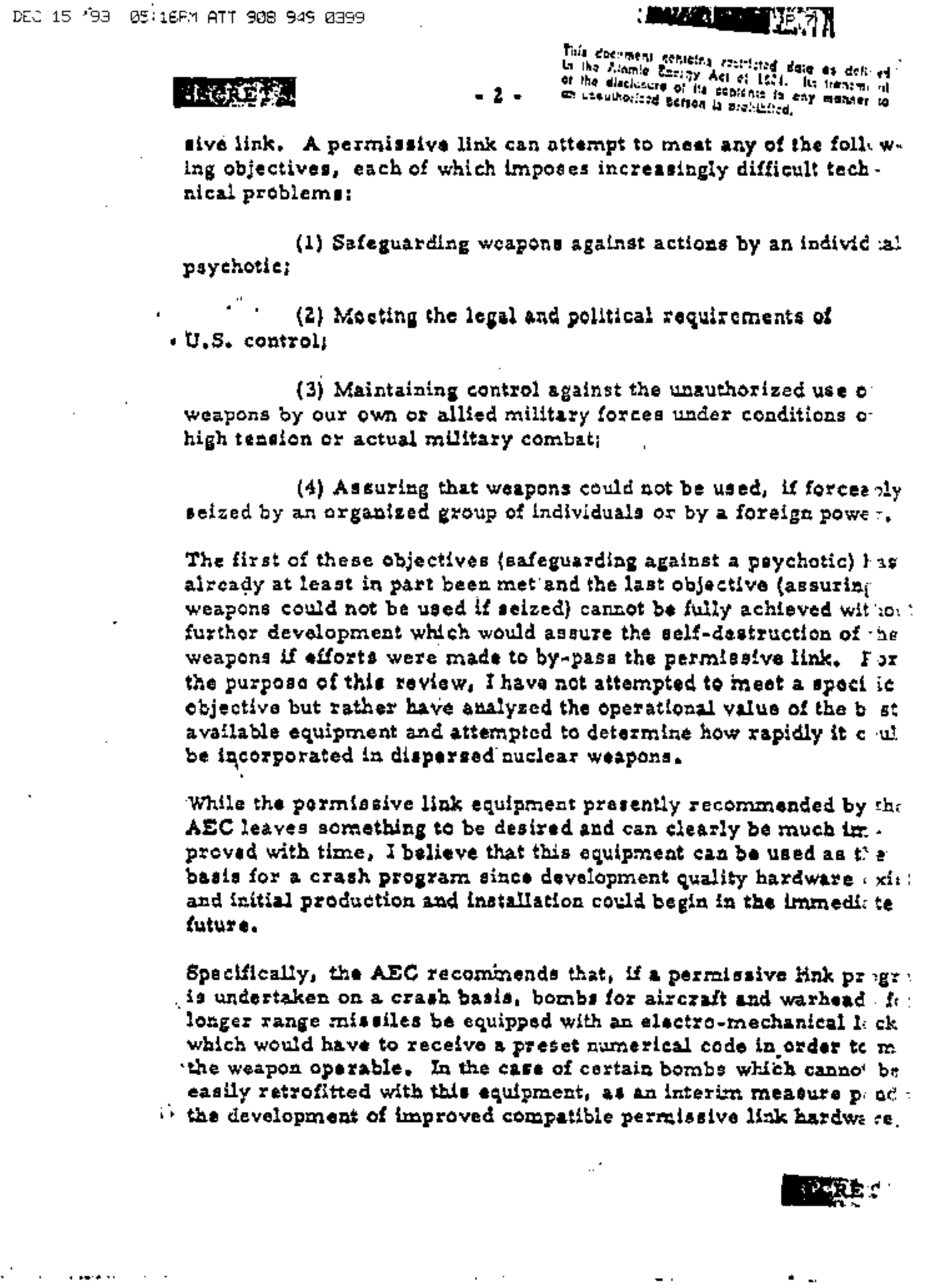
(2) Should a permissive link be incorporated at this time in all weapons dispersed to NATO (U.S. as well as non-U.S.) or just to non-U.S. weapons?

(3) Should a permissive link be incorporated at this time in weapons committed to NATO but based in the U.K. as well as weapons based on the European Continent?

These policy issues raise the more basic question as to what objective one is attempting to accomplish by incorporating a permis-

[redacted]

SANITIZED
E.O. 12356, Sec. 3
NLK-94-5
By SKF NARA



DEC 15 '93 05:16PM ATT 908 949 0399

[redacted]

This document contains restricted data as defined in the Atomic Energy Act of 1954. Its transmittal or the disclosure of its contents in any manner to an unauthorized person is prohibited.

SECRET - 2 -

sive link. A permissive link can attempt to meet any of the following objectives, each of which imposes increasingly difficult technical problems:

(1) Safeguarding weapons against actions by an individual psychotic;

(2) Meeting the legal and political requirements of U.S. control;

(3) Maintaining control against the unauthorized use of weapons by our own or allied military forces under conditions of high tension or actual military combat;

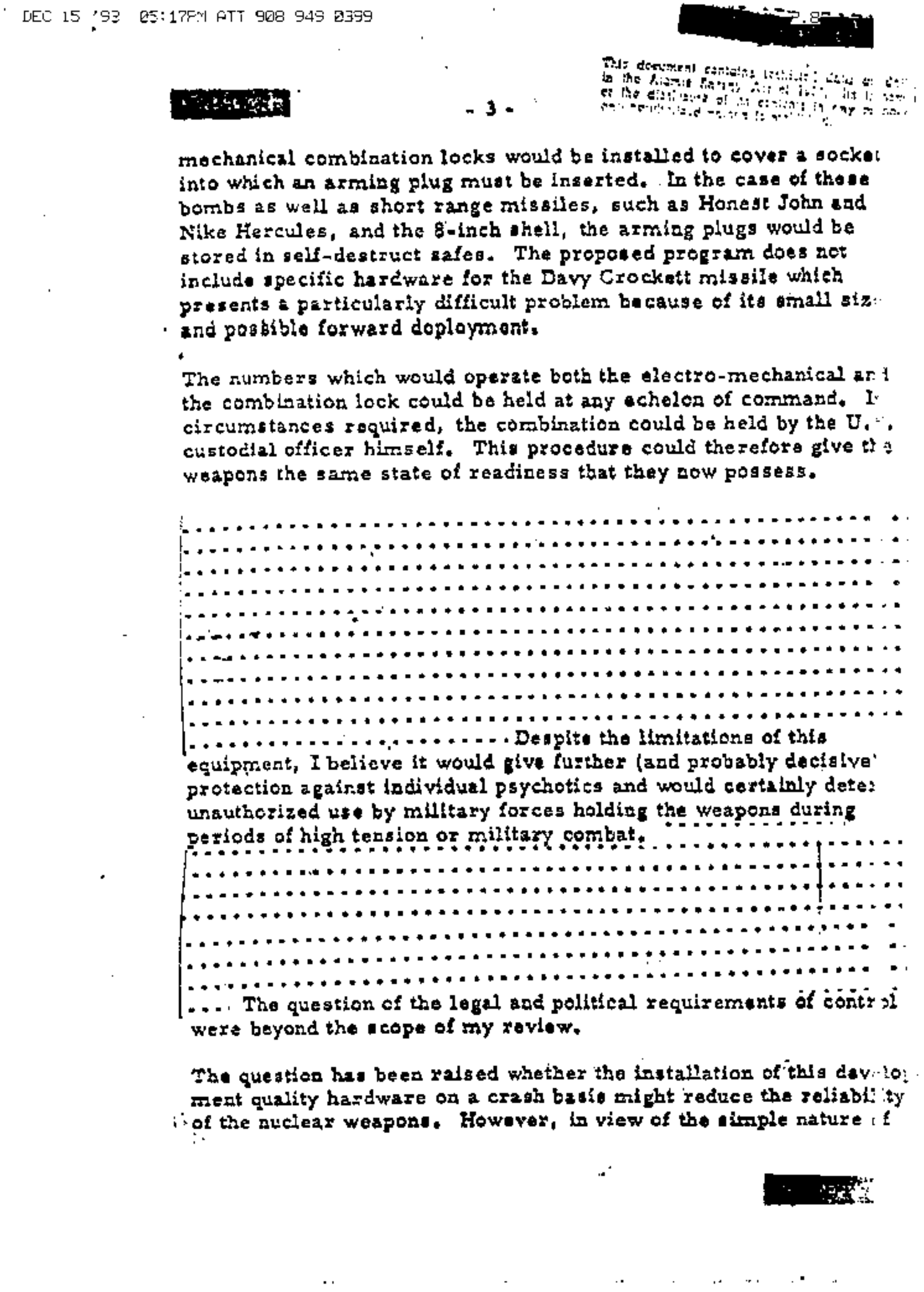
(4) Assuring that weapons could not be used, if forceably seized by an organized group of individuals or by a foreign power.

