Public Key Encryption

Anders Västberg

vastberg@kth.se

Slides by M. Goodrich, R. Tamassia and Göran Andersson

Recap Symmetric Key Encryption II

- Block ciphers
- Hill cipher
- Transposition cipher
- AES
- Block Cipher Modes

Public Key Encryption

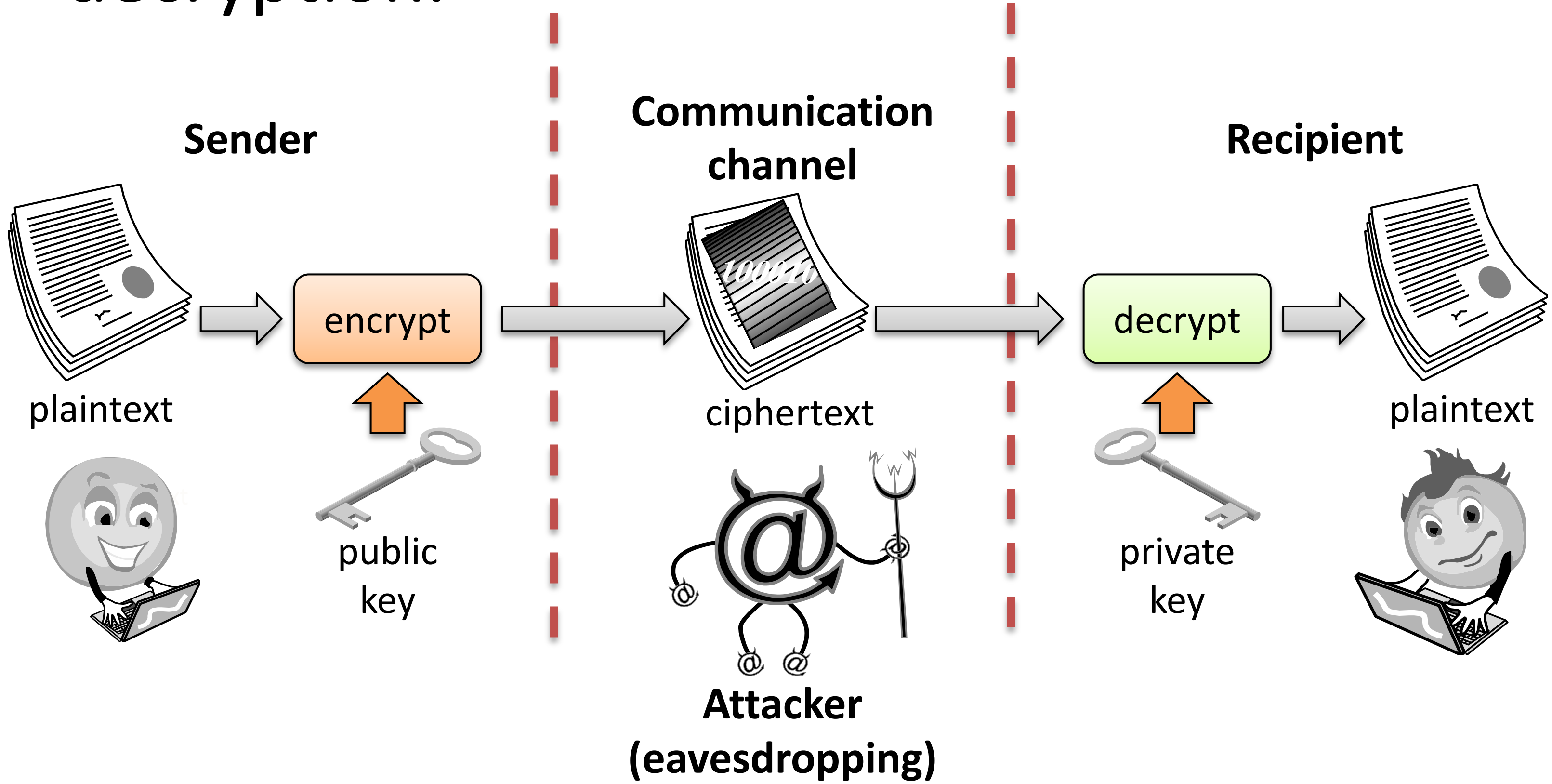
- Concept of Public Key Encryption
- Math Concepts Review
- RSA
- Elgamal Cryptosystem
- Diffie-Hellman Key Exchange
- Man-in-the-middle attacks
- Prime numbers

Public-Key Cryptography

- Bob has two keys: a **private key,** S_B, which Bob keeps **s**ecret, and a **public key,** P_B, which Bob broadcasts widely.
- Alice encrypts using Bob's public key, P_B,
- $C = E_{P_B}(M)$
- Bob then uses his private key to decrypt the message
- $M = D_{S_B}(C)$.

