Public Key Cryptography

Lecture 5

Public Key Cryptography

Index

1. The Story of Alice and Bob
2. Public Key Cryptography
3. One-Way Functions and Trapdoor Functions
4. RSA (A First Example of Public Key Cryptosystem)

The Story of Alice and Bob

From ”The Alice and Bob After Dinner Speech” given at the Zurich Seminar, April 1984, by John Gordon:

> *... So let's talk about coding theory [note: actually cryptography!]. There are perhaps some of you here tonight who are not experts in coding theory, but rather have been dragged here kicking and screaming. So I thought it would be a good idea if I gave you a sort of instant, five minute graduate course in coding theory.*
>
> *Coding theorists are concerned with two things. Firstly and most importantly they are concerned with the private lives of two people called Alice and Bob. In theory papers, whenever a coding theorist wants to describe a transaction between two parties he doesn't call them A and B. No. For some longstanding traditional reason he calls them Alice and Bob.*

The Story of Alice and Bob (cont.)

Now there are hundreds of papers written about Alice and Bob. Over the years Alice and Bob have tried to defraud insurance companies, they've played poker for high stakes by mail, and they've exchanged secret messages over tapped telephones.

If we put together all the little details from here and there, snippets from lots of papers, we get a fascinating picture of their lives. This may be the first time a definitive biography of Alice and Bob has been given.

In papers written by American authors Bob is frequently selling stock to speculators. From the number of stock market deals Bob is involved in we infer that he is probably a stockbroker. However from his concern about eavesdropping he is probably active in some subversive enterprise as well. And from the number of times Alice tries to buy stock from him we infer she is probably a speculator. Alice is also concerned that her financial dealings with Bob are not brought to the attention of her husband. So Bob is a subversive stockbroker and Alice is a two-timing speculator.

The Story of Alice and Bob (cont.)

But Alice has a number of serious problems. She and Bob only get to talk by telephone or by electronic mail. In the country where they live the telephone service is very expensive. And Alice and Bob are cheapskates. So the first thing Alice must do is minimize the cost of the phone call.

The telephone is also very noisy. Often the interference is so bad that Alice and Bob can hardly hear each other. On top of that Alice and Bob have very powerful enemies. One of their enemies is the Tax Authority. Another is the Secret Police. This is a pity, since their favorite topics of discussion are tax frauds and overthrowing the government.

These enemies have almost unlimited resources. They always listen in to telephone conversations between Alice and Bob. And these enemies are very sneaky. One of their favorite tricks is to telephone Alice and pretend to be Bob.

The Story of Alice and Bob (cont.)

Well, you think, so all Alice has to do is listen very carefully to be sure she recognizes Bob's voice. But no. You see Alice has never met Bob. She has no idea what his voice sounds like.

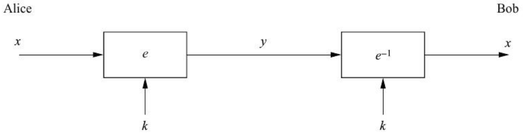
So you see Alice has a whole bunch of problems to face. Oh yes, and there is one more thing I forgot so say - Alice doesn't trust Bob. We don't know why she doesn't trust him, but at some time in the past there has been an incident.

Now most people in Alice's position would give up. Not Alice. She has courage which can only be described as awesome. Against all odds, over a noisy telephone line, tapped by the tax authorities and the secret police, Alice will happily attempt, with someone she doesn't trust, whom she cannot hear clearly, and who is probably someone else, to fiddle her tax returns and to organize a coup d'etat, while at the same time minimizing the cost of the phone call.

A coding theorist is someone who doesn't think Alice is crazy...