The first of these objectives (safeguarding against a psychotic) has already at least in part been met and the last objective (assuring weapons could not be used if seized) cannot be fully achieved without further development which would assure the self-destruction of the weapons if efforts were made to by-pass the permissive link. For the purpose of this review, I have not attempted to meet a specific objective but rather have analyzed the operational value of the best available equipment and attempted to determine how rapidly it could be incorporated in dispersed nuclear weapons.

While the permissive link equipment presently recommended by the AEC leaves something to be desired and can clearly be much improved with time, I believe that this equipment can be used as the basis for a crash program since development quality hardware exists and initial production and installation could begin in the immediate future.

Specifically, the AEC recommends that, if a permissive link program is undertaken on a crash basis, bombs for aircraft and warheads for longer range missiles be equipped with an electro-mechanical lock which would have to receive a preset numerical code in order to make the weapon operable. In the case of certain bombs which cannot be easily retrofitted with this equipment, as an interim measure pending the development of improved compatible permissive link hardware,

SECRET



DEC 15 '93 05:17PM ATT 908 949 0399

[redacted]

This document contains [illegible] in the Atomic Energy Act of [illegible] or the disclosure of its contents in any manner [illegible]

[redacted] - 3 -

mechanical combination locks would be installed to cover a socket into which an arming plug must be inserted. In the case of these bombs as well as short range missiles, such as Honest John and Nike Hercules, and the 8-inch shell, the arming plugs would be stored in self-destruct safes. The proposed program does not include specific hardware for the Davy Crockett missile which presents a particularly difficult problem because of its small size and possible forward deployment.

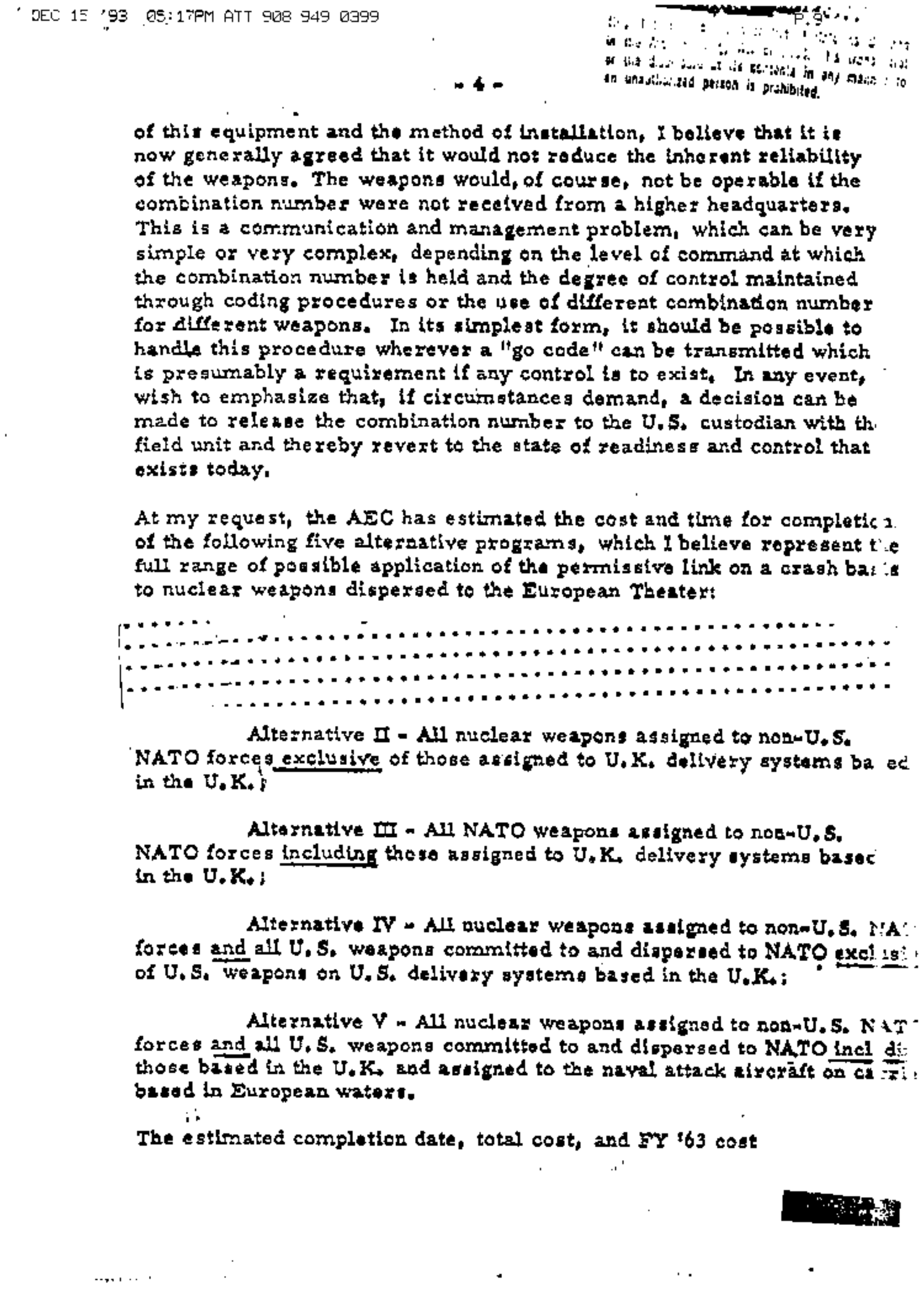
The numbers which would operate both the electro-mechanical and the combination lock could be held at any echelon of command. If circumstances required, the combination could be held by the U.S. custodial officer himself. This procedure could therefore give the weapons the same state of readiness that they now possess.

Despite the limitations of this equipment, I believe it would give further (and probably decisive) protection against individual psychotics and would certainly deter unauthorized use by military forces holding the weapons during periods of high tension or military combat.