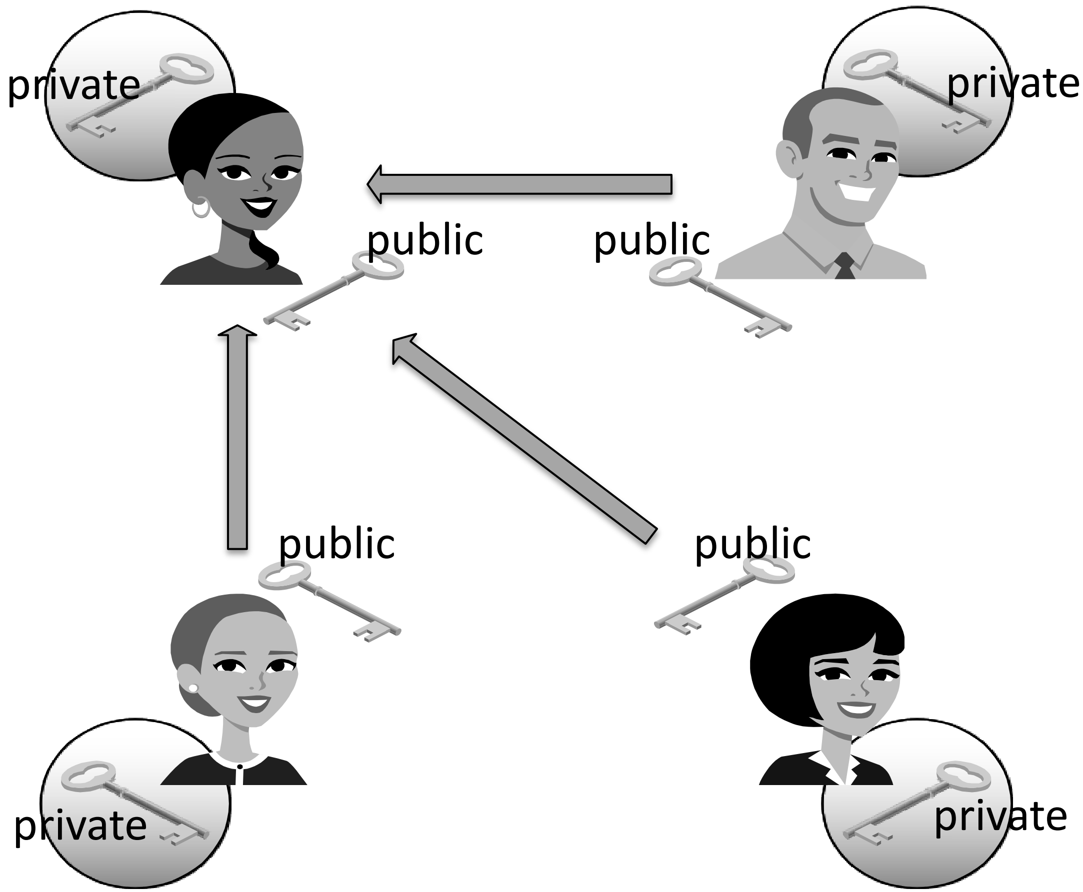
Public-Key Cryptography

- Separate keys are used for encryption and decryption.



Public Key Distribution

- Only one key pair is needed for each participant



$$2\,n \quad \text{keys}$$

$$n > 5 \Rightarrow 2\,n < \binom{n}{2}$$

Math Concepts (Review)

Fundamental theorem of arithmetic

- Every integer $n > 1$ has a unique decomposition of prime factors

$$n = \prod_{i=1}^{r} p_i^{e_i}$$

$$\text{Example}: 13\,276\,725 = 3 \times 5^2 \times 7 \times 11^3 \times 19$$

Coprime

- Two integers n_1 and n_2 are relatively prime or coprimes iff $\gcd(n_1, n_2) = 1$

$$\text{Example}: \gcd(15, 16) = \gcd\left(3 \times 5, 2^4\right) = 1$$

$$\text{Example}: \gcd(700, 392) = \gcd\left(2^2\, 5^2\, 7, 2^3\, 7^2\right)$$

$$= \gcd\left(\mathbf{2^2}\, 5^2\, \mathbf{7^1}, \mathbf{2^2}\, 2^1\, \mathbf{7^1}\, 7^1\right) = \mathbf{2^2}\, \mathbf{7^1} = 28 \neq 1$$

