Zero Knowledge Proofs

Steven G. Krantz

July 25, 2007

1 Basics and Background

Modern security considerations make it desirable for us to have new types of encryption schemes. It is no longer enough to render a message so that only the intended recipient can read it (and outsiders cannot). In today's complex world, and with the advent of high-speed digital computers, there are new demands on the technology of cryptography. The present brief paper will discuss some of these considerations.

In the old days (beginning even with Julius Caesar), it was enough to have a method for disguising the message that we were sending. For example, imagine that the alphabet is turned into numeric symbols by way of the scheme

$$A \longmapsto 0$$

$$B \longmapsto 1$$

$$C \longmapsto 2$$

and so forth.

Then use the encryption

$$n \longmapsto n + 3 \bmod 26\,. \qquad (\star)$$

And now convert these numbers back to roman letters. As a simple example, the phrase

WHAT ME WORRY?

translates to the string of integers

22 7 0 19 12 4 22 14 17 17 24

Notice that it is common, in elementary cryptography, to ignore punctuation and spaces.

The encryption ($\star$) turns this string of integers into

$$25 \quad 10 \quad 3 \quad 22 \quad 15 \quad 7 \quad 25 \quad 17 \quad 20 \quad 20 \quad 2$$

and this, in turn, transliterates to

$$\text{ZKDWPHZRUUC} \qquad (\star)$$

It is clear that anyone receiving the message ($\star$) would have no idea what it means—nor even what the message was about, or what context it fits into. On the other hand, such an encoded message is pretty easy to decrypt. Especially if the decrypter knows that we have used a simple "shift algorithm" to encode the message, and if in addition he/she knows that the most commonly used letters in the alphabet are E and then T, then it would be a fairly simple matter to reverse-engineer this encryption and recover the original message.

Today life is more complex. One can imagine that there would be scenarios in which

(1) You wish to have a means that a minimum-wage security guard (whom you don't necessarily trust) can check that people entering a facility know a password—but you don't want him to know the password.

(2) You wish to have a technology that allows anyone to encrypt a message—using a standard, published methodology—but only someone with special additional information can decrypt it.

(3) You wish to have a method to be able to convince someone else that you can perform a procedure, or solve a problem, or prove a theorem, without actually revealing the details of the process.

This may all sound rather dreamy, but in fact—thanks to the efforts and ideas of R. Rivest, A. Shamir, and L. Adleman—it is now possible. The so-called RSA encryption scheme is now widely used. For example, the e-mail messages that I receive on my cell phone are encrypted using RSA.

Banks, secure industrial sites, high-tech government agences (for example, the National Security Agency), and many other parts of our society routinely use RSA to send messages securely.

In this discussion, we shall describe how RSA encryption works, and we shall encrypt a message using the methodology. We shall describe all the mathematics behind RSA encryption, and shall proved the results necessary to flesh out the theory behind RSA. We shall also describe how to convince someone that you can prove the Riemann hypothesis—without revealing any details of the proof. This is a fascinating idea—something like convincing your mother that you have cleaned your room without letting her have a look at the room. But in fact the idea has profound and far-reaching applications.

2 Preparation for RSA

Background Ideas

We now sketch the background ideas for RSA. These are all elementary ideas from basic mathematics. It is remarkable that these are all that are needed to make this profound new idea work.

Computational Complexity

• Suppose that you have a deck of N playing cards and you toss them in the air. Now you want to put them back into their standard order. How many "steps" will this take? [We want to answer this question in such a manner that a machine could follow the instructions.]

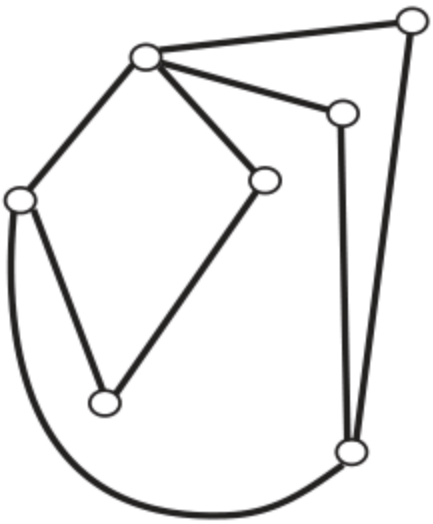
First we look through all N cards and find the first card in the ordering.

