Codes and Internet Security

The amazing story of how 200 year old results in the theory of numbers form the security basis for billions of dollars in internet commercial transactions today.

by

Charles W. Neville, April 1999

1. Introduction.

"Tenure track position in mathematics at the assistant or associate professor level. Strong preference given to applicants in a field of *applied or applicable mathematics*. No algebraists need apply."

> Job ad in *The Notices of the American Mathematical Society*, circa 1977.

The very year that ads like this were appearing, Rivest, Shamir and Adelman filed their patent on the fundamental method used to secure billions of dollars in internet commercial transactions. Their method makes essential use of the abstract algebra and elementary number theory deemed so useless by so many. This is the story of how that came to be.

2. Outline

1. Codes and Cryptography.
2. Internet Security.
3. Algebra and Number Theory.
4. The RSA Method.
5. How Secure Is the RSA Method?

Part 1. Codes and Cryptography

A Brief Introduction to Various Sorts of Codes.

1. Symmetric Ciphers

The classical situation: Knowing the key and the method, one can both encode and decode.

Simplest Example: A Simple Substitution Cipher like

$$\begin{array}{l} \mathrm{A} \longleftrightarrow \mathrm{T} \\ \mathrm{B} \longleftrightarrow \mathrm{P} \\ \mathrm{C} \longleftrightarrow \mathrm{X} \end{array}$$

$$\mathrm{CAB} \longleftrightarrow \mathrm{XTP}$$

2. Symmetric Ciphers

Other Examples:

1. The WWII German Enigma Code.
2. IBM's Lucifer Cipher.
3. The U.S. Data Encryption Standard (DES).

These are Much More Complex and Much Harder to Break than a simple substitution cipher, but they share with it the following trait: If an enemy manages to steal the encoding key and the method, he or she can easily decode your messages. Thus Key Security is of paramount importance in all these methods.

3. Asymmetric or Public Key Ciphers

A recent (1970's) development: There are two keys,

1. A Public Encryption Key.
2. A Separate Private Decryption Key.

You may let the whole world know the method and your Public Key so they may send you coded messages.

You must keep your private key very secret. An enemy can learn your Public Key and the method (perhaps because he or she sent you a message, but he Cannot Use This Knowledge to Decode Messages sent to you by others.

4. The Foundations of Public Key Cryptography

Public key cryptographic methods are based on a recent (1970's) development in theoretical computer science, the notion of Computational Complexity.

Key Idea: Some problems are Very Hard. Base your cryptographic method on a Very Hard problem.

Sample Hard Problem: Factorization.

This is probably easy: Factor 323. Answer: 17×19.

5. The Foundations of Public Key Cryptography

But how about this one: Factor

7406887751585867569251795143059236193447477077486722
8197406579496917297622889002203758802524412805681033
6642783314685956495693901714336056843776952571316733
9000549531257469006228006245716108881002895059577

6. The Foundations of Public Key Cryptography

I'll bet you would be hard pressed to come up with

7406887751585867569251795143059236193447477077486728
1974065794969172976228890022037588025244128056810366
4278331468595649569390171433605684377695257131673
9000549531257469006228006245716108881002895059 57

=

1509402497293445260998365996277047451139493435867383
880425876691549588470411353603813444238679891122 1

×

490716542795402777810805959749878926941176558019949
0474427239837091547927832034451262331586358355121 7

7. The Foundations of Public Key Cryptography

By the way, here's the Mathematica code I used to find the above:

```
p = Random[Integer, {10^100, 10^101}];
While[!PrimeQ[p], p = p+1];
Print[p];
q = Random[Integer, {10^100, 10^101}];
While[!PrimeQ[q], q = q+1];
Print[q];
Print[p*q];
```

Note the direction: I found two large primes and multiplied them together to get my 200 digit challenge number. Factoring was easy for me because I already knew the answer. Was it easy for you?