Math Concepts (Review)

Inverse

- In $\mathbf{Z}_n$ $i = a^{-1}$ is the (multiplicative) inverse to a iff
  $a\, i = 1 \bmod n$

  $$a \in \mathbf{Z}_n, \ : \gcd(a, n) = 1 \Leftrightarrow a^{-1} \in \mathbf{Z}_n$$

- This means that if a is a coprime to n then a^{-1} exists

Euler's totient function

- In $\mathbf{Z}_n$ $\phi(n)$ gives the number of coprimes to n

$$\phi(n) = n \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right)$$

Problem

- How many invertible elements are there in $\mathbf{Z}_{19}$?
- in $\mathbf{Z}_{63}$?

$$\phi(19) = 19\left(1 - \frac{1}{19}\right) = 19 - 1 = 18$$

$$\phi(63) = \phi\left(3^2 \times 7\right) = 3^2 \times 7\left(1 - \frac{1}{3}\right)\left(1 - \frac{1}{7}\right)$$

$$= 3\,(3 - 1)\,(7 - 1) = 36$$

Math Concepts (Review)

Euler's theorem

$$a \in \mathbf{Z}_n,\ \gcd(a, n) = 1 \Rightarrow a^{\phi(n)} \equiv 1 \bmod (n)$$

Example: What's $5^{10200} \bmod 10403$?

- Consider $\mathbf{Z}_n$, $n = 10403 = 101 \times 103$.
- We have $\phi(10403) = (101-1)(103-1) = 10200$.
- Then since n and 5 are coprimes we have: $5^{10200} = 5^{\phi(10403)} = 1 \bmod 10403$

Problem

- Compute 10^{842} in $\mathbf{Z}_{147}$

$$n = 147 = 3 \times 7^2 \Rightarrow \gcd(147,\ 10) = \gcd\left(3 \times 7^2,\ 2 \times 5\right) = 1$$

$$\phi(147) = 7\,(3-1)\,(7-1) = 84$$

$$\therefore\ 10^{842} = 10^{\phi(147)\times 10+2} \equiv_{147} 1 \times 10^2 = 100$$

RSA

- RSA encryption was introduced in 1977 by
  Ron **R**ivest, Adi **S**hamir and Len **A**dleman
- Its security is based on the fact that it is time-consuming to factorize large numbers
- The encryption method makes use of Euler's theorem

RSA Cryptosystem

- Setup:
  - $n = pq$, with p and q primes
  - e relatively prime to $\phi(n) = (p-1)(q-1)$
  - d inverse of e in $Z_{\phi(n)}$
- Keys:
  - Public key: $K_E = (n, e)$
  - Private key: $K_D = d$
- Encryption:
  - Plaintext M in Z_n
  - $C = M^e \bmod n$
- Decryption:
  - $M = C^d \bmod n$

- Example
  - Setup:
    - $p = 7, \;\; q = 17$
    - $n = 7 \cdot 17 = 119$
    - $\phi(n) = 6 \cdot 16 = 96$
    - $e = 5$
    - $d = 77$
  - Keys:
    - public key: $(119, 5)$
    - private key: 77
  - Encryption:
    - $M = 19$
    - $C = 19^5 = 66 \bmod 119$
  - Decryption:
    - $C = 66^{77} = 19 \bmod 119$

Complete RSA Example

- Setup:
  - $p = 5, q = 11$
  - $n = 5 \cdot 11 = 55$
  - $\phi(n) = 4 \cdot 10 = 40$
  - $e = 3$
  - $d = 27 \ (3 \cdot 27 = 81 = 2 \cdot 40 + 1)$
- Encryption
  - $C = M^3 \bmod 55$
- Decryption
  - $M = C^{27} \bmod 55$

M	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
C	1	8	27	9	15	51	13	17	14	10	11	23	52	49	20	26	18	2
M	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
C	39	25	21	33	12	19	5	31	48	7	24	50	36	43	22	34	30	16
M	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
C	53	37	29	35	6	3	32	44	45	41	38	42	4	40	46	28	47	54