Then we look through the remaining $N-1$ cards and find the second card in the ordering.

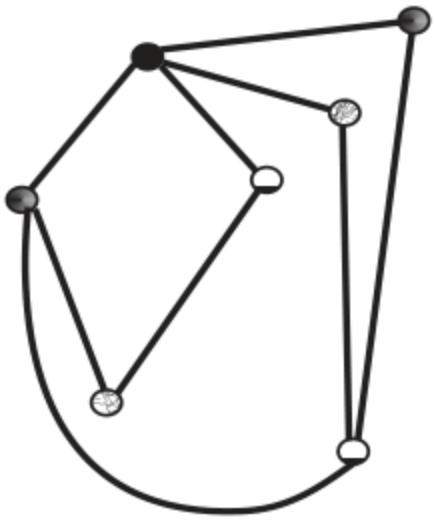
And so forth.

So the re-ordering of the cards takes

$$N + (N-1) + (N-2) + \cdots 3 + 2 + 1 = \frac{N(N+1)}{2}$$



A typical planar graph.



An admissible coloring for the planar graph.

Figure 1

steps. Notice that this answer is a quadratic polynomial in N. Thus we say that the problem can be solved in polynomial time.

• We have heard a rumor that the four-color theorem is true. So we have a graph with N vertices and we wish to color each vertex, using either red, yellow, blue, or green. See Figure 1, where we have used shading to suggest the coloring The only rule is that two adjacent vertices (i.e. vertices that are connected by a segment) cannot be the same color.

Of course the number of possible colorings is the number of functions from the set with N objects to the set with 4 objects. That is 4^N. The machine, being as dumb as it is, will simply try all the possible colorings until it finds one that works. Thus we see that the number of steps is now an exponential function of N.

We call this an exponential time problem.

• Another interesting exponential time problem is that of scheduling planes for an airline. If you have n cities and k planes and you take into account different populations, different demands, crew availability, fuel availability, and other factors, then it is easy to convince yourself that this is a problem of exponential complexity. The theory of linear programming can be used to reduce many problems of this kind to polynomial complexity. Linear programming is routinely used by the airlines for this purpose.

• Certainly one of the most famous exponential time problems is the "traveling salesman problem". The issue here is that a certain salesman wants to visit each of n cities precisely once. What is his most efficient path? It is not difficult to discern that there are exponentially many possible paths, and no evident strategy for picking one in any efficient manner.

Modular Arithmetic

This is a familiar idea, and we have already alluded to it earlier. The "right" way to define the idea is with cosets, but we shall content ourselves here with a more informal definition.

When we write $n \bmod k$ we mean simply the remainder when n is divided by k. Thus

$$25 = 1 \bmod 3\,,$$

$$15 = 3 \bmod 4\,,$$

$$-13 = -3 \bmod 5 = 2 \bmod 5\,.$$

It is an important fact—which again is most clearly seen using the theory of cosets—that modular arithmetic respects sums and products. That is,

$$a{+}b \bmod n = a \bmod n {+} b \bmod n \qquad \text{and} \qquad a{\cdot}b \bmod n = (a \bmod n){\cdot}(b \bmod n)\,.$$

We shall use these facts in a decisive manner below.

Fermat's Theorem

Let a and b be two (positive) integers. We say that a and b are relatively prime if they have no common prime factors. For example,

$$72 = 2^3 \cdot 3^2$$

$$175 = 5^2 \cdot 7$$

hence 72 and 175 are relatively prime.

If n is an integer, let $\mathcal{P}(n)$ be the set of integers less than n that are relatively prime to it. Let $\varphi(n)$ be the number of elements in $\mathcal{P}(n)$.

> **Theorem:** If n is a positive integer and k is relatively prime to n then
>
> $$k^{\varphi(n)} = 1 \bmod n\,.$$

Proof: The proof of this result is easy. For the collection $\mathcal{P}(n)$ of numbers relatively prime to n forms a group under multiplication. That is, if a is relatively prime to n and b is relatively prime to n then logic dictates that $a \cdot b$ is relatively prime to n. Now it is a fundamental fact—we cannot prove it here, but see [BMS]—that if a group has m elements and g is an element of the group then g^m is the group identity. Thus any element of the group, raised to the power $\varphi(n)$ (the number of elements in the group) will equal 1 modulo n. □

For later use, it is worth noting that if p, q are prime numbers and $n = p \cdot q$ then $\varphi(n) = (p-1) \cdot (q-1)$.