8. The Foundations of Public Key Cryptography

To review, I found two large primes, we'll call them p and q, and I multiplied them together to get my 200 digit number p×q. This is easy, at least with a computer. But it is Very Hard to go in the reverse direction and factor a given 200 digit number into the product of two 100 digit primes. It is Very Hard even with a computer. In fact, it would probably take a fast workstation a billion years to do it.

This asymmetry – multiplication is easy, factorization is not, forms the basis of the

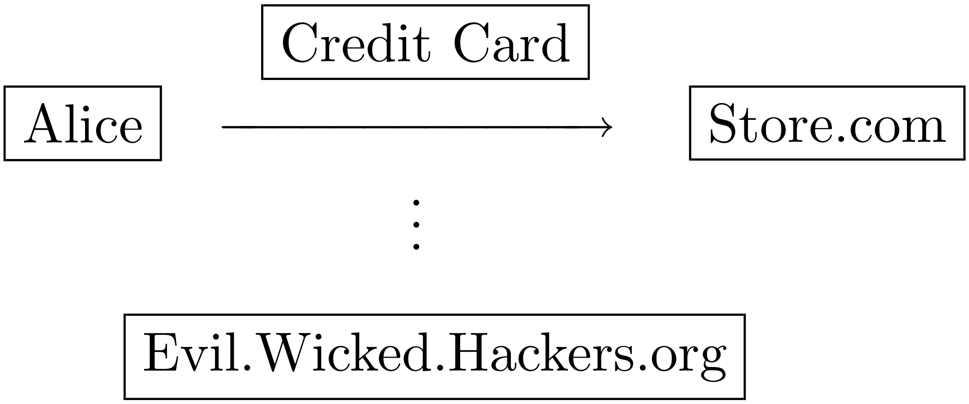
RSA Public Key Cryptography Method.

And the RSA method solves the Key Security problem.

Part 2. Internet Security

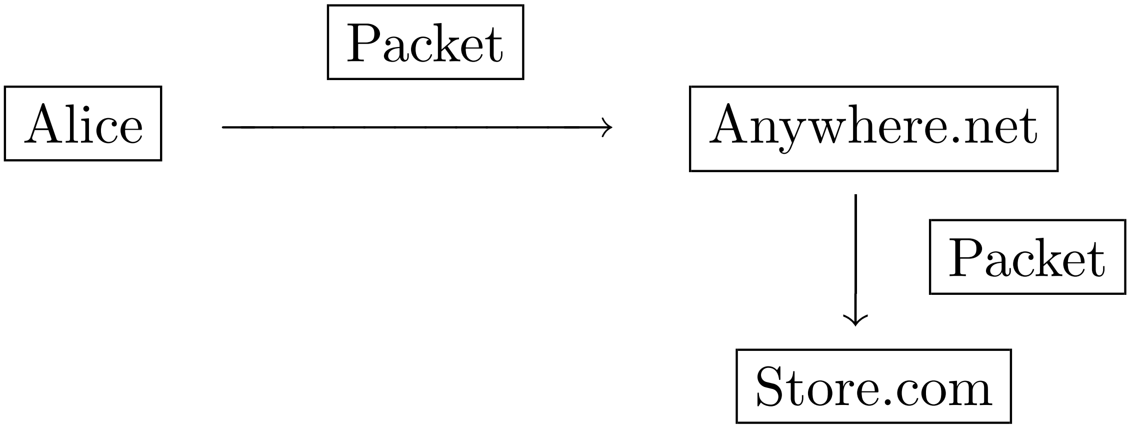
A Brief Introduction to the Need for Internet Security

