MATRIX METHODS FOR PRIVATE KEY CRYPTOGRAPHY

Frank Rubin
April 23, 2018

Cryptography website

MasterSoftware.biz

Email

MasterSoftware1@aol.com

The Jewel

There once was a King who wished to send a precious jewel to the Queen of a nearby country, whom he wanted to wed.

He had an impenetrable strongbox and a pickproof lock, but he did not dare to send the key with the courier, who could steal the jewel. He also could not send the key with a separate courier, because they could meet up and steal the jewel.

Instead, the King sent the courier without the key. When the strongbox reached the Queen, she added her own secure lock, and sent the box back.

When the strongbox reached the King, he removed his lock, and sent the box back with only the Queen's lock.

Now the Queen could remove her lock and receive the jewel.

King	=	Sender
Queen	=	Receiver
Jewel	=	Message
Lock	=	Encryption
Key	=	Decryption
Courier	=	Network

Private Key Cryptography

- Message M, typically a string of bytes.
- The sender applies his encryption S to get SM. This message is sent to the receiver.
- The receiver applies her encryption R to SM to get RSM. **R and S must commute.** The twice-encrypted message is sent back to the sender.
- The sender applies his inverse function S' to RSM to get S'RSM = S'SRM = RM. This goes back to the receiver.
- The receiver applies her inverse function R' to RM to get R'RM = M, the original message.

Pros and Cons

- **Pro:** Each party has a private key which never needs to distributed or shared.
- **Pro:** There are no key servers.
- **Pro:** Each key is used only once. (Can use a different key for each block of the message.)
- **Pro:** Parties can never run out of keys.
- **Con:** Two encryption steps and two decryption steps.
- **Con:** The message needs to be transmitted 3 times.

Assumptions

- Sender and receiver are communicating over an insecure network.
- There may be *passive* opponents who are intercepting all of the transmissions.
- Opponents can read messages, but cannot delete, insert or alter messages.
- This talk will not cover man-in-the-middle attacks.

How???

- How can you find R and S that commute?
- First idea: Simple substitution and transposition ciphers commute. Flaw: Whether R is a transposition or a substitution, if an opponent has intercepted both SM and RSM, then R is obvious.
- Second idea: Sender and receiver both use a one-time pad. Encryption is exclusive-or with a random string. Flaw: If an opponent intercepts all 3 messages, then exclusive-oring them gives the message. (SM)(RSM)(RM) = M.

Adi Shamir

- Sender and receiver agree on a large prime p, which may be publicly known.
- Sender and receiver choose encryption exponents s and r, and decryption exponents s' and r' such that $s's \equiv r'r \equiv 1 (\mathrm{mod}\ p-1)$.
- By Fermat's Little Theorem, for any M relatively prime to p we have $M^{r'r} \equiv M^{s's} \equiv M \ (\mathrm{mod}\ p)$.
- The three messages will then be $M^s (\mathrm{mod}\ p)$, $M^{rs} (\mathrm{mod}\ p)$ and $M^r (\mathrm{mod}\ p)$.
- Finding r or s from these messages involves solving the Discrete Logarithm Problem, which is known to be very difficult.

Massey-Omura

- Same concept, except exponentiation is over a Galois Field GF(2^m).
- The modulus operation (mod 2^m) is now just taking the last m bits of each product.
- May be faster (highly disputed).

Matrix Methods

- Treat the characters of the message as elements of a ring ***R***, typically a ring of size 256.
- Treat strings of characters as vectors or matrices over ***R***.
- $M = (M_1, M_2, M_3, \ldots, M_b)$ where b is the block size.
- $M = (M_{11}, M_{12}, M_{13}, \ldots, M_{21}, M_{22}, \ldots, M_{mn})$ where mn is the block size. (The matrix does not have to be square.)

Review: **Rings**

- A ring is an abstraction of the concept of a number.
- Examples of rings: integers, rational numbers, real numbers, integers modulo n, Gaussian integers, quaternions.
- Just like numbers, elements of a ring can be added and multiplied. The ring is closed under addition and multiplication.

Ring Addition

- Abelian group under addition.
- Ring addition is commutative a+b=b+a.
- Associative (a+b)+c=a+(b+c).
- There is an additive identity called 0 such that a+0=0+a=a.
- Every ring element a has an additive inverse -a such that a+(-a)=0.

Ring Multiplication

- May or may not be commutative.
- Associative (ab)c=a(bc).
- Distributive over addition, a(b+c)=ab+ac and (a+b)c=ac+bc.
- There is a multiplicative identity called 1 such that a1=1a=a.
- *Some* ring elements may have either a left multiplicative inverse a' such that a'a=1, or a right multiplicative inverse a” such that aa”=1.