The reason is that the only numbers less than or equal to n that are not relatively prime to n are $p, 2p, 3p, \ldots q \cdot p$ and $q, 2q, 3q, \cdots (p-1)q$.

There are q numbers in the first list and $p-1$ numbers in the second list. The set $\mathcal{P}(n)$ of numbers relatively prime to n is the complement of these two lists, and it therefore has

$$pq - q - (p-1) = pq - q - p + 1 = (p-1) \cdot (q-1) \equiv \varphi(n)$$

elements.

Relatively Prime Integers

Two integers a and b are relatively prime if they have no prime factors in common. As noted above, for example, 72 and 175 are relatively prime.

It is a fundamental fact of elementary number theory that if a, b are relatively prime then we can find other integers x and y such that

$$xa + yb = 1\,. \qquad (*)$$

For example, we have noted that $a = 72$ and $b = 175$ are relatively prime. The corresponding integers x, y are $x = -17$ and $y = 7$. Thus

$$(-17) \cdot 72 + 7 \cdot 175 = 1\,.$$

One can prove this result using Fermat's theorem above. For, since b is relatively prime to a, thus

$$b^{\varphi(a)} = 1 \bmod a\,.$$

But this just says that

$$b^{\varphi(a)} - 1 = k \cdot a$$

for some integer k. Unraveling this equation gives $(*)$.

In practice, one finds x and y using the Euclidean algorithm (otherwise known as long division).

In the example of 72, 175, one calculates:

$$\begin{aligned} 175 &= 2 \cdot 72 + 31 \\ 72 &= 2 \cdot 31 + 10 \\ 31 &= 3 \cdot 10 + 1\,. \end{aligned}$$

You know you are finished when the remainder is 1.

For now we have

$$\begin{aligned} 1 &= 31 - 3 \cdot 10 \\ &= 31 - 3 \cdot (72 - 2 \cdot 31) \\ &= 7 \cdot 31 - 3 \cdot 72 \\ &= 7 \cdot (175 - 2 \cdot 72) - 3 \cdot 72 \\ &= 7 \cdot 175 - 17 \cdot 72\,. \end{aligned}$$

That is the decomposition we seek.

3 The RSA System Enunciated

Description of RSA

Now we can quickly and efficiently describe how to implement the RSA encryption system, and we can explain how it works.

Imagine that George W. Bush has an important message that he wishes to send to Donald Rumsfeld. Of course Rumsfeld is a highly-placed man of many responsibilities, and you can imagine that Bush's message is quite secret. So he wants to encode the message:

Your time is up. Hasta la vista, baby.

So Bush goes to the library and finds the RSA encryption book. This is a readily available book that anyone can access. It is not secret. A typical page in the book reads like this:

Name	Value of n	Value of e
Puck, Wolfgang	4431 ... 7765	8894 ... 4453
Rehnquist, William	6668 ... 2345	1234 ... 9876
Riddle, Nelson	7586 ... 2390	4637 ... 4389
Rin Tin-Tin	5355 ... 5353	5465 ... 7648
Rogers, Roy	7859 ... 4359	3058 ... 2934
Roosevelt, Theodore	7835 ... 2523	7893 ... 4232
Rotten, Johnny	3955 ... 4343	4488 ... 9922
Roy, Rob	3796 ... 5441	2219 ... 3319
Rumsfeld, Donald	1117 ... 8854	9266 ... 2388
Russert, Tim	6464 ... 4646	3223 ... 3232
Schwarzenegger, Arnold	6894 ... 3242	7525 ... 2314
Simpson, Orenthal James	6678 ... 2234	4856 ... 2223

What does this information mean? Of course we know, thanks to Euclid, that there are infinitely many primes. So we can find prime numbers with

as many digits as we wish. Each number n in the RSA encryption book is the product of two 75-digit primes p and q: Thus $n = p \cdot q$. Each number e is chosen to be a number with at least 100 digits that is relatively prime to $\varphi(n) = (p-1) \cdot (q-1)$. Of course we do not publish the prime factorization of the number n; we also do not publish $\varphi(n)$. All that we publish is n and e for each individual.