Problem

- With $n = 323$ and $e = 5$ encrypt the message $m = 4$

$$c = m^e = 4^5 = 1024 = 55 + 3 \times 323 \equiv_{323} 55$$

Security

- Security of RSA based on difficulty of factoring
  - Widely believed
  - Best known algorithm takes exponential time
- RSA Security factoring challenge (discontinued)
- In 1999, 512-bit challenge factored in 4 months using 35.7 CPU-years
  - 160 175-400 MHz SGI and Sun
  - 8 250 MHz SGI Origin
  - 120 300-450 MHz Pentium II
  - 4 500 MHz Digital/Compaq
- In 2005, a team of researchers factored the RSA-640 challenge number using 30 2.2GHz CPU years
- In 2004, the prize for factoring RSA-2048 was $200,000
- Current practice is 2,048-bit keys
- Estimated resources needed to factor a number within one year

Length (bits)	PCs	Memory
430	1	128MB
760	215,000	4GB
1,020	342×10^{6}	170GB
1,620	1.6×10^{15}	120TB

Correctness

- We show the correctness of the RSA cryptosystem for the case when the plaintext $\boldsymbol{M}$ does not divide $\boldsymbol{n}$
- Namely, we show that

  $$(\boldsymbol{M}^{\boldsymbol{e}})^{\boldsymbol{d}} \bmod \boldsymbol{n} = \boldsymbol{M}$$
- Since $\boldsymbol{ed} \bmod \phi(\boldsymbol{n}) = 1$, there is an integer $\boldsymbol{k}$ such that

  $$\boldsymbol{ed} = \boldsymbol{k}\phi(\boldsymbol{n}) + 1$$
- Since $\boldsymbol{M}$ does not divide $\boldsymbol{n}$, by Euler's theorem we have

  $$\boldsymbol{M}^{\phi(\boldsymbol{n})} \bmod \boldsymbol{n} = 1$$
- Thus, we obtain

  $$\begin{aligned}
  &(\boldsymbol{M}^{\boldsymbol{e}})^{\boldsymbol{d}} \bmod \boldsymbol{n} = \\
  &\quad \boldsymbol{M}^{\boldsymbol{ed}} \bmod \boldsymbol{n} = \\
  &\quad \boldsymbol{M}^{\boldsymbol{k}\phi(\boldsymbol{n}) + 1} \bmod \boldsymbol{n} = \\
  &\quad \boldsymbol{M}\boldsymbol{M}^{\boldsymbol{k}\phi(\boldsymbol{n})} \bmod \boldsymbol{n} = \\
  &\quad \boldsymbol{M}\,(\boldsymbol{M}^{\phi(\boldsymbol{n})})^{\boldsymbol{k}} \bmod \boldsymbol{n} = \\
  &\quad \boldsymbol{M}\,(\boldsymbol{M}^{\phi(\boldsymbol{n})} \bmod \boldsymbol{n})^{\boldsymbol{k}} \bmod \boldsymbol{n} = \\
  &\quad \boldsymbol{M}\,(1)^{\boldsymbol{k}} \bmod \boldsymbol{n} = \\
  &\quad \boldsymbol{M} \bmod \boldsymbol{n} = \\
  &\quad \boldsymbol{M}
  \end{aligned}$$
- Proof of correctness can be extended to the case when the plaintext $\boldsymbol{M}$ divides $\boldsymbol{n}$

Algorithmic Issues

- The implementation of the RSA cryptosystem requires various algorithms
- Overall
  - Representation of integers of arbitrarily large size and arithmetic operations on them
- Encryption
  - Modular power
- Decryption
  - Modular power
- Setup
  - Generation of random numbers with a given number of bits (to generate candidates p and q)
  - Primality testing (to check that candidates p and q are prime)
  - Computation of the GCD (to verify that e and $\phi(n)$ are relatively prime)
  - Computation of the multiplicative inverse (to compute d from e)

Modular Power

- The repeated squaring algorithm speeds up the computation of a modular power $\boldsymbol{a}^{\boldsymbol{p}} \bmod \boldsymbol{n}$
- Write the exponent $\boldsymbol{p}$ in binary

  $\boldsymbol{p} = \boldsymbol{p}_{\boldsymbol{b}-1}\boldsymbol{p}_{\boldsymbol{b}-2} \cdots \boldsymbol{p}_1\boldsymbol{p}_0$
- Start with