1. The Need for Internet Security



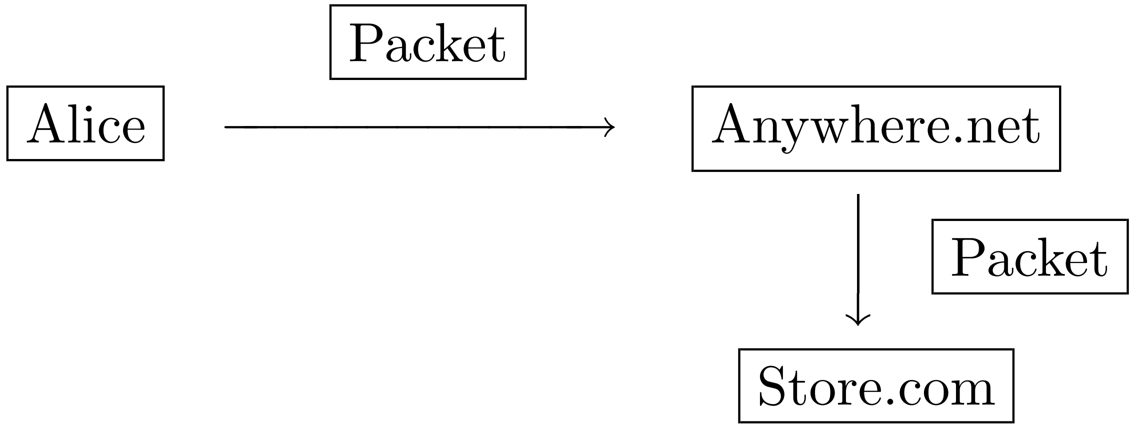
It is easy and legal to listen in to internet transactions because the internet is a Dynamically Routed Packet Switched Network. Thus we have to Encrypt transactions we want to keep confidential.

2. Digression – Packet Switched Networks



Unlike telephone calls, there is no direct dedicated connection for internet messages. Instead, messages are divided into packets, and each packet is dynamically routed from source to destination.

3. Digression – Packet Switched Networks



Packet

Alice

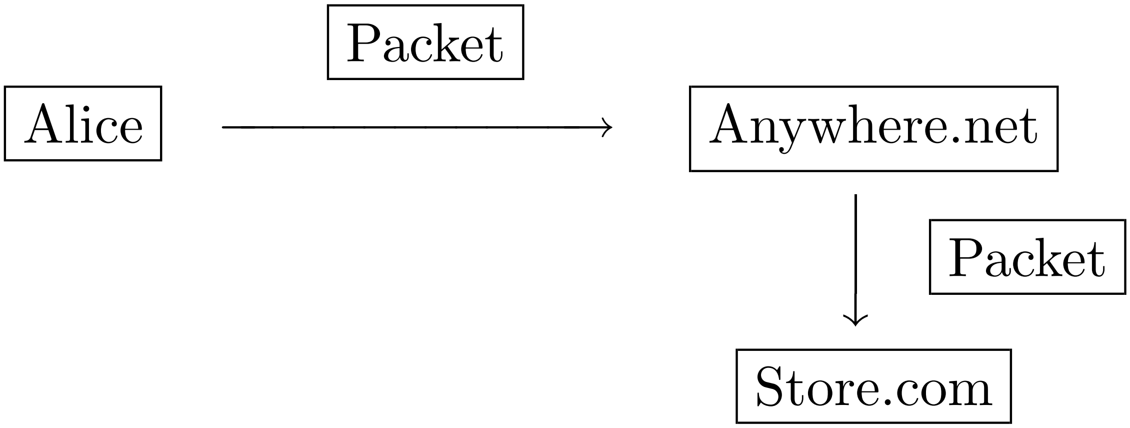
Anywhere.net

Packet

Store.com

Messages are divided into packets, and each packet is dynamically routed from source to destination. One packet may go via London, another through Boston, all part of the same message from here to LA.

