Quasi-inverse Based Cryptography

Thuc Dinh Nguyen and Van H. Dang

University of Science, Ho Chi Minh City, Vietnam
{ndthuc,dhvan}@fit.hcmus.edu.vn

Abstract. We are interested in monoids and its applications. If every element x in a monoid has a quasi-inverse y in the sense of von Neumann, that satisfies $x \cdot y \cdot x = x$ and $y \cdot x \cdot y = y$, the monoid is regular. Our purpose is to use regular monoids to build two abstract algebraic public key cryptosystems: key exchange protocol and public key encryption with keyword search scheme. In addition to illustrating how the two cryptosystems work, we provide instances of these abstract algebraic models and analyse them in terms of cryptanalysis security.

Keywords: public key cryptography, key establishment, searchable encrypted data, quasi-inverse, pseudo-inverse matrix.

1 Introduction

We consider two problems in public key cryptography: establishing a shared secret between two or more parties and searching on encrypted data using a public key system.

There are several protocols to establish a shared secret. Diffie-Hellman key agreement is a fundamental technique providing unauthenticated key agreement [1]. The security of Diffie-Hellman protocol rests on the intractability of the Diffie-Hellman problem and the related problem of computing discrete logarithms. RSA cryptosystem [2], the most widely used public key cryptosystem, can be used for this goal as well. The security of RSA system is based on the intractability of the integer factorization problem. In this paper, we will propose a new abstract key exchange protocol based on the quasi-inverse in the sense of von Neumann [3]. Besides, we will give an instance of this abstract model. The security of the proposed instance model is based on the matrix factorization problem, which is believed to be a hard problem.

Public-key Encryption with Keyword Search (PEKS) was introduced for the first time by D. Boneh et.al [5]. It is a mechanism which enables searching with encrypted keywords. They showed that PEKS could be used for exchanging sensitive emails via an untrusted mail server. In [4], B.R. Waters et.al pointed out that PEKS can also be used to build an encrypted and searchable audit log. In this paper, we will present an abstract construction of PEKS using the quasi-inverse concept, and provide an instance of this abstract PEKS scheme using pseudo-inverse matrices, or generalized inverse matrices [6]. The security of the proposed PEKS scheme instance is based on the matrix factorization problem, which will be investigated in Section 3.

B. Murgante et al. (Eds.): ICCSA 2013, Part IV, LNCS 7974, pp. 629–642, 2013.

Organization of the Paper. In Section 2 we review the concept of quasi-inverse and give an exposition of two abstract algebraic public key cryptography models: (i) key exchange protocol, and (ii) public key encryption with keyword search. In Section 3 we provide two instances of the abstract algebraic models based on the concept of pseudo-inverse matrix, that is also briefly summarized. We investigate the cryptanalysis security of these instance models as well. Finally, the conclusion are drawn in Section 4.

2 Public Key Cryptography

2.1 Quasi-inverse

Thanks to Classification of Finite Simple Groups theorem in [10], we have a better understanding of permutation groups. In a transformation monoid, the elements that are permutations form a group. B. Steinberg studied transformation monoids and properties of the monoids in [3].

Definition 1. *Let $\mathcal{M}$ be a monoid.*

(i) The element a of a monoid $\mathcal{M}$ is regular, or quasi-inverse in the sense of von Neumann, in $\mathcal{M}$ if there is an element b of $\mathcal{M}$ such that $a \cdot b \cdot a = a$ and $b \cdot a \cdot b = b$. Then b is called the quasi-inverse of a.
(ii) A monoid $\mathcal{M}$ is regular if all its elements are regular.

We will use the quasi-inverse property to develop two public key cryptography models.

2.2 Key Exchange Protocol

Key establishment is any process whereby a shared secret key becomes available to two or more parties for subsequent cryptographic usage.

Many various protocols have been proposed for the key establishment. In this subsection, we propose a new key exchange protocol for key establishment using quasi-inverse elements of a given regular monoid [3].