  $\boldsymbol{Q}_1 = \boldsymbol{a}^{\boldsymbol{p}_{\boldsymbol{b}-1}} \bmod \boldsymbol{n}$
- Repeatedly compute

  $\boldsymbol{Q}_{\boldsymbol{i}} = ((\boldsymbol{Q}_{\boldsymbol{i}-1})^2 \bmod \boldsymbol{n})\boldsymbol{a}^{\boldsymbol{p}_{\boldsymbol{b}-\boldsymbol{i}}} \bmod \boldsymbol{n}$
- We obtain

  $\boldsymbol{Q}_{\boldsymbol{b}} = \boldsymbol{a}^{\boldsymbol{p}} \bmod \boldsymbol{n}$
- The repeated squaring algorithm performs $\boldsymbol{O}(\log \boldsymbol{p})$ arithmetic operations

- Example
  - $3^{18} \bmod 19$ ($18 = 10010$)
  - $\boldsymbol{Q}_1 = 3^1 \bmod 19 = 3$
  - $\boldsymbol{Q}_2 = (3^2 \bmod 19)3^0 \bmod 19 = 9$
  - $\boldsymbol{Q}_3 = (9^2 \bmod 19)3^0 \bmod 19 =$ $81 \bmod 19 = 5$
  - $\boldsymbol{Q}_4 = (5^2 \bmod 19)3^1 \bmod 19 =$ $(25 \bmod 19)3 \bmod 19 =$ $18 \bmod 19 = 18$
  - $\boldsymbol{Q}_5 = (18^2 \bmod 19)3^0 \bmod 19 =$ $(324 \bmod 19) \bmod 19 =$ $17{\cdot}19 + 1 \bmod 19 = 1$

Problem

•Compute $9^{100} \bmod 147$

$$9^{100} \equiv_{147} ?,\ 100 = 1\,100\,100_2$$

$$Q_1 = 9^1 \equiv_{147} 9$$

$$Q_2 = 9^2\, 9^1 = 729 = -6 + 5 \times 147 \equiv_{147} -6$$

$$Q_3 = (-6)^2\, 9^0 = 36$$

$$Q_4 = 36^2\, 9^0 = 6^3 \times 6 \equiv_{147} 69 \times 6 = -27 + 3 \times 147 \equiv_{147} -27$$

$$Q_5 = (-27)^2\, 9^1 = (-27 \times 9)\,(-27) \equiv_{147} 51\,(-27) \equiv_{147} -54$$

$$Q_6 = (-54)^2\, 9^0 = 2^2 \times \left(3^3\right)^2 \equiv_{147} 4 \times 27^2 \equiv_{147} -24$$

$$Q_7 = (-24)^2\, 9^0 = 4 \times 12^2 \equiv_{147} 4\,(-3) \equiv_{147} 135$$

$$9^{100} \equiv_{147} 135$$

Modular Inverse

Theorem

Given positive integers $\boldsymbol{a}$ and $\boldsymbol{b}$, let $\boldsymbol{d}$ be the smallest positive integer such that

$$d = ia + jb$$

for some integers $\boldsymbol{i}$ and $\boldsymbol{j}$.

We have

$$d = \gcd(a,b)$$

- Example
  - $a = 21$
  - $b = 15$
  - $d = 3$
  - $i = 3, j = -4$
  - $3 = 3 \cdot 21 + (-4) \cdot 15 = 63 - 60 = 3$

- Given positive integers $\boldsymbol{a}$ and $\boldsymbol{b}$, the extended Euclid's algorithm computes a triplet (d,i,j) such that
  - $d = \gcd(a,b)$
  - $d = ia + jb$
- To test the existence of and compute the inverse of $x \in Z_n$, we execute the extended Euclid's algorithm on the input pair (x,n)
- Let (d,i,j) be the triplet returned
  - $d = ix + jn$