4. Digression – Packet Switched Networks



Packet

Alice

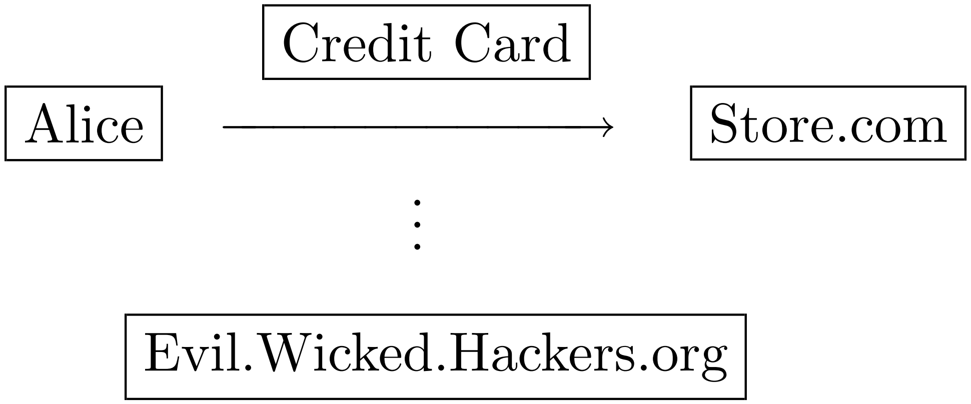
Anywhere.net

Packet

Store.com

Packets are dynamically routed according to available intermediate points. The intermediate points Have To Listen In to forward their packets. This makes it easy and legal for anyone to listen in.

5. Reprise – The Need for Internet Security



Credit Card

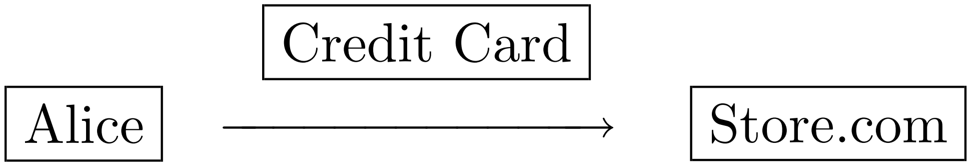
Alice

Store.com

Evil.Wicked.Hackers.org

It is easy and legal to listen in to internet transactions because the internet is a Dynamically Routed Packet Switched Network. Thus we have to Encrypt transactions we want to keep confidential.

6. Securing Internet Transactions



Credit Card

Alice

Store.com

Typical Protocol –

Both use the DES (U.S. Data Encryption Standard) to encode transmissions.

Alice's web browser generates a random DES key which is sent to Store.com.

7. Need for Public Key Cryptography

$$\boxed{\text{Alice}} \quad \xrightarrow{\boxed{\text{Credit Card}}} \quad \boxed{\text{Store.com}}$$

Typical Protocol continued –

Question: “*Sed quis custodiet ipsos Custodes*?” Who shall guard the guardsman? (Juvenal, circa 60 – 130 AD). How do we guard the DES key?

Answer: We encode the key itself with a code so secure that even the nasty folks at Evil.Wicked.Hackers.org can’t break the code, Even If They Intercept the Key!

8. Public Key Cryptography – The RSA Method

Credit Card
Alice
Store.com

Typical Protocol continued –

The server at Store.com generates a random public RSA key and sends it to Alice. Alice uses Store.com's public key to encode her DES key transmission.

The nasty folks at Evil.Wicked.Hackers.org are stymied, because they can't break the RSA code, even knowing the RSA public key, to get at the DES key.

9. Public Key Cryptography – The RSA Method

Review Question –

How does it work?

Answer –

The public RSA key is (essentially) the product of two Very large prime numbers (say of 100 or more decimal digits each). To break the RSA code (it is thought), the folks at Evil.Wicked.Hackers.org have to Factor the key. This is so computationally difficult that it will take them millions of years to do it.

Part 3. Algebra and Number Theory

The Mathematics behind the RSA Method

1. Greatest Math Hits of 300 BC

As part of his *Elements*, Euclid describes the algorithm involving successive remainders that we now know as *The Euclidean Algorithm*, for finding the greatest common divisor of two numbers.

Two consequences: Suppose $N \in \mathbb{Z}$, the set of integers, and $N \neq 0$.

a. Given a number $e \in \mathbb{Z}$, there exists another number $d \in \mathbb{Z}$ such that $ed = 1 \bmod N$ if and only if the greatest common divisor of e and N is 1.

b. By reading the process of applying the Euclidean algorithm backwards, one can find d given e.