(1) Alice and Bob agree to use a same regular monoid $\mathcal{M}$.
(2) Alice chooses a secret quasi-inverse element f in $\mathcal{M}$.
(3) Bob chooses a secret quasi-inverse element g in $\mathcal{M}$.
(4) Alice computes $X = f \cdot h$, then sends X to Bob, where h is the quasi-inverse of $f : f \cdot h \cdot f = f$ and $h \cdot f \cdot h = h$.
(5) Bob computes $Z = y \cdot g$ and $k_b = g \cdot X$, then sends Z and k_b to Alice, where y is the quasi-inverse of $g : g \cdot y \cdot g = g$ and $y \cdot g \cdot y = y$.
(6) Alice computes $k_a = Z \cdot f$, then sends k_a to Bob.
(7) Alice computes $K = k_b \cdot f$.
(8) Bob computes $K = g \cdot k_a$.
(9) Alice and Bob now share a secret key K. This is because
$K = k_b \cdot f = g \cdot X \cdot f = g \cdot f \cdot h \cdot f = g \cdot f = g \cdot y \cdot g \cdot f = g \cdot Z \cdot f = g \cdot k_a$.

2.3 Public Key Encryption with Keyword Search

In 2004, D. Boneh et al. [5] introduced a scheme of Public key Encryption with Keyword Search (PEKS) and provided constructions based on a variant of the Decision Diffie-Hellman assumption. In this subsection, we propose a new construction based on the quasi-inverse element of a given regular monoid $\mathcal{M}$ [3]. We shall need cryptographic hash functions [7]: $H_1 : \{0,1\}^* \longrightarrow \mathcal{M}$, and $H_2 : \mathcal{M} \longrightarrow \{0,1\}^n$, and and $H_3 : \{0,1\}^* \times \mathcal{M} \longrightarrow \mathcal{M}$. Our PEKS works as follows:

– KeyGen(): Do the following:
  1. Choose a random quasi-inverse element h in $\mathcal{M}$,
  2. Compute the quasi-inverse f of $h : f \cdot h \cdot f = f$ and $h \cdot f \cdot h = h$,
  3. Set the public key $K_{pub} = f \cdot h$, and the private key $K_{priv} = f$,
  4. Return K_{pub}, K_{priv}.
– PEKS(m, K_{pub}): Given a message m in $\{0,1\}^*$ and the public key K_{pub}, do the following:
  1. Hash m into an element z in $\mathcal{M}$ by using H_1: $z = H_1(m)$,
  2. Compute the value $P = z \cdot K_{pub}$,
  3. Return P.
– Trapdoor(m, K_{priv}): Given a message m in $\{0,1\}^*$ and the private key K_{priv}, do the following:
  1. Let $q = H_3(m, K_{priv}) \in \mathcal{M}$,
  2. Compute a pair of values $T_1 = H_2(H_1(m) \cdot K_{priv} \cdot q)$ and $T_2 = K_{priv} \cdot q$,
  3. Return $T = (T_1, T_2)$.
– Test(P, T): Given a value of PEKS, P, and a trapdoor, $T = (T_1, T_2)$, returns true if $H_2(P \cdot T_2) = T_1$, and false otherwise.

We prove the consistency of the proposed PEKS scheme as follows.

Let P be PEKS of the message m, and $T = (T_1, T_2)$ be the trapdoor of message m'. According to Trapdoor function for the message m', we have

$$T_1 = H_2(H_1(m') \cdot K_{priv} \cdot q) = H_2(H_1(m') \cdot f \cdot q)$$

Besides, we apply Test function for the message m to compute

$$H_2(P \cdot T_2) = H_2(H_1(m) \cdot K_{pub} \cdot K_{priv} \cdot q) = H_2(H_1(m) \cdot f \cdot h \cdot f \cdot q) = H_2(H_1(m) \cdot f \cdot q)$$

If $m = m'$ then $H_2(P \cdot T_2) = T_1$, hence Test returns ”true”. Otherwise, Test returns ”false”.

Note that we can define cryptographic hash functions using available hash functions, such as SHA [8], MD5 [9].

3 Instances of Our Proposed Models

3.1 Pseudo-inverse Matrix

For the convenience of the reader we repeat the relevant concept about full rank factorization of matrices, the proof of which is included in Appendix A, and the concept of pseudo-inverse and quasi-inverse in the sense of von Neumann.

Proposition 2. *Let A denote a matrix in $\mathbb{R}^{m \times n}$.*
If $rank(A) = r$, there exist matrices B in $\mathbb{R}^{m \times r}$ and C in $\mathbb{R}^{r \times n}$ such that $A = B \cdot C$, where $rank(B) = rank(C) = r$.

We say $A = B \cdot C$ is the full rank factorization of A.

