Applications to Chinese Remainder Theorem

Abstract

We demonstrate the usefulness of a simple mathematical result- the Chinese Remainder Theorem (CRT). A short informal introduction is followed by a formal analysis of the Chinese Remainder Theorem. Further, we discuss its application to a simple *k*-threshold system for secret sharing and for clever RSA variants, namely RSA-CRT and Rebalanced-RSA-CRT.

Introduction

A common math puzzle is to find a positive integer *x* which when divided by 2,3,5 gives remainder 1 and is divisible by 7. Does a solution necessarily exist? If yes, is there more than one solution? Such questions are formally studied using the Chinese Remainder Theorem.

Statement: Given a system of congruences to different moduli:

$x == a_1 \bmod m_1$,

$x == a_2 \bmod m_2$,

. . .

$x == a_r \bmod m_r$.

and if each pair of moduli are relatively prime: gcd $(m_i, m_j)=1$ for $i \neq j$, has exactly one common solution modulo $M=m_1 * m_2 * \ldots * m_r$ and any two solutions are congruent to one another modulo *M*.

Proof: We first prove the existence of a unique solution modulo *M*. Suppose *x*' and *x*'' are two solutions, then $x' == x''$ mod m_i, for each $(0<i<=r)$. i.e. $m_i \mid (x'-x'')$ for each $(0<i<=r)$. Since each m_i are relatively prime in pairs, $M \mid (x'-x'')$ and we have $x' == x''$ mod *M*. Therefore we have a unique solution modulo *M* and any two solutions are congruent to one another modulo *M*. Let $M_i=M/m_i$; which means M_i is the product of all the moduli except for the i^{th} term. Clearly, gcd $(M/m_i; m_i)=1$ and using the Euclidean algorithm, we can find N_i such that $M/m_i * N_i == 1$ mod m_i. We set $x = a_1*(M/m_1)*N1 + a_2*(M/m_2)*N_2 + \ldots + a_r*(M/m_r\}*N_r$. Consider each term in the sum modulo m_i. We get $x ==$ $a_i*(M/m_i\}*N_i == a_i$ mod m_i. All other terms go to zero since $M/m_j == 0$ mod m_i when *i*(not equal)*j*. Hence *x* satisfies all the congruences in the system. We have already shown that x is a unique solution modulo M. Hence the proof.

Note: If the moduli m_1, m_2. . . m_r are not relatively prime in pairs, there may be no solution to the congruence.

E.g. Using Chinese Remainder Theorem solve

x== 2 mod 3,

x== 3 mod 5,

x== 4 mod 11,

x==5 mod 16.

Clearly, the moduli are relatively prime in pairs. M= 3*5*11*16 = 2640. M_1 = 2640/3 = 880, M_2 = 2640/5 = 528, M_3 = 2640/11 = 240, M_4 = 2640/16 = 165.

We have

880*N_1 == 1 mod 3,

520*N_2 == 1 mod 5,

240*N_3 == 1 mod 11,

165*N_4 == 1 mod 16.

Solving using the Extended Euclidean algorithm, we have N_1 = 1, N_2 = 2, N_3 = 5, N_4 = -3.

Therefore, x = 2*880*N_1 + 3*528*N_2 + 4*240*N_3 + 5*165*N_4

= 2*880*1 + 3*528*2 + 4*240*5 + 5*165*(-3)

= 7253 (mod 2640)

= 1973.

We have x = 1973 as a common solution to the above system of congruence. All other solutions are of the form 1973+ M*i, i=1, 2, 3 . . . and so on.

Chinese Remainder Theorem in a *k*-threshold system for secret sharing

Let N be a very large secret integer. To prevent misuse, the secret N is split among n servers nation wide. During an emergency, the partial secrets are collected and worked together to retrieve N. However, if p servers are down (for a number of reasons), N can be retrieved from the remaining $k = n$-p servers, as long as n-p >= k, the threshold value. Such a scheme is called k-threshold system for secret sharing. If we choose p_1, p_2, . . .,p_r to be r different primes such that k^{th} root of $N < p_i$ $_{much}$< k-1th root of N, for each p_i, then the secret N can be unlocked using k servers but not by a fewer number of servers. We illustrate this scheme using an over simplified example.

Let N = 1000, the secret integer. We are to split the secret among 3 servers such that N can be retrieved by working together the partial secrets of 3 servers but not by the participation of fewer servers. We find

cube-root of N < p_i $_{much}$< square-root of N.

cube-root of 1000 < p_i $_{much}$< square-root of 1000.

10 < p_i $_{much}$< 31.6.

We choose p_1=11, p_2=13, p_3=17. Finding the residues of N modulo p_i, we get

x== 10 mod 11,

x== 12 mod 13,

x== 14 mod 17.

With the help of 3 servers, the secret $x = N$ is retrieved as follows:

Find $M = p_1*p_2*p_3$ = 11*13*17 = 2431. M_1 = 2431/11 =221, M_2 = 2431/13 =187, M_3 = 2431/17 = 143.