2. Greatest Math Hits of 1750

Leonhard Euler figures out the laws of exponents for modular arithmetic:

$$\begin{aligned} k^m &\neq k^n, \quad \text{if } n = m \bmod N \\ k^m &= k^n, \quad \text{if } n = m \bmod \phi(N) \end{aligned}$$

This is more commonly stated as Euler's Theorem:

$$k^{\phi(N)} = 1 \bmod N$$

Here, $\phi(N)$ is the Euler ϕ function, $\phi(N) =$ the number of integers coprime to N in the range $1 \ldots N$, and, of course, k must be coprime to N in the above.

3. More Math Hits from 1750

Leonhard Euler also figures out how to express $\phi(N)$ as a formula:

Def: $\phi(N) =$ the number of integers coprime to N in the range $1 \dots N$.

Theorem: $\phi(N) = N(1 - 1/p_1)(1 - 1/p_2) \cdots (1 - 1/p_k)$, where $p_1, p_2, \dots p_k$ are the distinct prime factors of N.

In particular, if $N = pq$, where p and q are distinct primes, then $\phi(N) = (p-1)(q-1)$. Two hundred years later, Rivest, Shamir and Adelman will make essential use of this in their public key encryption method, and you and I will use it every time we buy something over the internet.

4. Proof Sketch of the $\phi(N)$ Formula

Recall the *inclusion – exclusion principle* from combinatorics:

$$\left|\bigcup A_i\right| = \Sigma|A_i| - \Sigma|A_i \cap A_j| + \Sigma|A_i \cap A_j \cap A_k| \cdots$$

Here $|A|$ is the number of elements in the set A.

Consider the definition of $\phi(N)$, $\phi(N) = |A|$, where the set $A = \{k : k \text{ is coprime to } N, 1 \leq k \leq N\}$. Obtain the formula for $\phi(N)$ by applying the inclusion – exclusion principle to the complement of A. Here, $A_i =$ will be the set of multiples of i in the range $1 \ldots N$, and the subscript i will run over the set of integers which evenly divide N (not including 1 and N).

5. Greatest Math Hits of 1790

Joseph-Louis Lagrange writes one of the first treatises on *group theory*, and he proves what today is known as

Lagrange's Theorem: The order of a subgroup divides the order of the group.

Proof Sketch of Euler's Theorem: Consider the set U_N of invertible elements (with respect to multiplication) in $\mathbb{Z}_N$, the ring of integers mod N. The set U_N forms an Abelian group under multiplication. The set U_N consists exactly of the (equivalence classes of) integers coprime to N. Thus, the order of the group U_N is $\phi(N)$. If $[k] \in U_N$, the order of the subgroup generated by $[k]$ divides the order of U_N, by Lagrange's Theorem. Therefore, $k^{\phi(N)} = 1 \bmod N$.

6. Do We Really Need Quotient Objects?

No, there is a very clever and more elementary group theoretic proof of Euler's theorem that uses simply proved facts about Abelian groups, and which avoids the use of Lagrange's theorem, and thus the use of quotient groups. (cf. Biggs, *Discrete Mathematics*.)

However, we are headed towards the applications of number theory and abstract algebra to public key cryptography. The most recent methods, such as elliptic curve methods, are heavily algebraic, and require an enormous amount of abstract algebra and algebraic geometry.

7. Finding Large Primes

Fermat's Theorem (circa 1650): *Let p be a prime. Then $k^{p-1} = 1$ mod p, for every integer k between* 1 *and* $p - 1$.

Proof: $\phi(p) = p - 1$. Apply Euler's theorem.

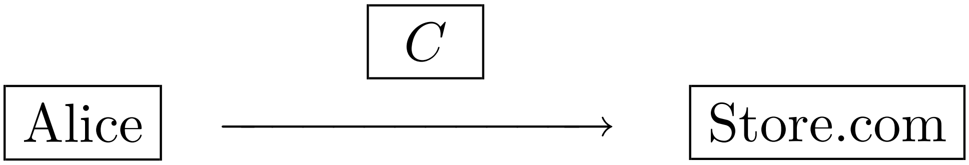
Application (Fermat): To find a large prime, pick a large candidate prime p at random. Pick several integers k and test if $k^{p-1} = 1$ mod p . If so, p is probably prime. If not, pick another large candidate prime at random and repeat the process.