If a ring element has both a left and a right multiplicative inverse they are the same a'=a”.

a'(aa”) = (a'a)a” Associative

a'1 = 1a”

a' = a”

Review: **Matrices**

- Matrices are rectangular arrays of elements, called scalars, taken from some ring ***R***.
- Matrices can be added element-by-element. If $A+B=C$ then $A_{ij}+B_{ij}=C_{ij}$ for all i,j.
- A p×q matrix A can be multiplied by a q×r matrix B to get a p×r matrix C by forming the inner products of the rows of A with the columns of B.

$$C_{ij} = A_{i1}B_{1j} + A_{i2}B_{2j} + A_{i3}B_{3j} + \ldots + A_{iq}B_{qj}$$

- Using this form of addition and multiplication, square matrices themselves form a ring.

Matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}_{n \times m} = \left(a_{ij} \right)_{n \times m}$$

Over Commutative Rings

- Square matrices over commutative rings have some special properties.
- The determinant is a sum of signed products of elements. $\det(A) = A_{11}A_{22} - A_{12}A_{21}$.
- The matrix can be inverted only if the value of its determinant is an invertible element of ***R***.
- Multiplying any row or any column by k multiplies the determinant by k.
- Swapping two rows or columns leaves the determinant unchanged, except for possible sign change.
- Adding or subtracting a multiple of one row or column to/from another does not change the value.

- The number of linearly independent rows is the rank of the matrix.
- Rank is invariant under transposing rows, transposing columns, multiplying a row or column by any invertible ring element, adding a multiple of one row or column to another.

Over Non-Commutative Rings

- Matrices over non-commutative rings do not have the same properties.
- Left-multiplying the first column by k left-multiplies the determinant by k.
- Right-multiplying the last column by k right-multiplies the determinant by k.
- For all other rows and columns multiplying by k multiplies the determinant by k only if k is commutative, otherwise it gives unpredictable results, even changing from 0 to non-zero, or vice-versa.

- Swapping two columns gives unpredictable results, even 0 to non-zero.
- Adding one row or column to another row or column gives unpredictable results.
- Rows of a matrix can be left- or right-linearly dependent.
- The (left or right) rank of a matrix can vary according to the order in which linearly-dependent rows are eliminated.
- **Cannot use Gaussian reduction to find the inverse matrix.**

Strategy

- Matrices over non-commutative rings present numerous difficulties.
- Let's turn those lemons into lemonade.
- We will use that to our advantage, and our opponents' disadvantage.
- Three methods: left-side, right-side and 2-sided.

Left Side

- Treat b characters of the message M as a column vector

$$M = (M_1, M_2, M_3, \ldots, M_b).$$

- Encrypt and decrypt by multiplying M on the left by matrices over the non-commutative ring ***R***.
- Sender sends SM.
- Receiver returns RSM.
- Sender sends S'RSM = RM. Receiver retrieves R'RM = M.

Right Side

- Treat b characters of the message M as a row vector

  $$M = (M_1, M_2, M_3, \ldots, M_b).$$

  and multiply on the right by matrices over the non-commutative ring ***R***.
- Sender sends MS.
- Receiver returns MSR.
- Sender sends MSRS' = MR. Receiver retrieves MRR' = M.

Both Sides

- Treat b^2 characters of the message as a matrix over ***R***. The block size is b^2.
- Multiply on both sides by matrices.
- Sender sends SMT.
- Receiver returns RSMTQ.
- Sender sends S'RSMTQT' = RMQ.
- Receiver retrieves R'RMQQ' = M.

WARNING!

It does not work if the sender multiplies a message vector on one side and the receiver multiplies on the opposite side.

Two **BIG** Problems

- Problem 1: Matrix multiplication is not commutative.
- Problem 2: How can the sender and receiver compute the inverse matrices S' and R' ?
- Solution 1: Choose S and R from commutative family $\mathcal{F}$ of matrices.
- But... how do you generate a commutative family of matrices over a non-commutative ring?

Clarification

- To be clear, $\mathcal{F}$ is a commutative family of matrices, not a family of commutative matrices.
- Matrices in the family need not be commutative. If A is in the commutative family, and B is not in the family, then generally AB≠BA.
- Members of $\mathcal{F}$ always commute with one another, but rarely with outsiders.
- Diagonal matrices can be commutative, but the family of diagonal matrices produces a very weak encryption.

Quick Answer

- Choose any invertible square matrix A of high multiplicative order, say $m>10^{25}$.
- Multiplicative order of A is the smallest positive integer m such that $A^m=I$, the identity matrix.
- Then I, A, A^2, A^3, …, A^{m-1} is a commutative family of matrices.
- If X is any invertible square matrix, then

  I, XAX', XA^2X', XA^3X' …, $XA^{m-1}X'$ is another commutative family of matrices.
- If m is small, an opponent can try them all.

Choosing a Ring

- In order to have small matrices of very high multiplicative order, and to make it difficult for an opponent to invert the matrices we need the ring ***R*** to have:
- Lots of non-commutative elements.
- Lots of elements of high multiplicative order (the least k such that $a^k=1$).
- Trade-off: having a very high multiplicative order means fewer elements of maximal order.

Invertible Matrices

- The security of the matrix methods rests on the fact that it is difficult for an opponent to invert the encryption matrices.
- So... how can the sender generate S and S' ?
- **Answer**: we construct A and X to be invertible, and calculate A' and X' from the outset. Then when we choose $S=XA^kX'$ from the family, we set $S'=X'(A')^kX$.
- Even though we cannot invert a general matrix, we can construct a matrix that will be easy to invert, namely an upper triangular matrix where all the diagonal elements are both commutative and invertible.