The question of the legal and political requirements of control were beyond the scope of my review.

The question has been raised whether the installation of this development quality hardware on a crash basis might reduce the reliability of the nuclear weapons. However, in view of the simple nature of

[redacted]



DEC 15 '93 05:17PM ATT 908 949 0399 P.9

[illegible] in the [illegible] of the [illegible] of its contents in any manner to an unauthorized person is prohibited.

- 4 -

of this equipment and the method of installation, I believe that it is now generally agreed that it would not reduce the inherent reliability of the weapons. The weapons would, of course, not be operable if the combination number were not received from a higher headquarters. This is a communication and management problem, which can be very simple or very complex, depending on the level of command at which the combination number is held and the degree of control maintained through coding procedures or the use of different combination number for different weapons. In its simplest form, it should be possible to handle this procedure wherever a "go code" can be transmitted which is presumably a requirement if any control is to exist. In any event, wish to emphasize that, if circumstances demand, a decision can be made to release the combination number to the U.S. custodian with th field unit and thereby revert to the state of readiness and control that exists today.

At my request, the AEC has estimated the cost and time for completic of the following five alternative programs, which I believe represent t e full range of possible application of the permissive link on a crash ba is to nuclear weapons dispersed to the European Theater:

[redacted]

Alternative II - All nuclear weapons assigned to non-U.S. NATO forces exclusive of those assigned to U.K. delivery systems ba ed in the U.K.;

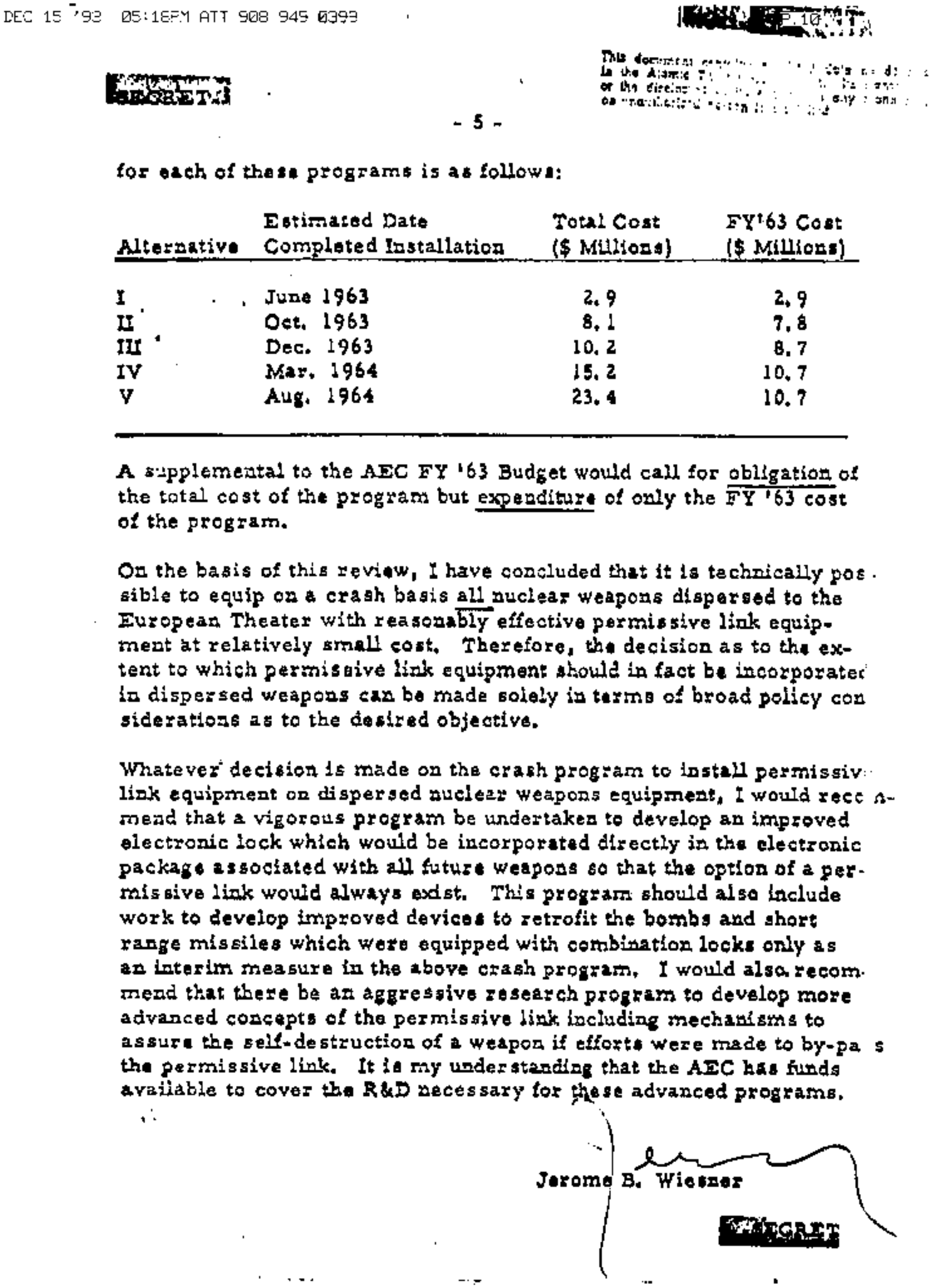
Alternative III - All NATO weapons assigned to non-U.S. NATO forces including those assigned to U.K. delivery systems basec in the U.K.;

Alternative IV - All nuclear weapons assigned to non-U.S. NA forces and all U.S. weapons committed to and dispersed to NATO exclusi of U.S. weapons on U.S. delivery systems based in the U.K.;

Alternative V - All nuclear weapons assigned to non-U.S. NAT forces and all U.S. weapons committed to and dispersed to NATO incl di those based in the U.K. and assigned to the naval attack aircraft on ca ri based in European waters.

The estimated completion date, total cost, and FY '63 cost

[redacted]



DEC 15 '93 05:16PM ATT 908 945 0399 [redacted] P.10

SECRET

This document ... in the Atomic ... or the ... [illegible]

- 5 -

for each of these programs is as follows:

Alternative	Estimated Date Completed Installation	Total Cost ($ Millions)	FY'63 Cost ($ Millions)
I	June 1963	2.9	2.9
II	Oct. 1963	8.1	7.8
III	Dec. 1963	10.2	8.7
IV	Mar. 1964	15.2	10.7
V	Aug. 1964	23.4	10.7

A supplemental to the AEC FY '63 Budget would call for obligation of the total cost of the program but expenditure of only the FY '63 cost of the program.

On the basis of this review, I have concluded that it is technically possible to equip on a crash basis all nuclear weapons dispersed to the European Theater with reasonably effective permissive link equipment at relatively small cost. Therefore, the decision as to the extent to which permissive link equipment should in fact be incorporated in dispersed weapons can be made solely in terms of broad policy considerations as to the desired objective.

Whatever decision is made on the crash program to install permissive link equipment on dispersed nuclear weapons equipment, I would recommend that a vigorous program be undertaken to develop an improved electronic lock which would be incorporated directly in the electronic package associated with all future weapons so that the option of a permissive link would always exist. This program should also include work to develop improved devices to retrofit the bombs and short range missiles which were equipped with combination locks only as an interim measure in the above crash program. I would also recommend that there be an aggressive research program to develop more advanced concepts of the permissive link including mechanisms to assure the self-destruction of a weapon if efforts were made to by-pass the permissive link. It is my understanding that the AEC has funds available to cover the R&D necessary for these advanced programs.