Now an important point to understand is that Bush does not need to understand any mathematics or any of the theory of RSA encryption in order to encode his message. [Well, it would be nice if he understood modular arithmetic. But he is, after all, the President of the United States.] All he does is this:

(1) First he breaks the message into units of five letters. We call these "words", even though they may not be English language words.

For the message from Bush to Rumsefeld, the "words" would be

YOURT IMEIS UPHAS TALAV ISTAB ABY

(2) He transliterates each "word" into a sequence of numerical digits, using our usual scheme of translation.

(3) Then he encodes each transliterated word w with the rule

$$w \longmapsto w^e \bmod n\,.$$

Bush will send to Rumsfeld this sequence of encrypted words. That is all there is to it.

The real question now is:

What does it take to decrypt the encoded message? How can Rumsfeld read the message?

This is where some mathematics comes into the picture. We must use Fermat's theorem, and we must use our ideas about relatively prime integers. But the short answer to the question is this. If $\widetilde{w}$ is a word encrypted according to the simple scheme described above, then we decrypt it with this algorithm:

> We find integers x and y so that $xe + y\varphi(n) = 1$ and then we calculate
>
> $$\widetilde{w}^x \bmod n\,.$$

That will give the decrypted word w with which we began. [We shall provide the mathematical details of this assertion in the next section.] Since w has only five characters, and n has 150 characters, we know that $w \bmod n = w$—so there is no ambiguity arising from modular arithmetic. We can translate w back into roman characters, and we recover our message.

Now here is the most important point in our development thus far:

> In order to encrypt a message, we need only look up n and e in the public record RSA encryption book. But, in order to decrypt the message, we must know x. Calculating x necessitates knowing $\varphi(n)$, and that necessitates knowing the prime factorization of n.

It is a theorem that calculating the prime factorization of an integer with k digits is a problem of exponential complexity in k. For an integer with 150 digits, using a reasonably fast computer, it would take several years to find the prime factorization.[1]

4 The RSA Encryption System Explicated

Explanation of RSA

In fact, with all the preliminary setup that we have in place, it is a simple matter to explain the RSA encryption system.

For suppose that we selected an $n = p \cdot q$ and an e relatively prime to $\varphi(n) = (p-1)\cdot(q-1)$ corresponding to a particular person listed in the RSA

[1] A bright student who heard my lecture on this topic pointed out that—if your message is just five letters long—then there are only 26^5 possible encryptions. You could decrypt the message by trial and error. Because of considerations like this, we often find it convenient to append a 50-digit random integer to the message. This technique is discussed in detail near the end of the paper.

encryption book. If we are the certified decrypter, then we know the prime factorization of n—that is, we know that $n = p \cdot q$ for p and q prime.

We therefore know $\varphi(n) = p \cdot q - p - q + 1 = (p-1) \cdot (q-1)$ and so we can calculate the x and the y in the identity $xe + y\varphi(n) = 1$. Once we know x, then we know everything. For

$$\begin{aligned} \widetilde{w}^x \bmod n &= [w^e]^x \bmod n \\ &= w^{ex} \bmod n \\ &= w^{1-y\varphi(n)} \bmod n \\ &= w \cdot [w^{\varphi(n)}]^{-y} \bmod n \\ &= w \cdot 1^{-y} \bmod n \\ &= w \bmod n \\ &= w \end{aligned}$$

since w is certainly relatively prime to n.

This shows how we recover the original word w from the encrypted word $\widetilde{w} = w^e \bmod n$.

5 Zero Knowledge Proofs

How to Keep a Secret

We shall now give a quick and dirty description of how to convince someone that you can prove **Proposition A** without revealing any details of the proof of **Proposition A**. The idea comes across most clearly if we deal again with colorings of graphs.

So suppose once again that we are given a graph. See Figure 2. We are the prover, and there is a remotely located verifier.

It is our job to convince the verifier that we know how to 4-color this graph. But we do not want the verifier to actually know how to color the graph. We only want him to be convinced that we know how to do it.

We begin by transmitting the adjacency matrix to the verifier. This data is exhibited in Table 3.