Case 1: $d = 1$

i is the inverse of x in Z_n

Case 2: $d > 1$

x has no inverse in Z_n

Problem

- Compute 117^{-1} in $\mathbf{Z}_{337}$

$$n = p = 337,\ a = 117$$

$$a\,a^{-1} \equiv_{337} 1 \Leftrightarrow 1 = 117\,a^{-1} + 337\,x$$

$$337 = 3\times 117 - 14,\ \ 117 = 8\times 14 + 5,\ \ 14 = 3\times 5 - \underline{1}$$

$$1 = -\underline{14} + 3\times\underline{5} = -\underline{14} + 3\,(\underline{117} - 8\times\underline{14}) = 3\times\underline{117} - 25\times\underline{14}$$

$$= 3\times\underline{117} - 25\times(-\underline{337} + 3\times\underline{117}) = -72\times\underline{117} + \underline{337}\times 25$$

$$\therefore a^{-1} = -72 \equiv_{337} 265$$

Problem

- With $n = 323$ and $e = 5$, find d and decrypt the cipher $c = 300$ (home work)

RSA Review

Alice and Bob want to communicate. They each create two large primes $p, q \geq 1024$ b and then compute

$$n = p\,q,\ \ \phi = (p-1)\,(q-1),\ \ e:\ \gcd(e, \phi) = 1,\ \ d \equiv_{\phi} e^{-1}$$

Both Alice and Bob publish their (e, n) but keep d secret.

RSA Review

If $m \neq 0$ and n_B are not coprimes:

$$d \equiv_{\phi} e^{-1} \Leftrightarrow d\,e = 1 + k\,\phi = 1 + k' \operatorname{lcm}(p-1, q-1)$$

$$\therefore d\,e \equiv_{p-1} 1,\; d\,e \equiv_{q-1} 1$$

Elgamal Cryptosystem

- Public key cryptosystem
- Invented by Taher Elgamal
- Built on modulo arithmetic
- Security built on discrete logarithm problem

Elgamal Cryptosystem

Alice and Bob want to communicate. They each create a large prime p and find a generator (primitive root) g in $\mathbf{Z}_p$. Then pick a random number $x < p - 1$.

$y = g^x \bmod p$

Both Alice and Bob publish their (p, g, y) but keep x secret.

Problem

- Bob publish $(p, g, y) = (13, 6, 2)$ and keeps $x = 5$ secret. Alice generates $k = 5$ and wants to send $m = 5$. Help her!

Alice:

$$a = g^k = 6^5 = 6^2\, 6^2\, 6 \equiv_{13} (-3)\,(-3)\,6 = 54 \equiv_{13} 2$$

$$b = m\, y^k = 5 \times 2^5 \equiv_{13} 5 \times 6 \equiv_{13} 4$$

$$(a, b) = (2, 4)$$

Bob:

$$b\,(a^x)^{-1} = 4\left(2^5\right)^{-1} = 4 \times 6^{-1} \equiv_{13} 4 \times 11 \equiv_{13} = 5 = m$$

Diffie-Hellman Key Exchange

- Key exchange protocol
- Invented by Whitfield Diffie and Martin Hellman
- Built on modulo arithmetic
- Security built on discrete logarithm problem
- Vulnerable to man-in-the-middle attack

Diffie-Hellman Key Exchange

Alice and Bob want to exchange a key over an unsecure channel. Alice and Bob agree on a large prime p and a generator g (primitive root) to $\boldsymbol{Z}_p$. These are public.