Definition 3. *Let A denote a matrix in $\mathbb{R}^{m \times n}$ having $rank(A) = r$ and the full rank factorization $A = B \cdot C$, where $B_{m \times r}$ is the matrix of basic columns from A and $C_{r \times n}$ is the matrix of non-zero rows from E_A (E_A is the unique reduced echelon form derived from A by means of row operations). The matrix defined by*

$$A^{\dagger} = C^T \cdot \left(B^T \cdot A \cdot C^T\right)^{-1} \cdot B^T$$

is called pseudo-inverse of A.

Theorem 4. *Given a matrix A in $\mathbb{R}^{m \times n}$ such that $rank(A) = r$, let $A = B \cdot C$ be the full rank factorization of A, where $B_{m \times r}$ is the matrix of basic columns from A and $C_{r \times n}$ is the matrix of non-zero rows from E_A (E_A is the unique reduced echelon form derived from A by means of row operations). The matrix defined by*

$$A^{\dagger} = C^T \cdot \left(B^T \cdot A \cdot C^T\right)^{-1} \cdot B^T$$

is quasi-inverse in the sense of von Neumann, that satisfies

$$\begin{aligned} A \cdot A^{\dagger} \cdot A &= A, \\ A^{\dagger} \cdot A \cdot A^{\dagger} &= A^{\dagger} \end{aligned}$$

Proof. Notice that:

$$\begin{aligned} A^{\dagger} &= C^T \cdot \left(B^T \cdot A \cdot C^T\right)^{-1} \cdot B^T \\ &= C^T \cdot \left(B^T \cdot B \cdot C \cdot C^T\right)^{-1} \cdot B^T \\ &= C^T \cdot \left(C \cdot C^T\right)^{-1} \cdot \left(B^T \cdot B\right)^{-1} \cdot B^T, \end{aligned}$$

hence

$$\begin{aligned} A \cdot A^{\dagger} \cdot A &= B \cdot C \cdot C^T \cdot \left(C \cdot C^T\right)^{-1} \cdot \left(B^T \cdot B\right)^{-1} \cdot B^T \cdot B \cdot C \\ &= B \cdot C \\ &= A. \end{aligned}$$

and

$$\begin{aligned} A^\dagger \cdot A \cdot A^\dagger &= (C^T \cdot \big(C \cdot C^T\big)^{-1} \cdot \big(B^T \cdot B\big)^{-1} \cdot B^T) \cdot (B \cdot C) \\ &\cdot (C^T \cdot \big(C \cdot C^T\big)^{-1} \cdot \big(B^T \cdot B\big)^{-1} \cdot B^T) \\ &= C^T \cdot \big(C \cdot C^T\big)^{-1} \cdot \big(B^T \cdot B\big)^{-1} \cdot B^T \cdot B \cdot C \cdot C^T \cdot \big(C \cdot C^T\big)^{-1} \\ &\cdot \big(B^T \cdot B\big)^{-1} \cdot B^T \\ &= C^T \cdot \big(C \cdot C^T\big)^{-1} \cdot \big(B^T \cdot B\big)^{-1} \cdot B^T \\ &= A^\dagger \end{aligned}$$

Proposition 5. *Given a matrix A in $\mathbb{R}^{m\times n}$.*

(i) If $rank(A) = n$, then $A^\dagger = \big(A^T \cdot A\big)^{-1} \cdot A^T$;
(ii) If $rank(A) = m$, then $A^\dagger = A^T \cdot \big(A \cdot A^T\big)^{-1}$

Proof. We will prove case (i). The case (ii) can proved in the same manner.

If $rank(A_{m\times n}) = n$ then matrix $(A^T \cdot A)_{n\times n}$ is non-singular, hence there exists $\big(A^T \cdot A\big)^{-1}$.

Let $A^\dagger = \big(A^T \cdot A\big)^{-1} \cdot A^T$. We justify $A^\dagger$ by the properties of quasi-inverse in Theorem 4 as follows:

$$A \cdot A^\dagger \cdot A = A \cdot (A^T \cdot A)^{-1} \cdot A^T \cdot A = A,$$
$$A^\dagger \cdot A \cdot A^\dagger = (A^T \cdot A)^{-1} \cdot A^T \cdot A \cdot (A^T \cdot A)^{-1} \cdot A^T = A^\dagger.$$

□

3.2 Pseudo-inverse Matrix Over the Field $\mathbb{Z}_p$

Let A denote a matrix in $\mathbb{Z}_p^{m\times n}$, where p is a prime. We can now prove the following result.