Modern methods, such as the Rabin – Solovay – Strassen algorithm actually used by your favorite internet store's server, are elaborations of Fermat's method.

Part 4. The RSA Method

The RSA Method Details
Or
Where the Math Is Used

1. The RSA Method Details



C

Alice

Store.com

$N = pq$, $\quad p$ and q large primes,
e coprime to $\phi(N) = (p-1)(q-1)$.

Store.com's Public Key Pair: (N, e).

M Alice's message, C the coded message,
$C = M^e \bmod N$. (e is called the *encoding exponent*.)

Alice sends C to Store.com.

2. The RSA Method Details

$$\begin{tikzcd}\boxed{\text{Alice}} \arrow[r, "\boxed{C}"] & \boxed{\text{Store.com}}\end{tikzcd}$$

Store.com's Private Key: d, chosen so $ed = 1 \bmod \phi(N)$.
Store.com decodes Alice's message using the formula,

$$M = C^d$$

(d is called the *decoding exponent*.)

3. Where Euler's 1750 Theorem Is Used

$$\boxed{\text{Alice}} \quad \xrightarrow{\boxed{C}} \quad \boxed{\text{Store.com}}$$

It is why Store.com's decoding method works:

$$\begin{aligned}
ed &= 1 \bmod \phi(N) \\
ed &= 1 + k\phi(N), \quad \text{where } k \text{ is an integer.} \\
C^d &= (M^e)^d = M^{ed} = M^{1 + k\phi(N)} \\
&= M \cdot (M^{\phi(N)})^k = M \cdot 1^k = M,
\end{aligned}$$

since, by Euler's theorem, $M^{\phi(N)} = 1$.

4. Where Euclid's 300 BC Algorithm Is Used

$$C = M^e, \quad M = C^d$$

It is how Store.com finds the private decoding exponent d:

Store.com knows the two large prime factors of N, p and q, so it can easily compute $\phi(N) = (p-1)(q-1)$. After choosing its public encoding exponent e, Store.com applies the *Euclidean Algorithm* to compute d so that $ed = 1 \bmod \phi(N)$.

5. Where Fermat's 1650 Little Theorem Is Used

It forms a large part of the basis of modern methods for testing whether a given number is prime.

Example methods: The one implemented in Mathematica's PrimeQ function we used, or in Store.com's Secure Sockets Protocol software.

Important use: To find the two large randomly generated primes p and q needed for the RSA method.

To find a large prime: Generate a large random number. Test consecutive integers until we find one that is prime, just as we did using Mathematica, or as Store.com's Secure Sockets Protocol software does.

6. Possible Flaws in the RSA Method

$$C = M^e, \quad M = C^d$$

There is a remote chance that the decoding method

$$M = C^d$$

will fail because M is not coprime to N. (Remember, Euler's Theorem requires this.) But we can ignore this possibility in the real world because it is so improbable.

Exercise: Use inclusion-exclusion to show the probability of failure is $(p + q - 1)/(p - 1)(q - 1)$

7. The Security of the RSA Method

$$C = M^e, \quad M = C^d$$

Why you can't compute d even knowing N and e:

If $ed = 1 \bmod N$, it would be easy for you compute the private decoding exponent d from the public N and e. But, $ed = 1 \bmod \phi(N)$, and $\phi(N) = (p-1)(q-1)$. It is strongly believed that computing d is equivalent to factoring N (remember $N = pq$), and it is strongly believed that factoring N is a very hard problem.

Part 5. How Secure Is the RSA Method?

Example RSA Messages.

A Challenge Message Broken.

How Big Is Big?

How Secure Is RSA?