Jerome B. Wiesner

SECRET

RSA Algorithm

- R. Rivest, A. Shamir, L. Adleman (1977)
  - James Ellis came up with the idea in 1970, and proved that it was theoretically possible. In 1973, Clifford Cocks a British mathematician invented a variant on RSA; a few months later, Malcom Williamson invented a Diffie-Hellman analog
  - Only revealed till 1997
- Patent expired on September 20, 2000.
- Block cipher using integers *0~n-1*
  - Thus block size k is less than $log_2 n$
- Algorithm:
  - Encryption: $C=M^e \ mod \ n$
  - Decryption: $M=C^d \ mod \ n$
- Both sender and the receiver know *n*

RSA (public key encryption)

- Alice wants Bob to send her a message. She:
- selects two (large) primes p, q, TOP SECRET,
- computes $n = pq$ and $\phi(n) = (p\text{-}1)(q\text{-}1)$, $\phi(n)$ also TOP SECRET,
- selects an integer e, $1 < e < \phi(n)$, such that $\gcd(e, \phi(n)) = 1$,
- computes d, such that $de \equiv 1 \pmod{\phi(n)}$, d also TOP SECRET,
- gives **public** key (e, n), keeps **private** key (d, n).

Requirements

- Possible to find *e* and *d* such that
  - $M=M^{de} \bmod n$ for all message M
- Easy to conduct encryption and decryption
- Infeasible to compute *d*
  - Given *n* and *e*

RSA Example

1. Select primes: p=17 & q=11
2. Compute $n = pq$ =17×11=187
3. Compute ø(n)=(p-1)(q-1)=16×10=160
4. Select e : gcd(e,160)=1; choose e=7
5. Determine d: de=1 mod 160 and d < 160 Value is d=23 since 23×7=161= 10×160+1
6. Publish public key KU={7,187}
7. Keep secret private key KR={23,17,11}

RSA Example cont

- sample RSA encryption/decryption is:
- given message `M = 88` (nb. `88<187`)
- encryption:

  $C = 88^7 \bmod 187 = 11$

- decryption:

  $M = 11^{23} \bmod 187 = 88$

Key Generation

- Recall Euler Theorem
  - $a^{\phi(n)+1} = a \bmod n$ for all $0<a<n$ and $gcd(a,n)=1$
  - Then $ed=1 \bmod \phi(n)$ is sufficient to make algorithm correct (need more proofs)
- RSA chooses the following
  - Integer $n=pq$ for two primes p and q
  - Select e, such that $\gcd(e, \phi(n))=1$
  - Compute the inverse of e mod $\phi(n)$
    - The result is set as *d*

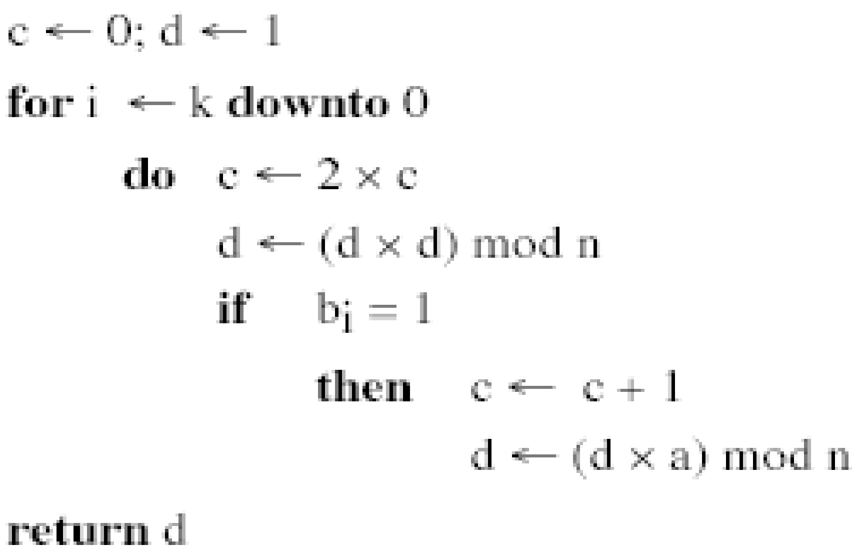
Key Generation

- The prime numbers *p* and *q* must be sufficiently large
  - They are chosen by applying primality testing of randomly chosen large numbers
  - About *n/ln n* prime numbers less than *n*
    - Implies needs to check about *2/ln n* random numbers to find 2 primes numbers around *n*
    - Compute *n=pq,* keep *p* and *q* secret!
- Select random number *e*
  - Test gcd(*e*, ϕ(*n*))=1, and get *d* if equation holds

Exponentiation

- can use the Square and Multiply Algorithm
- a fast, efficient algorithm for exponentiation
- concept is based on repeatedly squaring base
- and multiplying in the ones that are needed to compute the result
- look at binary representation of exponent
- only takes O($\log_2$ n) multiples for number n
  - eg. $7^5 = 7^4.7^1 = 3.7 = 10 \mod 11$
  - eg. $3^{129} = 3^{128}.3^1 = 5.3 = 4 \mod 11$

Exponentiation



```
$c \leftarrow 0$; $d \leftarrow 1$
for $i \leftarrow k$ downto 0
        do   $c \leftarrow 2 \times c$
             $d \leftarrow (d \times d) \bmod n$
             if   $b_i = 1$
                  then   $c \leftarrow c + 1$
                         $d \leftarrow (d \times a) \bmod n$
return d
```

More on Exponention (PGP)

- To compute C^d mod n, we compute
  - C^d mod p and
  - C^d mod q
- Remember that the receiver could keep p,q
- Then Chinese Remainder Theorem to find
  - C^d mod n
  - Here we use the fact that the following equation has one unique solution in the range of [1, pq]

$$\begin{cases} x = a \quad \bmod p \\ x = b \quad \bmod q \end{cases}$$

Security of RSA

- Brute force: try all possible private keys
- Factoring integer *n*, then know $\phi(n)$
  - Not proven to be NPC
- Determine $\phi(n)$ directly without factoring
  - Equivalent to factoring! (1996)
  - Two variables and two equations

$$\begin{cases} \phi(n) = (p-1)(q-1) \\ n = pq \end{cases}$$

- Determine *d* directly without knowing $\phi(n)$
  - Currently appears as hard as factoring
    - But not proven, so it may be easier!

Practical Considerations

- Testing p, q using probability first, then deterministic methods
- A good random number generator is needed for p,q
  - 'random' ***and*** 'unpredictable'
- Primes *p* and *q* should be in similar scale
- Both *p-1* and *q-1* should have large prime factor
- The gcd(p-1,q-1) should be small
- The encryption key *e* = 2 should not be used
- The decryption key *d* should larger than $n^{1/4}$
- RSA is much slower than symmetric cryptosystems.
  - In practice, typically encrypts a secret message with a symmetric algorithm, encrypts the (comparatively short) symmetric key with RSA, and transmits both the RSA-encrypted symmetric key and the symmetrically-encrypted message to Alice.