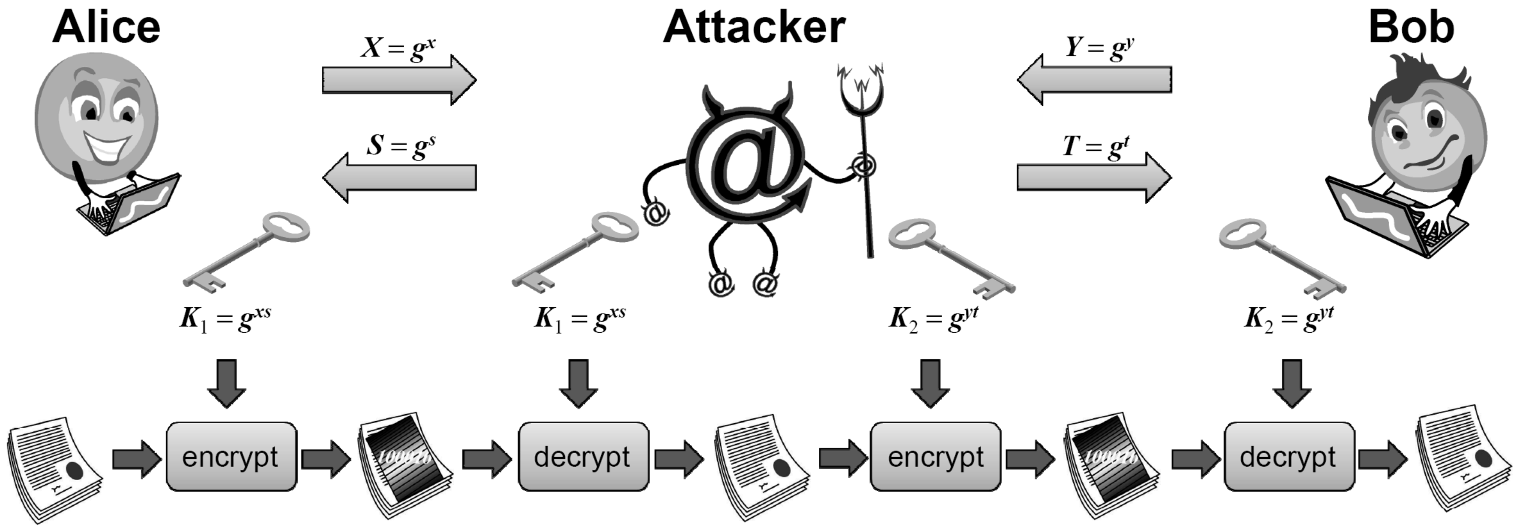


Figure 8.11: The man-in-the-middle attack against the DH protocol. First, by intercepting and modifying the messages of the DH protocol, the attacker establishes a secret key, K_1, with Alice and secret key, K_2, with Bob. Next, using keys K_1 and K_2, the attacker reads and forwards messages between Alice and Bob by decrypting and reencrypting them. Alice and Bob are unaware of the attacker and believe they are communicating securely with each other.

Problem

- With $p = 13$ and $g = 2$, find the common key if Alice and Bob generates the random numbers 3 and 7 respectively.

Alice:

$$X = g^x = 2^3 \equiv_{13} 8$$

$$k = Y^x = 11^3 \equiv_{13} 5$$

Bob:

$$Y = g^y = 2^7 \equiv_{13} 11$$

$$k = X^y = 8^7 \equiv_{13} 5$$

Since we have complete information we can also compute

$$k = g^{xy} = 2^{3\times 7} = 128^3 \equiv_{13} (-2)^3 = -8 \equiv_{13} 5$$

Pseudoprimality Testing

The number of primes less than or equal to n is about $n / \ln(n)$

Thus, we expect to find a prime among $O(b)$ randomly generated numbers with b bits each

Testing whether a number is prime (primality testing) is a difficult problem, though polynomial-time algorithms exist

Example: Rabin-Miller algorithm

Probability of Hitting a Prime

♯primes $\leq x$ is denoted by $\pi(x)$. There are continuous estimates of $\pi(x)$, e.g. Riemann prime counting function $R(x)$. A very simple and well-known estimate is $x/\ln(x)$, which slightly under-estimates $\pi(x)$.

Using this simple estimate the probability of hitting a prime can be computed. Assume the interval is $[0, n) = \left[0, 2^b\right)$, where n consists of b bits. The probability of hitting a prime with an odd random integer is then

$$p_1 = 2\,\frac{\pi(n)}{n} \approx 2\,\frac{n/\ln(n)}{n} = \frac{2}{\ln(n)} = \frac{2}{b\ln(2)}$$

With b independent odd trials we have the following probability of hitting at least one prime:

$$p_b = 1 - (1 - p_1)^b \approx 1 - \left(1 - \frac{2}{b\ln(2)}\right)^b \to 1 - e^{-2/\ln(2)} \approx 0.944$$

Probability of Hitting a Prime

Now if we focus on the interval $[n/2, n) = \left[2^{b-1}, 2^b\right)$ we get the prime hitting probability