Private Key (Symmetric) Cryptography

- characteristic for classical cryptography
- once the encryption key was known, the decryption key could be easily recovered (if not the same)
- basic protocol and analogy:



Shortcomings:

- key distribution problem
- number of keys
- non-repudation

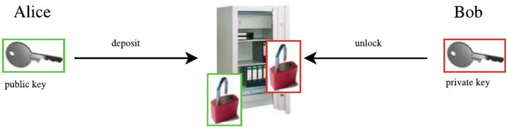
Public Key (Asymmetric) Cryptography

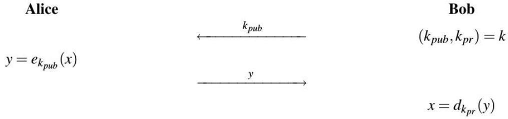
- 1969-1975: Ellis, Cocks, Williamson
  (UK's Government Communications Headquarters)
  1976: Diffie and Hellman (USA)
- *Idea*: given the encryption key, one cannot determine the decryption key in a "reasonable" time.
  Hence $f : \mathcal{P} \to \mathcal{C}$ can be easily computed once the encryption key K_E is known, but $f^{-1} : \mathcal{C} \to \mathcal{P}$ is very difficult (practically impossible) to be computed without knowing the decryption key K_D.

The story of Alice, Bob and their friends begins...

Public Key (Asymmetric) Cryptography (cont.)

- analogy, basic protocol and key transport:





Alice | **Bob**

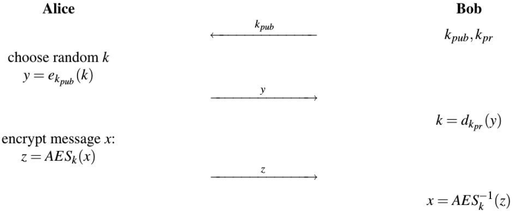
Bob: $(k_{pub}, k_{pr}) = k$

$\xleftarrow{\quad k_{pub} \quad}$

Alice: $y = e_{k_{pub}}(x)$

$\xrightarrow{\quad y \quad}$

Bob: $x = d_{k_{pr}}(y)$



Alice | **Bob**

Bob: k_{pub}, k_{pr}

$\xleftarrow{\quad k_{pub} \quad}$

Alice: choose random k
$y = e_{k_{pub}}(k)$

$\xrightarrow{\quad y \quad}$

Bob: $k = d_{k_{pr}}(y)$

Alice: encrypt message x:
$z = AES_k(x)$

$\xrightarrow{\quad z \quad}$

Bob: $x = AES_k^{-1}(z)$

Public Key (Asymmetric) Cryptography (cont.)

Main security mechanisms of public key algorithms:

- **Key Establishment**
  There are protocols for establishing secret keys over an insecure channel. Examples for such protocols include the Diffie-Hellman key exchange or RSA key transport protocols.
- **Non-Repudiation**
  Providing non-repudiation and message integrity can be realized with digital signature algorithms, e.g., RSA, DSA or ECDSA.
- **Identification**
  We can identify entities using challenge-and-response protocols together with digital signatures, e.g., in applications such as smart cards for banking or for mobile phones.
- **Encryption**
  We can encrypt messages using algorithms such as RSA or ElGamal.

Public Key (Asymmetric) Cryptography (cont.)

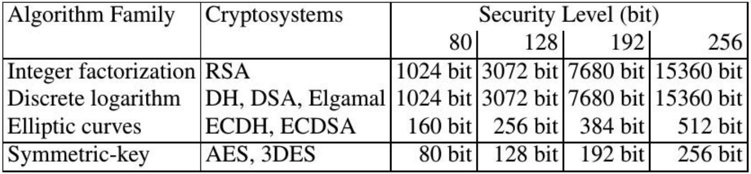
Public key algorithm families of practical relevance:

- **Integer-Factorization Schemes** (mid 1970's)
  Several public-key schemes are based on the fact that it is difficult to factor large integers. The most prominent representative of this algorithm family is RSA.
- **Discrete Logarithm Schemes** (mid 1970's)
  There are several algorithms which are based on what is known as the discrete logarithm problem in finite fields. The most prominent examples include the Diffie-Hellman key exchange, ElGamal encryption or the Digital Signature Algorithm (DSA).
- **Elliptic Curve (EC) Schemes** (mid 1980's)
  A generalization of the discrete logarithm algorithm are elliptic curve public-key schemes. The most popular examples include the Elliptic Curve Diffie-Hellman key exchange (ECDH) and the Elliptic Curve Digital Signature Algorithm (ECDSA).