Fixed point of RSA

- How many m such that
  - $m^e=m \bmod n$ assume that gcd(m, n)=1
  - It is same as $m^{e-1}=1 \bmod n$
  - Thus, $m^{e-1}=1 \bmod p$ and $m^{e-1}=1 \bmod q$
  - Solutions gcd(e-1,p-1)*gcd(e-1,q-1)
    - Need more proofs.

continue

- Solving $m^{e-1}=1 \bmod p$
- Each number in $[1, p-1]$ is written as

  $g^i \bmod p$ where i is an integer in $[1, p-1]$ and g is primitive root

  Then $m^{e-1} = 1 \bmod p$ is same as finding i such that

  $(g^i)^{e-1} = 1 \bmod p$

  Since g is primitive root, we have

  $p-1$ divides $i(e-1)$

  Let $d = \gcd(p-1, i(e-1))$

  Then $\frac{p-1}{\mathrm{d}}$ divides $\frac{i(e-1)}{\mathrm{d}}$

  Since $\gcd(\frac{p-1}{\mathrm{d}}, \frac{e-1}{\mathrm{d}}) = 1$, we have $\frac{p-1}{\mathrm{d}}$ divides i.

  Thus, the choices for i is $\frac{p-1}{\mathrm{d}} j$, where $j = 1,2,\ldots d$.

Cyclic Attack

- Continuously re-encrypt the ciphertext
- Compute c^e mod n, $m^{e^2} = (m^e)^e$mod n, m^{e^3} mod n…till it reaches same C, then the previous one is m
- Need period large
- Let r be the largest prime of p-1, L be the largest prime of r-1
- Then period is at least L with high probability
  - Implies that we often need find a large prime x
  - Based on this, find a large prime of y=kx+1 format (by trying k=2,3,…)
  - Based on y, then find a large prime p=t y+1 format
    - Try difference values for t=2,3,4…

Avoid Cyclic attack

- **Strong Primes**
- It has been suggested that choosing "Strong Primes" for p and q increases the number of cycles required to break the encryption:
- p is a strong prime if p-1 and p+1 both have large factors, t and w
  - t-1 and t+1 have a large factor
  - w-1 and w+1 have a large factor
- In analysis with small key size (10-70 bit) we saw no correlation between key size and cycles required to crack.
- ●**Large Primes**
  - In tests, keys greater than 60 bits were not crackable using a cyclic attack in 24 hours
  - As RSA currently uses keys that are 1024 bits or higher, it would take many years to crack
  - This type of attack is not deemed feasible with current hardware

How to deal with p, q

- Delete them securely
- Or used for speed-up calculation from CRT
  - Compute $M^e \mathit{mod}\ p$ and $M^e \mathit{mod}\ q$
  - Then find using $M^e \mathit{mod}\ n$ based on CRT

Timing Attacks

- Keep track of how long a computer takes to decrypt a message!
  - Paul Kocher, 1995, Dec-7
  - Stunning attack strategy and cipher only attack!
  - Guessing the key bit by bit
- Countermeasures (Rivest `11 Dec 1995`)
  - Constant exponentiation time
  - Random delay
  - Blinding (add a random number for encryption and decryption)

Attack when same keys for Encryption and Signature

- Collect ciphertext c (send to Alice), want to find m=c^d mod n
- Attacker chooses random r
- Compute x= r^e mod n; y=xc mod n; and t= r^{-1} mod n
- Attacker gets Alice to sign y with private key using RSA: y^d mod n
  - That is why not use the same key for encryption and digital signature
- Alice sends u= y^d mod n to Attacker
- Attacker then computes tu mod n→m

Other attacks on RSA

- Comprised decryption key
  - If the private key d (for decryption of received ciphertext) of a user is comprised, then the user has to reselect n and e and d
  - It cannot use the old number n to produce the key-pairs!
  - Otherwise attacker already can factor n almost surely!
- The number n can only be used by one person
  - If two user uses the same n, even they do not know the factoring of n, they still could figure out the factoring of n with probability almost one.
    - Similar as above

Factoring algorithm given the decryption key d

```
Algorithm $Factor(n,e,d)$ // given n, e, and d, find a factor of n

Choose a random w in [1, n - 1]
compute $x = \gcd(w,n)$
if $1 < x < n$ then output x as a factor, and quit
write $ed-1 = 2^s r$, where r is odd
compute $v = w^r \mod n$
If $v = 1 \mod n$, then quit and failed
while $v \neq 1 \mod n$ do
    $v_0 = v;$
    $v = v^2 \mod n$
If $v_0 = -1 \mod n$ then quit and faild
else compute $x = \gcd(v_0 + 1, n)$ as one factor of n
```

Bit security of RSA

- Given ciphertext C,
  - We may want to find the last bit of M, denoted by parity(C)
  - We may want to find if M>n/2, denoted by half(C)
  - We may want to find all bits of M
- The above three attacks are the same!
  - If we can solve one, we can solve the other two!

Bit security of RSA

1. The following two are equivalent
   1. Find the last bit of M, denoted by parity(C)
   2. Find if M>n/2, denoted by half(C)

$C = M^e \bmod n$

Then the encrytion of 2M is $(2M)^e \bmod n = 2^e M^e \bmod n = 2^e C \bmod n$

If $M < n/2$, then $2M < n$ and $(2M \bmod n)$ is an even number

If $M > n/2$, then $2M > n$ and $(2M \bmod n = 2M - n)$ is an odd number

Thus, *when* $M < n/2$, the last bit of the message returned by parity$(2^e C)$ will be 0;

when $M > n/2$, the last bit of the message returned by parity$(2^e C)$ will be 1.

Thus, if we can solve parity(C), then we can use parity$(2^e C)$ to answer whether $M > n/2$ or not.

Then, consider the other direction. Assume that wecan solve half(C).

Then we use the following property

1)If M is an even number, then it is easy toshow that $\frac{\mathrm{M}}{2} = M * 2^{-1} \bmod n$

In other words, $M * 2^{-1} \bmod n$ is an integer $< n/2$ when M is even.

2) When M is an odd number, assume that $M * 2^{-1} \equiv x \bmod n$.

Then we have $2 * M * 2^{-1} \equiv 2x \bmod n, \Rightarrow M \equiv 2x \bmod n$

Since M is an odd number, obviously, $x > n/2$.

Thus, $M * 2^{-1} \bmod n$ is an integer $> n/2$ when M is odd

Notice that the encrytion of $M * 2^{-1} \bmod n$ is $C*(2^{-1} \bmod n) \bmod n$

Thus, half$(C*(2^{-1} \bmod n) \bmod n)$ returns "message is > n/2" $\Rightarrow M$ is odd

half$(C*(2^{-1} \bmod n) \bmod n)$ returns "message is < n/2" $\Rightarrow M$ is even

Bit security of RSA

1. The following two are equivalent
   1. Find the last bit of M, denoted by parity(C)
   2. Find all bits of M

$C = M^e \bmod n$

Then the encrytion of 2M is $(2M)^e \bmod n = 2^e M^e \bmod n = 2^e C \bmod n$

If $M < n/2$, then $2M < n$ and $(2M \bmod n)$ is an even number

If $M > n/2$, then $2M > n$ and $(2M \bmod n = 2M - n)$ is an odd number

Thus, *when* $M < n/2$, the last bit of the message returned by parity$(2^e C)$ will be 0;

when $M > n/2$, the last bit of the message returned by parity$(2^e C)$ will be 1.