Using Chinese Remainder Theorem,

221*N_1==1 mod 11, N_1=1.

187*N_2==1 mod 13, N_2=8.

143*N_3==1 mod 17, N_3=5.

x = 10*221*1 + 12*187*8 + 14*143*5

= 30172(mod 2431) = 1000, which is the desired secret.

We now attempt to retrieve the secret N using two servers

x== 10 mod 11,

x== 12 mod 13.

M = 11*13 = 143, M_1 = 143/11 = 13, M_2 = 143/13 = 11.

Using Chinese Remainder Theorem,

13*N_1==1 mod 11, N_1=6.

11*N_2==1 mod 13, N_2=6.

x = 10*13*6 + 12*11*6 = 1572(mod 143) = 142.

All other solutions are of the form 142 + (143*i), i =1, 2, 3 and so on yielding solutions namely: 285, 428, 571, 714, 857, 1000, . . .and so forth. Here p_1 = 11 and p_2 = 13 are $_{much}$< (square-root of N). The only knowledge to an adversary with two partial secrets is that the solution is of the form 142 + (143*i). He will have to guess from the many values of i, when each P_i is $_{much}$< square-root of N. When the values of p_i and N are very large, guessing becomes practically impossible.

With the participation of 3 servers under the constrain that, cube-root of N < each p_i, the entire secret N can be reconstructed. This is because applying the Chinese Remainder Theorem yields a solution modulo ($p_1*p_2*p_3$). We have p_1 > cube-root of N, p_2 > cube-root of N and p_3 > cube-root of N. Hence $p_1*p_2*p_3> N$ and we have a solution of the form $x = x_0 + (p_1*p_2*p_3)*i$. Since $M = p_1*p_2*p_3> N$, we have $x = x_0$ with $i = 0$ as desired.

Chinese Remainder Theorem in RSA-CRT and Rebalanced RSA-CRT

In RSA-CRT, it is a common practice to employ the Chinese Remainder Theorem during decryption. It results in a decryption much faster than modular exponentiation. RSA-CRT differs from the standard RSA in key generation and decryption. The value of d, the secret exponent cannot be made short. As soon as d< $N^{0.292}$, RSA system can be totally broken. Keeping this in mind we make use of the following scheme.

RSA-CRT key generation

1. Let p and q be very be two very large primes of nearly the same size such that gcd (p-1, q-1) = 2.
2. Compute $N = p*q$.
3. Pick two random integers d_p and d_q such that gcd (d_p, p-1)=1, gcd (d_q, q-1)=1 and d_p==d_q mod 2.
4. Find a d such that d==d_p mod p-1 and d==d_q mod q-1.
5. Compute e=d^{-1} (mod Phi (N)).

The public key is <N, e> and the private key is <p, q, d_p, d_q>. Since gcd (d_p, p-1)=1 and d==d_p mod p-1, we have gcd (d, p-1)=1. Similarly, gcd (d, q-1)=1. Hence gcd (d, phi (N))=1 and by step 5, e can be computed.

To apply the Chinese Remainder Theorem in step 4, the respective moduli have to be relatively prime in pairs for a solution to necessarily exist. We observe that p-1 and q-1 are even and that we cannot directly apply the Chinese Remainder Theorem. However, gcd ((p-1)/2, (q-1)/2)=1. Since gcd (d_p, p-1)=1 and gcd (d_q, q-1)=1, essentially d_p, d_q are odd integers and d_p-1, d_q-1 are even integers. We have gcd (d, p-1)=1, which implies that d is odd and d-1 is even.

To find a solution to

d==d_p mod p-1,

d==d_q mod q-1.

We find a solution to

d-1==d_p –1 mod p-1,

d-1==d_q –1 mod q-1.

By applying the cancellation law and taking the common factor 2 out, we have

x=d'== (d-1)/2==(d_p –1)/2 mod(p-1)/2,

x=d'==(d-1)/2==(d_q –1)/2 mod(q-1)/2.

Using Chinese Remainder Theorem we find d such that d = (2*d') +1.

RSA-CRT Decryption

Since RSA-CRT encryption is same as that of the standard RSA encryption procedure, we now turn our attention to RSA-CRT decryption. Let M be the plaintext and C the ciphertext.

Theorem: If C is not divisible by p and d_p==d mod p-1, then C^{dp}==C^d (mod p).

For decryption we find

1. M_p=C^{dp}(mod p)= C^d(mod p) and M_q=C^{dq}(mod q)= C^d(mod q).
2. Then using Chinese Remainder Theorem, we find a solution for

   M=M_p(mod p)= C^d(mod p),

   M=M_q=C^{dq}(mod q)= C^d(mod q).

We now illustrate the scheme using an over simplified example. Choose p = 7, q = 11, gcd (p-1, q-1) = 2, N = p*q = 7*11 = 77, phi (N) = (p-1)*(q-1) = 6*10 = 60.