Public Key (Asymmetric) Cryptography (cont.)

An algorithm is said to have a *security level of n bit* if the best known attack requires 2^n steps.

Bit lengths of public-key algorithms for different security levels:



Algorithm Family	Cryptosystems	Security Level (bit)			
		80	128	192	256
Integer factorization	RSA	1024 bit	3072 bit	7680 bit	15360 bit
Discrete logarithm	DH, DSA, Elgamal	1024 bit	3072 bit	7680 bit	15360 bit
Elliptic curves	ECDH, ECDSA	160 bit	256 bit	384 bit	512 bit
Symmetric-key	AES, 3DES	80 bit	128 bit	192 bit	256 bit

The computational complexity of the three algorithm families grows roughly with the cube bit length. As an example, increasing the bit length from 1024 to 3072 in a given RSA signature generation software results in an execution that is $3^3 = 27$ times slower!

Intermezzo: key exchange extra motivation

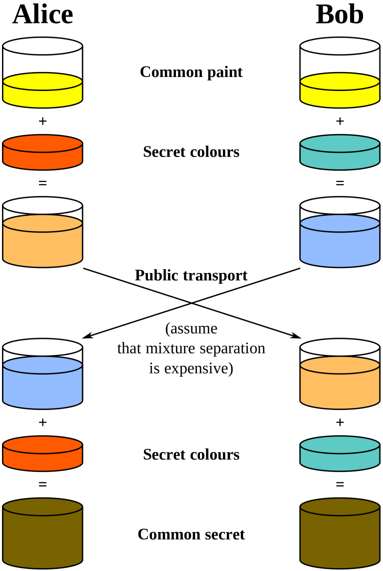
Problem 1

Alice needs a bike and Bob offers to lend her his bike. Unfortunately, their schedule is very tight and they are not able to meet directly, even if both of them have classes at the same university. Each of them may use one lock and the bike stand from the university. Describe a way in which they may transfer the bike from Bob to Alice, without the risk of being stolen from the bike stand.

Problem 2

Alice and Bob want to establish a common secret color for their private use. They cannot communicate without the risk of being intercepted by Eve. From the common color Eve knows a basic color, which is assumed to be public. Alice and Bob may each choose a secret color and may send a single public message to each other. Describe a way in which they may obtain the same common secret color.

Intermezzo: Diffie-Hellman Key Exchange



One-Way Functions and Trapdoor Functions

New notions appeared: *one-way function* and *trapdoor function*.

Definition

A function $f : X \to Y$ is called a *one-way function* if $f(x)$ is "easy" to compute $\forall x \in X$, but for "most" elements $y \in Im(f)$ it is "computationally infeasible" to find $z \in X$ such that $f(z) = y$.

Definition

A *trapdoor function* is a one-way function $f : X \to Y$ with the property that given some extra information it becomes feasible to find $\forall y \in Im(f)$, an element $z \in X$ such that $f(z) = y$.

Are there such functions? No one has yet definitively proved their existence under reasonable (and rigorous) definitions of "easy" and "computationally infeasible". But there are many good candidates. Their status might be not permanent!

One-Way Function

Example. Let $f : \mathbb{Z}_{17}^{*} \to \mathbb{Z}_{17}^{*}$, $f(x) = 3^{x} \bmod 17$.
Explicitly, we have the following correspondence $x \mapsto f(x)$:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
3	9	10	13	5	15	11	16	14	8	7	4	12	2	6	1

- Given $x \in \mathbb{Z}_{17}^{*}$, it is relatively easy to find $f(x)$.
- However, given a number such as 7, without having the table in front of you, it is harder to find x given that $f(x) = 7$.
- Even for very large numbers, $f(x)$ can be computed efficiently, whereas the process of finding x from $f(x)$ is much harder.

This example gives the flavor of the concept of one-way function.

Trapdoor Function

Example. Consider the primes $p = 48611$, $q = 53993$ and let $n = pq = 2624653723$. Define $f : \mathbb{Z}_n \to \mathbb{Z}_n$ by $f(x) = x^3 \bmod n$.