Thus, if we can solve parity(C),then we can use parity$(2^e C)$ to answer whether $M > n/2$ or not.

We continute this by checking parity$(2^{2e} C) \Rightarrow$ tells whether $2M \bmod n > n/2$ or not

We continute this by checking parity$(2^{ie} C) \Rightarrow$ tells whether $2^{i-1} M \bmod n > n/2$ or not

Other Public Key Systems

- Rabin Cryptosystem
    - Decryption is not unique
- Elgamal Cryptosystem
    - Expansion of the plaintext (double)
- Knapsack System
    - Already broken
- Elliptic Curve System
    - If directly implement Elgamal on elliptic curve
        - Expansion of plaintext by 4; Restricted plaintext
    - Menezes-Vanston system is more efficient

Rabin Cryptosystem

- Procedure
  - Let $n=pq$ and $p=3 \bmod 4$, $q=3 \bmod 4$
  - Publish n, and a number $b<n$
  - For message m
    - $C=m(m+b) \bmod n$
  - The receiver decrypts ciphertext C
    - $(b^2/4+C)^{1/2}-b/2$

Analysis

- For receiver, need solve equation
  - $x^2+xb=C \bmod n$
  - Let $x_1=x+b/2$, $c=b^2/4+C$, then need
    - Solve $x_1^2 = c \bmod n$
  - Chinese Remainder Theorem implies that
    - $x_1^2 = c \bmod p$
    - $x_1^2 = c \bmod q$
  - When p=3 and q=3 mod 4
    - Solution $x_1=c^{(p+1)/4} \bmod p$ and $x_1=c^{(q+1)/4} \bmod q$
    - Then Chinese Remainder Theorem again to combine solution

Security

- Breaking it <➔ factoring n
- Secure against
  - Chosen plaintext attack
- Not secure against
  - Chosen ciphertext attack
  - Decoding produces three false results in addition to the correct one, so that the correct result must be guessed. This is the major disadvantage of the Rabin cryptosystem and one of the factors which have prevented it from finding widespread practical use.
  - It has been proven that decoding the Rabin cryptosystem is equivalent to the integer factorization problem, which is rather different than for RSA.

Dealing with 4 solutions

- By adding redundancies, for example, the repetition of the last 64 bits, the system can be made to produce a single root.
- If this technique is applied, the proof of the equivalence with the factorization problem fails.

Breaking Rabin System Same as Factoring

- If we can do factorization of n, we clearly can solve the equation $x^2=a \bmod n$
- Assume that we can solve the equation $x^2=a \bmod n$ and get all 4 solutions x_1,x_2,x_3,x_4,where $x_1=n-x_2$, and $x_3=n-x_4$.
- Then $x_1^2= x_3^2 \bmod n$; thus,
- $(pq) \mid (x_1 - x_3)(x_1 + x_3)$
- It is easy to show that (pq) does not divide $(x_1 - x_3)$ or $(x_1 + x_3)$
- Thus, $\gcd(n, x_1 - x_3)$ is one factor of n, and $\gcd(n, x_1 + x_3)$ is another

ElGamal Cryptosystem

- Based on Discrete Logarithm
  - Find unique integer x such that $g^x = y \ mod \ p$
    - Here g is a primitive element in Z_p, p is prime
- Procedure
  - Make p, g, y public, keep x secret
  - Encryption:
    - $E_k(m)=(g^k \ mod \ p, \ m \ y^k \ mod \ p)$
  - Decryption
    - $D_k(y_1,y_2)=y_2(y_1^x)^{-1}$ mod p

Efficiency

- ElGamal encryption is probabilistic, meaning that a single plaintext can be encrypted to many possible ciphertexts, with the consequence that a general ElGamal encryption produces a 2:1 expansion in size from plaintext to ciphertext.
- Encryption under ElGamal requires two exponentiations;
  - however, these exponentiations are independent of the message and can be computed ahead of time if need be.
  - Decryption only requires one exponentiation (instead of division, exponentiate y_1 to $p\text{-}1 - x$).
  - Unlike in the RSA and Rabin systems, ElGamal decryption *cannot* be sped up via the Chinese remainder theorem.

Security of ElGamal

- ElGamal is a simple example of a **semantically secure** asymmetric key encryption algorithm (under reasonable assumptions).
- ElGamal's security rests, in part, on the difficulty of solving the discrete logarithm problem in *G*.
    - Specifically, if the discrete logarithm problem could be solved efficiently, then ElGamal would be broken. However, the security of ElGamal actually relies on the so-called Decisional Diffie-Hellman (DDH) assumption. This assumption is often stronger than the discrete log assumption, but is still believed to be true for many classes of groups.

Semantic Security

- Semantic security is a widely-used definition for security in an PKS.
  - For a cryptosystem to be semantically secure, it must be infeasible for a computationally-bounded adversary to derive significant information about a message (plaintext) when given only its ciphertext and the corresponding public encryption key.
- Semantic security considers only the case of a "passive" attacker, i.e., one who observes ciphertexts and generates chosen ciphertexts using the public key
- Indistinguishability definition is used more commonly than the original definition of semantic security.

Indistinguishability: semantic security.

- Indistinguishability under Chosen Plaintext Attack (IND-CPA) is commonly defined by the following game:
  - A probabilistic polynomial time-bounded adversary is given a public key, which it may use to generate any number of ciphertexts (within polynomial bounds).
  - The adversary generates two equal-length messages *m*0 and *m*1, and transmits them to a challenge oracle along with the public key.
  - The challenge oracle selects one of the messages by flipping a uniformly-weighted coin, encrypts the message under the public key, and returns the resulting ciphertext *c* to the adversary.

Cont.

- The underlying cryptosystem is IND-CPA (and thus semantically secure under chosen plaintext attack) if
  - the adversary cannot determine which of the two messages was chosen by the oracle, with probability significantly greater than 1/2 (the success rate of random guessing).
  - Prob(correct guess)-1/2 < 1/p(k) for a polynomial p, and k is the message size
- a semantically secure encryption scheme must by definition be probabilistic, possessing a component of randomness; if this were not the case, the adversary could simply compute the deterministic encryption of *m*0 and *m*1 and compare these encryptions with the returned ciphertext *c* to successfully guess the oracle's choice.

Deal with deterministic PKS

- RSA, can be made semantically secure (under stronger assumptions) through the use of random encryption padding schemes such as Optimal Asymmetric Encryption Padding (OAEP).

- ElGamal scheme is semantically secure

Security of ElGamal Encryption

- If the computational Diffie-Hellman (CDH) assumption holds the underlying cyclic group *G of order q*, then the encryption function is one-way
  - The CDH assumption states that, given
    - (g, g^a, g^b)
  - for a randomly-chosen generator *g* and random
    - Numbers a, b,
  - it is computationally intractable to compute the value
    - g^{ab}.
  -

Cont.

- If the decisional Diffie-Hellman assumption (DDH) holds in *G*, then ElGamal achieves semantic security. Semantic security is not implied by the computational Diffie-Hellman assumption alone
  - Consider a (multiplicative) cyclic group *G* of order *q*, and with generator *g*. The DDH assumption states that, given g^a and g^b for randomly-chosen 0<a,b <q, the value g^{ab} "looks like" a random element in *G*.
  - This intuitive notion is formally stated by saying that the following two probability distributions are computationally indistinguishable (in the security parameter *q*):
    - (g^a, g^b, g^{ab}), where *a* and *b* are randomly and independently chosen from [1,q-1].
    - (g^a, g^b, g^c), where *a,b,c* are randomly and independently chosen from [1,q-1].