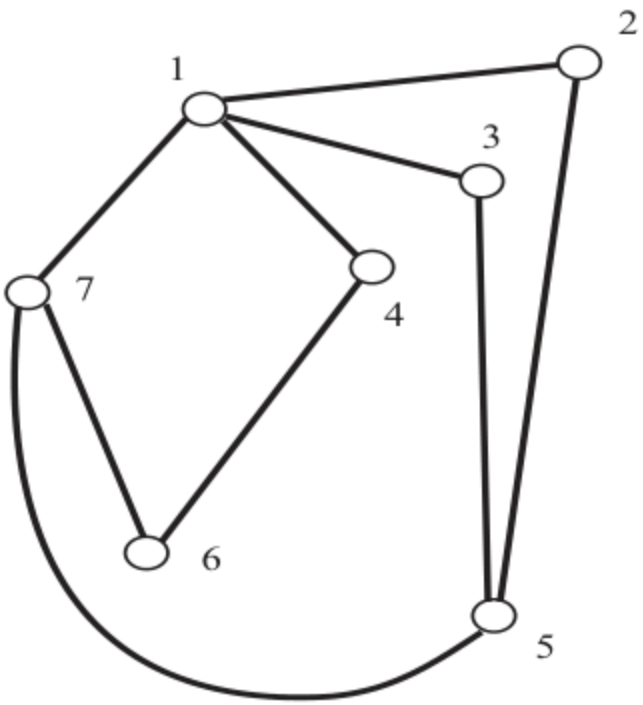


Figure 2

	1	2	3	4	5	6	7
1	x	x	x	x			x
2	x	x			x		
3	x		x		x		
4	x			x		x	
5		x	x		x		x
6				x		x	x
7	x				x	x	x

Table 3

This transmission is straightforward, and need not be encoded. We simply tell the verifier: "In position $(1, 1)$ of the matrix there is an x"; "In position $(1, 2)$ of the matrix there is an x"; In position $(1, 5)$ of the matrix there is no x." And so forth.

Now we number the colors 1, 2, 3, 4. Here

$$\begin{aligned} 1 &\leftrightarrow \text{blue} \\ 2 &\leftrightarrow \text{red} \\ 3 &\leftrightarrow \text{yellow} \\ 4 &\leftrightarrow \text{green} \end{aligned}$$

Finally, imagine that our coloring of the graph is as in Figure 4. Of course we have used shading to suggest the coloring. We wish to communicate to the verifier that we have a valid coloring—in such a manner that he can check it, but he cannot learn any of the details of the coloring.

As we see from Figure 4, the coloring is encoded as

$$12 \quad 23 \quad 34 \quad 41 \quad 51 \quad 64 \quad 73\,.$$

This is read as "node 1 is colored with color 2 (red)," "node 2 is colored with color 3 (yellow), etc. We will transmit these pairs of digits, suitably encoded, to the verifier.

The trouble is that the verifier already knows that there are only seven nodes in the graph, and only four colors, so he could (with a little effort) figure out what color has been assigned to what node just with a little trial

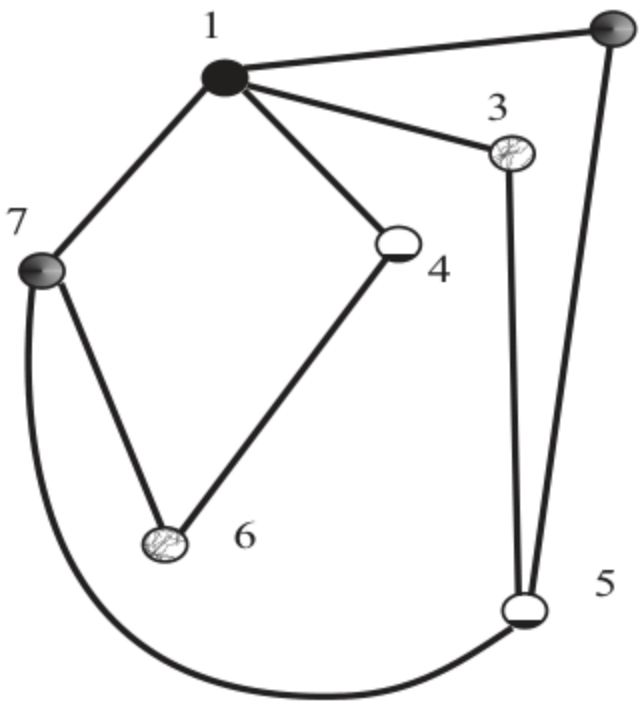


Figure 4

and error—even though the information has been encoded. So this will not do.