- Computing $f(x)$ is a relatively simple thing to do.
- To reverse the procedure is much more difficult. This is the case if the factors of n are unknown and large.
- However, if the factors p and q of n are known, then there is an efficient algorithm for computing such modular cube roots.

This example illustrates the concept of a trapdoor function.

If one selects p and q to be very large distinct prime numbers (each having about 100 decimal digits), then it is a difficult problem to deduce p and q simply from n. This is the well-known *Integer Factorization Problem* and a source of many trapdoor functions.

RSA

- Rivest, Shamir, Adleman (1977)
- Integer Factorization Problem ($n = pq$)
- There is not any known polynomial-time algorithm for that.

1. Key generation. Alice creates a public key and a private key.

1.1. Generates 2 random large distinct primes p, q of approximately same size.
1.2. Computes $n = pq$ and $\varphi(n) = (p-1)(q-1)$ (the Euler function).
1.3. Randomly selects $1 < e < \varphi(n)$ with $gcd(e, \varphi(n)) = 1$.
1.4. Computes $d = e^{-1} \bmod \varphi(n)$.
1.5. Alice's public key is $K_E = (n, e)$; her private key is $K_D = d$.

e and d are called the *encryption exponent* and the *decryption exponent* respectively, and n is called the *modulus*.

RSA (cont.)

2. Encryption. Bob sends an encrypted message to Alice.

2.1. Gets Alice's public key $K_E = (n, e)$.

2.2. Represents the message as a number m between 0 and $n - 1$.

2.3. Computes $c = m^e \bmod n$.

2.4. Sends the ciphertext c to Alice.

3. Decryption. Alice decrypts the message from Bob.

3.1 Alice uses the private key $K_D = d$ to get the message $m = c^d \bmod n$.

- $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_n$
- the encryption function is $f : \mathbb{Z}_n \to \mathbb{Z}_n$, $f(m) = m^e \bmod n$.
- the decryption function is $f^{-1} : \mathbb{Z}_n \to \mathbb{Z}_n$, $f^{-1}(c) = c^d \bmod n$.

Correctness of RSA

Theorem

RSA is correct.

Proof. We show that $c^d \equiv m \pmod{n}$.
Since $ed \equiv 1 \pmod{\varphi(n)}$, $\exists k \in \mathbb{Z}$ such that

$$ed = 1 + k\varphi(n) = 1 + k(p-1)(q-1).$$

Let us show that $m^{ed} \equiv m \pmod{p}$.
If $(m,p) = p$, then it is clear.
If $(m,p) = 1$, then $m^{p-1} \equiv 1 \pmod{p}$ by Fermat's Little Theorem (note that $p \nmid m$). We have

$$m^{ed} \equiv m^{1+k(p-1)(q-1)} \equiv m \cdot (m^{(p-1)})^{k(q-1)} \equiv m \pmod{p}.$$

Similarly $m^{ed} \equiv m \pmod{q}$. Since $p \neq q$ are primes, we deduce that $m^{ed} \equiv m \pmod{p \cdot q}$, whence $c^d \equiv (m^e)^d \equiv m \pmod{n}$.

Integer Factorization vs. Euler's Function

Theorem

Suppose that $n = pq$ for some primes p and q. Then the knowledge of p and q is equivalent to the knowledge of $\varphi(n)$.

Proof.
Case 1. n **even**. Then $p = 2$, $q = \frac{n}{2}$ and $\varphi(n) = \frac{n}{2} - 1$.
Case 2. n **odd**. Given n, p, q we get

$$\varphi(n) = (p-1)(q-1) = n + 1 - (p+q)\,.$$

Given n and $\varphi(n)$, we have

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

Notice that $p + q$ is even, so denote $2b = n + 1 - \varphi(n)$. Then p, q are the roots of $x^2 - 2bx + n = 0$, that is, $b \pm \sqrt{b^2 - n}$.