Cont.

- ElGamal encryption is not secure under chosen ciphertext attack.
  - For example, given an encryption (c_1,c_2) of some (possibly unknown) message *m*, one can easily construct a valid encryption (c_1,2c_2) of the message 2*m*.

- Other schemes related to ElGamal which achieve security against chosen ciphertext attacks have also been proposed.
  - The Cramer-Shoup system

Malleability

- An encryption algorithm is malleable
  - if it is possible for an adversary to transform a ciphertext into another ciphertext which decrypts to a related plaintext.
  - That is, given an encryption of a plaintext *m*, it is possible to generate another ciphertext which decrypts to *f*(*m*), for a known function *f*, without necessarily knowing or learning *m*.
- Malleability is often an undesirable property in a general-purpose cryptosystem, since it allows an attacker to modify the contents of a message.
  - But it does allow searchable encryption (Symmetric Searchable Encryption)

ElGamal is malleable.

- In the ElGamal cryptosystem, a plaintext m is encrypted as $E(m) = (g^k, mY^k)$, where (g, Y) is the public key.
- Given such a ciphertext (c_1, c_2), an adversary can compute (c_1, tc_2), which is a valid encryption of tm, for any t.

Bit security of Discrete Log

- Given $g^x = y \bmod p$
  - *We may want to find the value of x*
  - *Find some bits of x*
- Assume that $p-1 = 2^s t$
  - We can find the last s bits of x for sure
  - But to find the other bits of x is same as to find all bits of x!
- Example, the last bit of x is
  - 0 ←→ y is QR iff $y^{(p-1)/2} = 1 \bmod p$
  - 1 ←→ y is NQR iff $y^{(p-1)/2} = -1 \bmod p$

DH Assumption

- Consider a cyclic group G of order q. The DDH assumption states that,
  - given (g,g^a,g^b) for a randomly-chosen generator g and random , the value g^{ab} "looks like" a perfectly random element of G.
- This intuitive notion is formally stated by saying that the following two ensembles are computationally indistinguishable:
  - (g,g^a,g^b,g^{ab}), where g,a,b are chosen at random as described above (this input is called a "DDH tuple");
  - (g,g^a,g^b,g^c), where g,a,b are chosen at random *and* c is chosen at random.
- Diffie-Hellman problem
  - computing g^{ab} from (g,g^a,g^b)

Knapsack Cryptosystem

- Based on subset sum problem
  - Given a set, find a subset with half summation value
  - It is NPC problem generally
- Superincreasing set if $s_i > \Sigma_{j<i} s_j$
- The subset problem over superincreasing set can be solved in polynomial time!
- Been broken by Shamir, 1984
  - Using integer programming tech by Lenstra

Solve Subset Problem

- Let T be the half summation, t=T;
- For i=n downto 1 do
  - If $t \geq s_i$ then
    - $t=t-s_i$
    - Set $x_i=1$
  - Else $x_i=0$
- If $\Sigma x_i s_i = T$ then $(x_1, x_2, \ldots x_n)$ is the solution
  - Else, there is no solution

Knapsack System

- Procedure
  - Select a superincreasing set *s*
  - Let *p* be prime larger than set summation of *s*,
  - Select integer *a*, keep *s*, *a*, *p* secret
  - Make vector $t=(as_1, as_2, \ldots as_n) \bmod p$ public
  - Encryption
    - Ciphertext $C = E(x_1,x_2,\ldots x_n)=\Sigma x_i t_i \bmod p$
  - Decryption
    - Solve the subset summation problem $(s, a^{-1}C \bmod p)$

Break of this system

- Adi Shamir, A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem. CRYPTO 1982, pp279-288.
- http://dsns.csie.nctu.edu.tw/research/crypto/HTML/PDF/C82/279.PDF

Elliptic Curve Cryptography

- majority of public-key crypto (RSA, D-H) use either integer or polynomial arithmetic with very large numbers/polynomials
- imposes a significant load in storing and processing keys and messages
- an alternative is to use elliptic curves
  - offers same security with smaller bit sizes
  - was suggested independently by Neal Koblitz[1] and Victor S. Miller[2] in 1985.
- Paper
  - http://www.ams.org/journals/mcom/1987-48-177/S0025-5718-1987-0866109-5/S0025-5718-1987-0866109-5.pdf