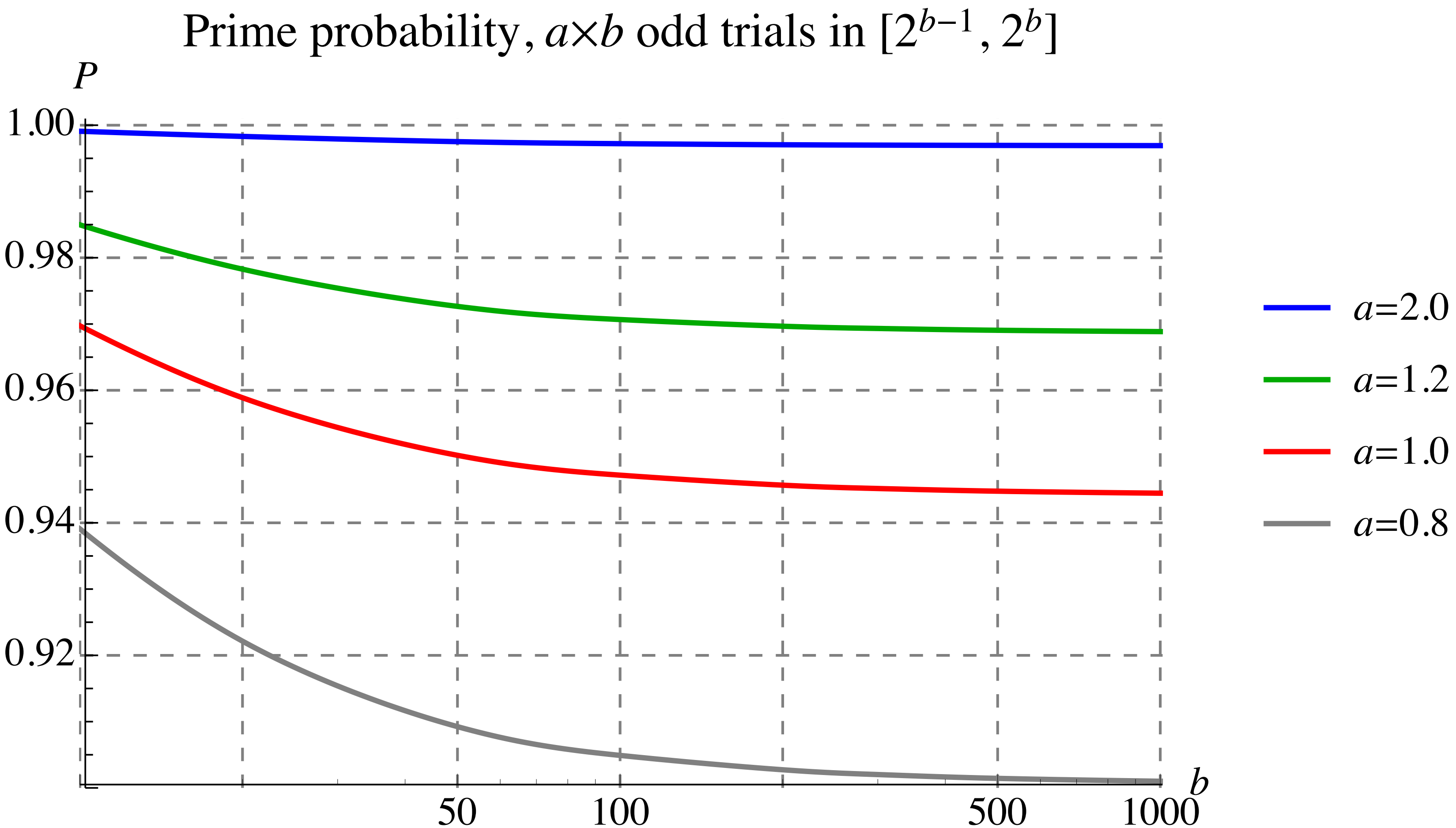
$$p_2 = 2\,\frac{\pi(n) - \pi(n/2)}{n - n/2} \approx 2\,\frac{n/\ln(n) - (n/2)/\ln(n/2)}{n - n/2} = \frac{2}{b\ln(2)}\left(1 - \frac{1}{b-1}\right)$$

With b independent odd trials we have the following probability of hitting at least one prime:

$$p_b' = 1 - (1 - p_2)^b \approx 1 - \left(1 - \frac{2}{b\ln(2)}\left(1 - \frac{1}{b-1}\right)\right)^b \to 1 - e^{-2/\ln(2)} \approx 0.944$$

The limit as $b \to \infty$ will be the same as in the $\left[0, 2^b\right)$ case. This means that we have a around 94 % probability of hitting at least one prime with b trials of b bits odd random numbers in the interval $\left[2^{b-1}, 2^b\right)$.

Probability of Hitting a Prime



Summary

- RSA
- Elgamal Cryptosystem
- Diffie-Hellman Key Exchange
- Man-in-the-middle attacks
- Prime numbers

That's all folks!