Utility of public-key cryptosystems

- widely used in nowadays life
- usually used for sending small amounts of information (keys), being slower than the classical private-key cryptosystems

Example. General setting:

- Use the RSA cryptosystem.
- Use a 27-letters alphabet for plaintext and ciphertext: _ (blank) with numerical equivalent 0 and letters $A - Z$ (the English alphabet) with numerical equivalents 1-26.

_	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

- Plaintext message units are blocks of k letters, whereas ciphertext message units are blocks of l letters. The plaintext is completed with blanks, when necessary.
- We must have $27^k < n < 27^l$.

RSA example - encryption

Let $k = 2$, $l = 3$, $K_E = (n, e) = (1643, 67)$. We have $27^2 < 1643 < 27^3$.

- Plaintext: algebra_
- Split the plaintext: al /ge /br /a_
- Write the numerical equivalents: 39 194 72 27
  $(\text{al} \mapsto \boxed{1} \cdot 27 + \boxed{12} = 39,\ \text{ge} \mapsto \boxed{7} \cdot 27 + \boxed{5} = 194,$
  $\text{br} \mapsto \boxed{2} \cdot 27 + \boxed{18} = 72,\ \text{a\_} \mapsto \boxed{1} \cdot 27 + \boxed{0} = 27)$
- Encrypt ($m^e \bmod n$): 1428 498 919 1503
  $(39^{67} \bmod 1643 = \cdots = 1428,\ 194^{67} \bmod 1643 = \cdots = 498,$
  $72^{67} \bmod 1643 = \cdots = 919,\ 27^{67} \bmod 1643 = \cdots = 1503)$
- Write the literal equivalents: AYX _RL AGA BAR
  $(1428 = \boxed{1} \cdot 27^2 + \boxed{25} \cdot 27 + \boxed{24} \mapsto \text{AYX},$
  $498 = \boxed{0} \cdot 27^2 + \boxed{18} \cdot 27 + \boxed{12} \mapsto \text{\_RL},$
  $919 = \boxed{1} \cdot 27^2 + \boxed{7} \cdot 27 + \boxed{1} \mapsto \text{AGA},$
  $1503 = \boxed{2} \cdot 27^2 + \boxed{1} \cdot 27 + \boxed{18} \mapsto \text{BAR})$
- Ciphertext: AYX_RLAGABAR

RSA example - encryption (cont.)

Some details:

We compute $39^{67} \bmod 1643$ by repetead squaring modular exponentiation.
We have $67 = 2^0 + 2^1 + 2^6$. Compute modulo 1643:

$$
\begin{aligned}
39^{(2^0)} &= 39, \\
39^{(2^1)} &= 39^{(2^0)} \cdot 39^{(2^0)} = 39 \cdot 39 = 1521, \\
39^{(2^2)} &= 39^{(2^1)} \cdot 39^{(2^1)} = 1521 \cdot 1521 = 97, \\
39^{(2^3)} &= 39^{(2^2)} \cdot 39^{(2^2)} = 97 \cdot 97 = 1194, \\
39^{(2^4)} &= 39^{(2^3)} \cdot 39^{(2^3)} = 1194 \cdot 1194 = 1155, \\
39^{(2^5)} &= 39^{(2^4)} \cdot 39^{(2^4)} = 1155 \cdot 1155 = 1552, \\
39^{(2^6)} &= 39^{(2^5)} \cdot 39^{(2^5)} = 1552 \cdot 1552 = 66.
\end{aligned}
$$

Then $39^{67} = 39^{2^0+2^1+2^6} = 39 \cdot 1521 \cdot 66 = 1428 \pmod{1643}$.

RSA example - decryption

We have $n = 1643 = 31 \cdot 53$, hence $\varphi(n) = 30 \cdot 52 = 1560$.
We have $K_D = d = e^{-1} \bmod \varphi(n) = 67^{-1} \bmod 1560 = \cdots = 163$.