1. Illustration: The First RSA Message

(Rivest, Shamir and Adelman, *Comm ACM*, 1978.)

Primes: $p = 47, \quad q = 59$ (a toy example!)
Calculations:

$$N = pq = 2773, \quad \phi(N) = (p-1)(q-1) = 2668$$
$$e = 17, \quad d = 157$$

Public key: $N = 2773, \quad e = 17$ (for encoding).
Private key: $d = 157$ (for decoding).

2. The First RSA Message – continued

Alphabet encoding: Space = 00, A = 01, B = 02, . . .

$$\begin{aligned} M &= \text{“ITS ALL GREEK TO ME”} \\ &= \ 0920\ 1900\ 0112\ 1200\ 0718 \\ &\ \ 0505\ 1100\ 2015\ 0013\ 0500 \end{aligned}$$

(Note the division into blocks of 4 digits, so each block is $< N$.)

3. The First RSA Message – continued

To encode, $e = 17$, so

$$\begin{aligned} C_1 &= 0920^{17} = 0948 \bmod 2773, \\ C_2 &= 1900^{17} = 2342 \bmod 2773, \ldots \\ C &= \ 0948\ 2342\ 1084\ 1444\ 2663 \\ & \quad\ 2390\ 0778\ 0774\ 0219\ 1659 \end{aligned}$$

To decode, $d = 157$, so

$$\begin{aligned} M_1 &= 0948^{157} = 0920 \bmod 2773 \\ M_2 &= 2342^{157} = 1900 \bmod 2773, \ldots \end{aligned}$$

4. The First Real RSA Message

(RSA-129 Challenge Message, *Scientific American*, 1977.)

$$\begin{aligned} C = {} & 9686\ 9613\ 7546\ 2206\ 1477\ 1409\ 2225\ 4355 \\ & 8829\ 0575\ 9991\ 1245\ 7431\ 9874\ 6951\ 2093 \\ & 0816\ 2982\ 2514\ 5708\ 3569\ 3147\ 6622\ 8839 \\ & 8962\ 8013\ 3919\ 9055\ 1829\ 9451\ 5781\ 5154 \\ N = {} & 114{,}381{,}625{,}757{,}888{,}867{,}669{,}235{,}779{,}976, \\ & 146{,}612{,}010{,}218{,}296{,}721{,}242{,}362{,}562{,}561, \\ & 842{,}935{,}706{,}935{,}245{,}733{,}897{,}830{,}597{,}123, \\ & 563{,}958{,}705{,}058{,}989{,}075{,}147{,}599{,}260{,}026, \\ & 879{,}543{,}541 \end{aligned}$$

5. The First Real RSA Message – continued

And $e = 9007$.

Published in Martin Gardner's "Mathematical Recreations" column in *Scientific American*, August, 1977, as a challenge problem.

Known as RSA-129, because the encryption modulus N was a 129 (decimal) digit number.

Rivest estimated it would take "40 Quadrillion Years" to factor N and break the code.

6. RSA-129 Broken

In the summer of 1993, A. Lenstra organized a team of 600 volunteers and 1600 machines from All over the internet to tackle the RSA-129 challenge using a new *elliptic curve* factorization method.

On April 26, 1994, they Broke The Code!

The RSA-129 challenge message is,

THE MAGIC WORDS ARE SQUEAMISH OSSIFRAGE

Not 40 quadrillion years, but 8 Months!

7. How Big Is Big?

Key Length in bits vs Time for all 100 million Pentium computers sold in 1995, working together, to break the key. Divide all these figures by 2 to 5 today (1999).

Key Length	Time to Break
429 bits	14.5 sec
512 bits	22 minutes
700 bits	153 days
1024 bits	280,000 years

Source: S. Garfinkel, *PGP*: *Pretty Good Privacy*, O'Reilly, 1995.

8. How Random Is Random?

The two primes p and q whose product forms the encryption modulus N must be chosen Randomly.

If you could make even imperfect predictions about the primes the encoding software's Random Number Generator makes, you might be able to break the code.