Thus, instead of encoding and sending 12 and 23 and 34 and so forth, instead the prover encodes and sends

$$\begin{array}{c} 12\, r_1 \\ 23\, r_2 \\ 34\, r_3 \\ \text{etc.} \end{array} \quad ,$$

where r_1, r_2, r_3 are 50-digit random integers.

More precisely, the step-by-step scenario is this:

(1) The prover sends the entire coloring to the verifier, in encoded form as indicated above.
(2) The verifier stares at his adjacency matrix. He notices that, for example, vertices 4 and 6 are adjacent. And he inquires specifically about those two vertices.
(3) The prover sends the verifier the colorings for those two particular vertices (with a 50-digit random integer attached to each, just as before).
(4) The verifier encrypts the information for those two vertices—using the pre-agreed-upon public key encryption system. He then checks that those two vertex colorings match colorings in the full coloring of the graph that was sent in Step **(1)**.

Now the verifier has checked that one pair of adjacent vertices is suitably colored (i.e., with different colors). If he wants to perform further verifications, then the preceding steps are repeated. Except that first the prover assigns numbers to the four colors red, yellow, blue, and green in some new random fashion. And he chooses an entirely new set of random 50-digit integers. Then he sends the entire colored graph, gets a query from the verifier, and so forth.

If there are n nodes to be colored then there are $n(n-1)/2$ possible pairs of nodes. The probability that the prover *lied* about the coloring and that the verifier—in asking for the coloring of a particular pair of nodes—failed to catch the lie is $1/[n(n-1)]$. If the entire process is iterated again, then the probability that the verifier failed to catch the lie is $1/[n^2(n-1)^2]$. And so forth. Thus each successive verification increases the likelihood that the verifier may be certain of his check.

The point of this procedure is that the verifier can check that any pair of adjacent vertices is colored correctly, that no two adjacent vertices are colored the same, but he cannot amalgamate the information and produce the entire coloring of the graph.

Of course the example we have presented is for graph coloring, just because that is simple to describe. But any proof whatsoever can be translated into binary code and then rendered as a statement about the coloring of some graph. So in fact the example we have given is perfectly general.

Concluding Remarks

The RSA encryption scheme is one of the great ideas of modern coding theory. It is being developed and enhanced even as we speak. There are versions for multiple verifiers, for dishonest provers, and many other variants. The history of RSA is a remarkable one. There was a talk at the 1986 International Congress of Mathematicians in Berkeley about the method. After that, the government attempted to co-opt the method, retract all the preprints, and suppress the information. Interestingly, it was the National Security Agency (the branch of the government in Washington that specializes in cryptography) that stepped in and prevented the government intervention. Now RSA is in the public domain, and anyone can use it. It is a powerful tool.

References

[BMS] G. Birkhoff and S. MacLane, *A Survey of Modern Algebra*, 5th ed., A. K. Peters, Wellesley, MA, 1997.

[BLU] M. Blum, How to prove a theorem so no one else can claim it, *Proc. International Congress of Mathematicians* (Berkeley, CA, 1986), 1444-1451, AMS Providence, RI, 1987.

[BSMP] M. Blum, A. De Santis, S. Micali, G. Persiano, Noninteractive zero-knowledge, *SIAM J. Computing* 20(1991), 1084–1118.

[**BG**] M. Blum, S. Goldwasser, An efficient probabilistic public-key encryption scheme which hides all partial information, Advances in Cryptology (Santa Barbara, CA, 1984), 289–299, *Lecture Notes in Computer Science* 196, Springer-Verlag, Berlin, 1985.

[**FFP**] U. Feige, A. Fiat, and G. Persiano, Noninteractive zero-knowledge proof systems, Advances in Cryptology—CRYPTO '87 (Santa Barbara, CA, 1987), 52–72, *Lecture Notes in Computer Science* 293, Springer-Verlag, Berlin, 1988.

[**FFS**] U. Feige, A. Fiat, and A. Shamir, Zero-knowledge proofs of identity, *J. Cryptology* 1(1988), 77–94.