Let d_p = 5, gcd (d_p , p-1) = gcd (5,6) = 1.

d_q = 3, gcd (d_q , q-1) = gcd (3,10) = 1.

We are to find d such that

d==5 mod 6,

d==3 mod 10.

We cannot apply the Chinese Remainder theorem since gcd (6,10) ≠ 1, hence we convert the system of congruences in such a manner that the cancellation law can be applied

Therefore, we have

d-1==5-1 mod 6,

d-1==3-1 mod 10.

On applying the cancellation law,

(d-1)/2==(5-1)/2 mod (6/2),

(d-1)/2==(3-1)/2 mod (10/2).

x = d'= (d-1)/2== 2 mod 3,

x = d'= (d-1)/2== 1 mod 5.

Solving using Chinese Remainder Theorem,

M = 3*5 = 15, M_1 =15/3 = 5, M_2 = 15/5=3.

$5*N_1$==1 mod 3, N_1=2,

$3*N_2$==1 mod 5, N_2=2.

We have,

d' = x = 2*5*2 + 1*3*2 = 26(mod 15) = 11. Therefore d' = 11 and d = (2*d')+1 = (2*11) +1 = 23, d = 23.

Now we find, e such that

$e*d$==1 mod phi(N),

e*23==1 mod 60, e = 47.

Let the plaintext M=5.

$C=5^{47}$ mod 77 = 3.

For decryption, we find

$M = M_p$ mod $p = c^d$ mod p,

$M = M_q$ mod $q = c^d$ mod q.

$M_p = 3^5$ mod 7 = 243 mod 7 = 5,

$M_q = 3^3$ mod 11= 27 mod 11 = 5.

Using the Chinese Remainder Theorem,

M = 7*11 = 77, M_1 = 77/7 = 11, M_2 = 77/11 = 7.

$11*N_1$==1 mod 7, N_1=2,

$7*N_2$==1 mod 11, N_2=8.

x = 5*11*2 + 5*7*8 = 390 mod 77 =5. Thus $x = M = 5$, as desired. In this specific example (M_p and M_q)=5 is a common solution and it is not necessary to further apply the Chinese Remainder Theorem.

We now turn our attention to another RSA variant, the Rebalanced RSA-CRT. The main aim of Rebalanced RSA-CRT is to speed up RSA decryption by shifting the work to the encrypter. This behavior is particularly useful for RSA decryption in mobile devices like cell phones whose life is limited by its battery. Rebalanced RSA-CRT decryption is over three times faster than the standard RSA. The only difference between RSA-CRT and Rebalanced RSA-CRT is in choosing the values of d_p and d_q. In Rebalanced RSA-CRT, the size of e and d are of the order of phi (N), where as in standard RSA, e is usually a 16-bit or 32-bit integer. According to [6], the size of d_p and d_q should be at least 160-bits to achieve a security of 2^{80}. As a result, for Rebalanced RSA-CRT we always choose (d_p and d_q) > 160-bits. The remaining steps are same as that for RSA-CRT. The main drawback with this scheme is that the task of the encrypter is enormous, even for a high-end computer. A variant of Rebalanced RSA-CRT exists with encryption three times faster than the original Rebalanced RSA-CRT. For more details please refer [7]. The security of these RSA variants is an open question. For a security of 2^{80} against the best-known attacks on Rebalanced RSA-CRT, one should use d_p and d_q of length greater than 160 bits.

Conclusion

We discussed applications of Chinese Remainder Theorem to a *k*-threshold system for secret sharing and its use in clever RSA variants like RSA-CRT and Rebalanced RSA-CRT where decryption is over three times faster than standard RSA, which uses modular arithmetic.

Bibliography

[1]. Neal Koblitz, *A Course In Number Theory and Cryptography*, Springer, Second edition, 1994.

[2]. William Stallings, *Cryptography and Network Security*, Pearson Education, Third Edition.

[3]. Tom Apostol**,** *Introduction to Analytical Number Theory*, Springer International, Student edition, 1989.

[4]. Bruce Schneier, *Applied Cryptography*, Wiley Publications, Second edition, 2001.

[5]. Ivan Niven, Herbert S.Zuckerman, *An Introduction to the Theory of Numbers,* Wiley Eastern Limited.

[6]. Dan Boneh and Hovav Shacham, Winter/Spring 2002. *Fast Variants of RSA*. CryptoBytes-Volume 5, No. 1, Winter/Spring 2002, pg 1-9. Available: http://www.rsasecurity.com/rsalabs/cryptobytes/CryptoBytes_January_2002_final.pdf

[7]. Hung-Min Sun and Mu-En Wu, 2005, February. *An Approach Towards Rebalanced RSA-CRT with Short Public Exponent*. Cryptology ePrint Archive: Report 2005/053, Available: http://eprint.iacr.org/2005/053

[8]. Boneh.D, *Twenty years of attacks on the RSA Cryptosystem,* Notices of the American Mathematical Society, February 1999.

Authored by
Sarad A.V aka Data.