So encryption software *times your keystrokes* to get a start (provide a seed) for the Random Number Generator.

Is it random? "Beats me, but people think it is."
(S. Garfinkel, 1995)

9. How Secure Is RSA?

Answer: Empirically, very secure. RSA has remained unbroken (for large keys) for 20 years. (But then, again, if the National Security Agency broke it, they wouldn't tell us.)

Answer: Exactly and beyond doubt, we don't know. The security of RSA rests on these assumptions, None Of Which Have Ever Been Proved:

1. There is no fast algorithm for solving problems in the class of maximally hard problems in which *factorization of large numbers* lies. (This is the famous, *Is* $NP <> P$ *problem.*)

10. How Secure Is RSA?

Unproved Assumptions on which RSA rests – continued

2. There is no fast algorithm for factoring large numbers. We don't know of any, but we Cannot Prove that no such algorithm exists. Can you find one?

3. Breaking the RSA code is *equivalent to the factorization problem*. Even this, the weakest of the assumptions underlying the RSA method, has never been proved. So it remains possible that you can find a method for breaking the RSA code *without factoring* the encryption modulus N. Why don't you try? (But don't be disappointed if you don't succeed.)

11. Newer Methods

The RSA method is over 20 years old.

More recent methods, areas of Active Research:

Elliptic Curve Encryption.

Advantages: Shorter keys
Disadvantages: Relatively untested, could be breakable.
Characteristics: Highly algebraic, based on *Algebraic Geometry*, and integer points on *Elliptic Curves*, that is curves of the form

$$y^3 = ax^2 + bx + c,$$

often in a large *Finite Field*.

Part 6. References

Additional Reading on

1. Discrete Mathematics
2. Finding Large Primes
3. Public Key Cryptography
4. The RSA Method
5. Elliptic Curve Encryption
6. Internet Security and the Secure Sockets Layer
7. PGP: Pretty Good Privacy
   – a public domain RSA package

1. References

N. Biggs, *Discrete Mathematics*, Oxford, 1987.

W. Diffie and M. Hellman, “New Directions in Cryptography”, *IEEE Trans. Info. Theory* 22 (1976), pp. 644–654.

M. Gardner, “A New Kind of Cipher that Would Take Millions of Years to Break”, “Mathematical Recreations” column, *Scientific American*, August, 1977.

S. Garfinkel, *PGP: Pretty Good Privacy*, O’Reilly, 1995.

J. Hunter and W. Crawford, *Java Servelet Programming*, O’Reilly, 1998.

N. Koblitz, *Algebraic Aspects of Cryptography*, Springer Verlag, 1998.

2. References – continued

M. Rabin, “Probabilistic Algorithms”, in J. Traub (ed.), *Algorithms and Complexity*, Academic Press, 1976, pp. 21–39.

R. Rivest, A. Shamir and L. Adelman, “Cryptographic Communications System and Method”, U.S. Patent # 4,405,829, filed 1977, granted 1983.

R. Rivest, A. Shamir and L. Adelman, “A Method for Obtaining Digital Signatures and Public Key Cryptography”, *Comm ACM* 21 (1978), pp. 120–126.

R. Solovay and V. Strassen, “A Fast Monte Carlo Test for Primality”, *SIAM J. Comp.* 6 (1977), pp. 84–85. (Erratum, 7 (1978), p. 118.)

3. References – continued

D. Welsh, *Codes and Cryptography*, Oxford, 1990.

Web Sites:
http://www.rsa.com

Best Starter Book:
S. Garfinkel, *PGP*: *Pretty Good Privacy*, O'Reilly, 1995.

Begin Here for Very Serious Study:
D. Welsh, cited above.

4. Additions to the Web Version, (February 2002)

Secure Sockets Layer References:

http://home.netscape.com/eng/ssl3/draft302.txt

http://www.openssl.org

Notes:

The Secure Sockets Layer actually allows the use of a number of public key encryption methods in addition to the RSA method. The Diffie Hellman method, which involves the discrete logarithm, is a frequently used choice.