- Ciphertext: AYX_RLAGABAR
- Split the ciphertext: AYX /_RL /AGA /BAR
- Write the numerical equivalents: 1428 498 919 1503

  (AYX $\mapsto 1428 = \boxed{1} \cdot 27^2 + \boxed{25} \cdot 27 + \boxed{24}$,

  _RL $\mapsto 498 = \boxed{0} \cdot 27^2 + \boxed{18} \cdot 27 + \boxed{12}$,

  AGA $\mapsto 919 = \boxed{1} \cdot 27^2 + \boxed{7} \cdot 27 + \boxed{1}$,

  BAR $\mapsto 1503 = \boxed{2} \cdot 27^2 + \boxed{1} \cdot 27 + \boxed{18}$)
- Decryption ($c^d \bmod n$): 39 194 72 27

  ($1428^{163} \bmod 1643 = \cdots = 39$, $498^{163} \bmod 1643 = \cdots = 194$,
  $919^{163} \bmod 1643 = \cdots = 72$, $1503^{163} \bmod 1643 = \cdots = 27$)
- Write the literal equivalents: al ge br a_

  ($39 = \boxed{1} \cdot 27 + \boxed{12} \mapsto$ al, $194 = \boxed{7} \cdot 27 + \boxed{5} \mapsto$ ge,
  $72 = \boxed{2} \cdot 27 + \boxed{18} \mapsto$ br, $27 = \boxed{1} \cdot 27 + \boxed{0} \mapsto$ a_)
- Plaintext: algebra_

RSA example - decryption (cont.)

Some details:

We compute $67^{-1} \bmod 1560 = 163$ by the extended Euclidean algorithm.

$$\begin{aligned}
1560 &= 23 \cdot 67 + 19 \\
67 &= 3 \cdot 19 + 10 \\
19 &= 1 \cdot 10 + 9 \\
10 &= 1 \cdot 9 + 1 \\
9 &= 9 \cdot 1
\end{aligned}$$

Then $(1560, 67) = 1$, hence there exists $67^{-1} \bmod 1560$.
We have:

$$\begin{aligned}
1 &= 10 - 1 \cdot 9 = 10 - 1 \cdot (19 - 1 \cdot 10) = 2 \cdot 10 - 1 \cdot 19 \\
&= 2 \cdot (67 - 3 \cdot 19) - 1 \cdot 19 = 2 \cdot 67 - 7 \cdot 19 \\
&= 2 \cdot 67 - 7 \cdot (1560 - 23 \cdot 67) = 163 \cdot 67 - 7 \cdot 1560.
\end{aligned}$$

hence $67^{-1} \bmod 1560 = 163$.

RSA example - decryption (cont.)

Some details:

We compute 1428^{163} mod 1643 by repetead squaring modular exponentiation.
We have $163 = 2^0 + 2^1 + 2^5 + 2^7$. Compute modulo 1643:

$$\begin{aligned}
1428^{(2^0)} &= 1428,\\
1428^{(2^1)} &= 1428^{(2^0)} \cdot 1428^{(2^0)} = 1428 \cdot 1428 = 221,\\
1428^{(2^2)} &= 1428^{(2^1)} \cdot 1428^{(2^1)} = 221 \cdot 221 = 1194,\\
1428^{(2^3)} &= 1428^{(2^2)} \cdot 1428^{(2^2)} = 1194 \cdot 1194 = 1155,\\
1428^{(2^4)} &= 1428^{(2^3)} \cdot 1428^{(2^3)} = 1155 \cdot 1155 = 1552,\\
1428^{(2^5)} &= 1428^{(2^4)} \cdot 1428^{(2^4)} = 1552 \cdot 1552 = 66,\\
1428^{(2^6)} &= 1428^{(2^5)} \cdot 1428^{(2^5)} = 66 \cdot 66 = 1070,\\
1428^{(2^7)} &= 1428^{(2^5)} \cdot 1428^{(2^5)} = 1070 \cdot 1070 = 1372.
\end{aligned}$$

Then $1428^{163} = 1428^{2^0+2^1+2^5+2^7} = 1428 \cdot 221 \cdot 66 \cdot 1372 = 39$ (mod 1643).

Selective Bibliography

- N. Koblitz, *A Course in Number Theory and Cryptography*, Springer, 1994.
- A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997. [http://www.cacr.math.uwaterloo.ca/hac]
- C. Paar, J. Pelzl, *Understanding Cryptography*, Springer, 2009.