- Let U and V be such upper triangular matrices with inverses U' and V' where U has high multiplicative order, m.
- Choose integers j,k>1 with j mutually prime to the multiplicative order of U. Set $A=U^j$ and $X=V^k$.
- The desired commutative family $\mathcal{F}$ of matrices is I, XAX', XA^2X', XA^3X', …, $XA^{m-1}X'$.
- Sender and receiver can generate encryption matrices from $\mathcal{F}$ by choosing random exponents in the range 1 to m-2, inclusive.

Upper Triangular Matrix

$$\begin{bmatrix} 2 & 0 & 4 & 6 & 8 \\ 0 & 1 & 7 & 9 & 1 \\ 0 & 0 & 3 & 2 & 5 \\ 0 & 0 & 0 & 4 & 8 \\ 0 & 0 & 0 & 0 & 7 \end{bmatrix}$$

Easy to invert

Summary

- To send a message M, the sender multiplies M by a matrix S taken from the commutative family $\mathcal{F}$ (2 matrices for the 2-sided version) of matrices over the non-commutative ring ***R***.
- The receiver multiplies SM by her encryption matrix R to get RSM.
- The sender multiplies by the inverse matrix S', to remove his encryption, S'RSM=RM.
- The receiver removes her encryption by multiplying by the inverse matrix R', R'RM=M.
- The receiver can now read the message.

Optional Topics

- Making it fast.
- Analyzing the security.

Make it Fast

- Raising A to a random power takes about $3\log_2(m)/2$ matrix multiplications, where m is the size of the family $\mathcal{F}$.
- To reduce this from $3\log_2(m)/2$ to 1, precompute a set of generator matrices $G_1, G_2, \ldots, G_n$. Each G_i is XA^kX' for some random k. At least one of the exponents must be mutually prime to m.
- Sender generates an encryption matrix $G=G_iG_j$ where i and j are chosen randomly, $i \neq j$.
- G then replaces G_i.

- If the encryption software is supplied by a vendor, the vendor can provide the generators $G_1, G_2, \ldots, G_n$, but keep A and X secret.
- When customers install the software they can repeat the step $G_i := G_i G_j$ many times, so that neither the vendor, nor any party knows another party's generators.

Security

- Consider left-sided encryption with block size b.
- Suppose an opponent has intercepted the three messages SM, RSM and RM.
- From SM and RSM the opponent gets b linear equations for the b^2 unknown elements of R.
- The opponent also knows that R is a member of the commutative family $\mathcal{F}$. If A is in $\mathcal{F}$ then (XAX')R=R(XAX'). For simplicity, AR=RA.
- If A has maximal multiplicative order, then this gives b(b-1) linearly independent equations of the form $\Sigma_j A_{ij} R_{jk} = \Sigma_j R_{ij} A_{jk}$ (*bilinear* equations).

- If ***R*** is non-commutative then bilinear equations are tough. Even the general solution to **ax+xb=c** is not known.
- If the ring were commutative, you could simply flip $R_{ij}A_{jk}$ to get $A_{jk}R_{ij}$ yielding b(b-1) additional linear equations (best case).
- That would give a total of b^2 linear equations in b^2 unknowns. In principle, these can be solved to get R.
- So *perhaps* the matrix method is not secure using commutative rings.
- Let's look deeper.

Weak Linear, Strong Linear

- Consider the ring of integers modulo 12, and look at some linear equations.

  $5x = 1$ has one solution, namely 5.

  $2x = 6$ has 2 solutions, 3 and 9.

  $9x = 3$ has 3 solutions, 3, 7 and 11.

  $8x = 4$ has 4 solutions, 2, 5, 8 and 11.

  $6x = 6$ has 6 solutions, 1, 3, 5, 7, 9 and 11.

- So $5x=1$ is much stronger than $6x=6$ because it gives much more information about x.
- A set of linear equations over a ring can give many solutions.

- Since the ring ***R*** is non-commutative the terms $R_{ij} A_{jk}$ cannot be flipped.
- Instead there is a trick to convert the bilinear equations to linear equations over a larger set of unknowns.
- Replace each element of ***R*** with an expression like au+bv+cw where the coefficients a,b,c are commutative elements of ***R.***
- The fixed elements u,v,w are called *generators* of ***R***. There may be many sets of generators, which may vary in size. Generator u may be commutative, and may be chosen to be 1.
- The representation au+bv+cw of any given ring element is not unique.

- If there are g generators, then the opponent will have gb^2 linear equations for the gb^2 unknown elements of R.

  (**Looks like we're cooked!!** Fear not.)
- The equations will not be linearly independent. They can have gazillions of solutions.
- With commutative rings, any one of these solutions will work. That is, if the opponent finds a pseudo-inverse R" of R, the message can be retrieved by R"RM = M.
- With non-commutative rings only the correct solution will work. The opponent will need to search through, on average, half of all the solutions.