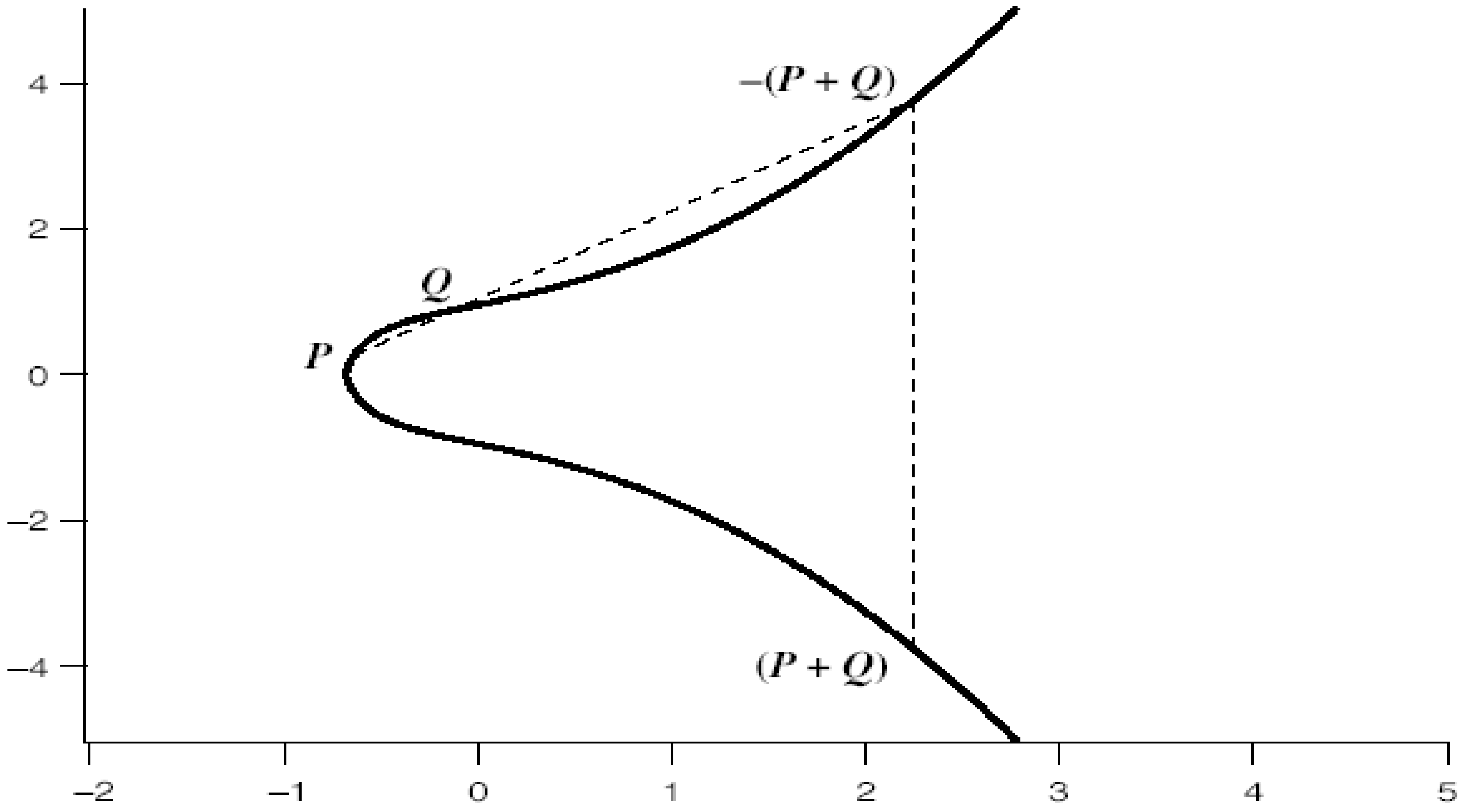
USE of ECC

- At the RSA Conference 2005, the National Security Agency (NSA) announced Suite B which exclusively uses ECC for digital signature generation and key exchange. The suite is intended to protect both classified and unclassified national security systems and information.[5]
- Recently, a large number of cryptographic primitives based on bilinear mappings on various elliptic curve groups, such as the Weil and Tate pairings, have been introduced. Schemes based on these primitives provide efficient identity-based encryption as well as pairing-based signatures, signcryption, key agreement, and proxy re-encryption.

Real Elliptic Curves

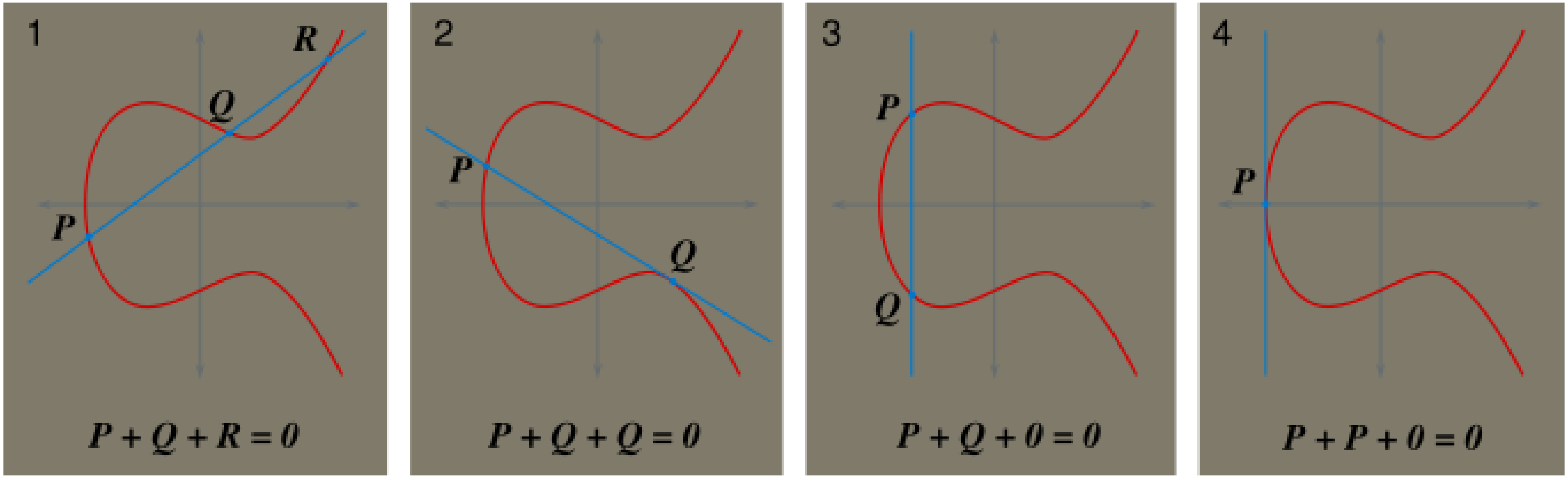
- an elliptic curve is defined by an equation in two variables x & y, with coefficients
- consider a cubic elliptic curve of form
  - $y^2 = x^3 + ax + b$
  - where x,y,a,b are all real numbers
  - also define zero point O
- have addition operation for elliptic curve
  - geometrically sum of Q+P is reflection of intersection R

Real Elliptic Curve Example



(b) $y^2 = x^3 + x + 1$

More ECC example



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 p
    - best in software
  - binary curves $E_{2m}(a,b)$ defined over $GF(2^n)$
    - use polynomials with binary coefficients
    - best in hardware

ECC using Fp

- An elliptic curve is the locus of points in the affine plane whose coordinates satisfy a certain cubic equation together with a point at infinity O
  - the point at which the locus in the projective plane intersects the line at infinity. In the case of characteristic p > 3 the *defining equation* of can be written: $y^2 = x^3 + ax + b$
  - where $a \in \mathbb{F}_p$ and $b \in \mathbb{F}_p$ are constants such that

$$4a^3 + 27b^2 \neq 0$$

Operations

- We define the negative of a point $P = (x,y)$ to be $-P = (x, -y)$ for $P \in E(\mathbb{F}_p)$
  - $P + Q = Q + P$ (*commutativity*)
  - $(P + Q) + R = P + (Q + R)$ (*associativity*)
  - $P + 0 = 0 + P = P$ (*existence of an identity element*)
  - there exists $(-P)$ such that $-P + P = P + (-P) = 0$ (*existence of inverses*)
  - Here 0 is a point at infinity

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 Diffie-Hellman

- can do key exchange analogous to D-H
- users select a suitable curve $E_p(a,b)$
- select base point $G=(x_1,y_1)$ with large order n s.t. $n*G=O$
  - Typically n is required to be prime and close to size of all points
- A & B select private keys $n_A<n$, $n_B<n$
- compute public keys: $P_A=n_A \times G$, $P_B=n_B \times G$
- compute shared key: $K=n_A \times P_B$, $K=n_B \times P_A$
  - same since $K=n_A \times n_B \times G$

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
  - Order of G is a large prime, close to number of points
- each user chooses private key $n_A < n$
- and computes public key $P_A = n_A \times G$
- to encrypt P_m : $C_m = \{kG, \; P_m + k \; P_A\}$, k random
- decrypt C_m compute:

$$P_m + kP_A - n_A(kG) \;=\; P_m + k(n_A G) - n_A(kG) \;=\; P_m$$

Choosing of parameters

- Several classes of curves are weak and should be avoided:
  - curves over F(2^m) with non-prime *m* are vulnerable to Weil descent attacks.[7][8]
  - curves such that *n* divides $p^B - 1$ (where *p* is the characteristic of the field – *q* for a prime field, or 2 for a binary field) for sufficiently small *B* are vulnerable to MOV attack[9][10] which applies usual DLP in a small degree extension field of to solve ECDLP. The bound *B* should be chosen so that discrete logarithms in the field are at least as difficult to compute as discrete logs on the elliptic curve .[11]
  - curves such that |F_q| =q are vulnerable to the attack that maps the points on the curve to the additive group of F_q [12][13][14]

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

security

- Since all the fastest known algorithms that allow to solve the ECDLP (baby-step giant-step, Pollard's rho, etc.), need $O(\sqrt{n})$ steps, it follows that the size of the underlying field shall be roughly twice the security parameter.