Theorem 6. ***(Uniqueness)*** *If A is a pseudo-invertible matrix, then $A^\dagger$ is unique by arguing that $A^\dagger$ is the unique solution of the four equations,*

$$A \cdot A^\dagger \cdot A = A; \tag{1}$$
$$A^\dagger \cdot A \cdot A^\dagger = A^\dagger; \tag{2}$$
$$\big(A \cdot A^\dagger\big)^T = A \cdot A^\dagger; \tag{3}$$
$$\big(A^\dagger \cdot A\big)^T = A^\dagger \cdot A. \tag{4}$$

Proof. Suppose that B and C are two pseudo-inverse matrices of A that satisfy the above properties, we have
$A \cdot B = (A \cdot B)^T$, and $A \cdot C = (A \cdot C)^T$,
$B \cdot A = (B \cdot A)^T$, and $C \cdot A = (C \cdot A)^T$,
$A \cdot B \cdot A = A$, and $A \cdot C \cdot A = A$,

$B \cdot A \cdot B = B$, and $C \cdot A \cdot C = C$.
We know that
$A \cdot B = (A \cdot B)^T = B^T \cdot A^T$,
and
$A \cdot C \cdot A = A \Leftrightarrow A^T \cdot C^T \cdot A^T = A^T$,
hence
$A \cdot B = (A \cdot B)^T = B^T \cdot A^T = B^T \cdot A^T \cdot C^T \cdot A^T$
In addition, we have
$A \cdot B = (A \cdot B)^T = B^T \cdot A^T$,
$A \cdot C = (A \cdot C)^T = C^T \cdot A^T$.
We deduce that
$A \cdot B = (A \cdot B)^T = B^T \cdot A^T = B^T \cdot A^T \cdot C^T \cdot A^T = A \cdot B \cdot A \cdot C = A \cdot C$
In the same manner, we can deduce
$B \cdot A = C \cdot A$
Therefore
$B = B \cdot A \cdot B = B \cdot A \cdot C = C \cdot A \cdot C = C$
This means that if the pseudo-inverse exists, it is unique. □

Next we want to apply Proposition 5 to generate a pseudo-invertible matrix A in $\mathbb{Z}_p^{m \times n}$ and its pseudo-inverse $A^\dagger$. The problem is the product $\left(A^T \cdot A\right)$ is possibly not invertible even if $rank(A) = n$. It follows that $\left(A^T \cdot A\right)^{-1}$ does not exist. Similarly, $\left(A \cdot A^T\right)^{-1}$ may not exist even if $rank(A) = m$. In such cases, we cannot find the pseudo-invesre $A^\dagger$ as in Proposition 5. We illustrate such cases by example as below.

Example 7. $A = \begin{pmatrix}1\\2\end{pmatrix}$ is pseudo-invertible over $\mathbb{Z}_7$, but it is not pseudo-invertible over $\mathbb{Z}_5$. Indeed,

- Over $\mathbb{Z}_7$, the matrix $A^T \cdot A = (1\ 2) \cdot \begin{pmatrix}1\\2\end{pmatrix} = (5)$ is invertible and its inverse is $(5)^{-1} \equiv (3) \pmod 7$. According to Proposition 5, the pseudo-inverse B is computed by $B = \left(A^T \cdot A\right)^{-1} \cdot A^T = \left((1\ 2) \cdot \begin{pmatrix}1\\2\end{pmatrix}\right)^{-1} \cdot (1\ 2) = (3\ 6)$.

  Now we justify B by the properties in Theorem 6 as follows:

$$A \cdot B \cdot A = \begin{pmatrix}1\\2\end{pmatrix} \cdot (3\ 6) \cdot \begin{pmatrix}1\\2\end{pmatrix} = \begin{pmatrix}1\\2\end{pmatrix} = A$$

$$B \cdot A \cdot B = (3\ 6) \cdot \begin{pmatrix}1\\2\end{pmatrix} \cdot (3\ 6) = (3\ 6) = B$$

$$(A \cdot B)^T = \left(\begin{pmatrix}1\\2\end{pmatrix} \cdot (3\ 6)\right)^T = \begin{pmatrix}3\ 6\\6\ 5\end{pmatrix} = \begin{pmatrix}1\\2\end{pmatrix} \cdot (3\ 6) = A \cdot B$$

$$(B \cdot A)^T = \left((3\ 6) \cdot \begin{pmatrix}1\\2\end{pmatrix}\right)^T = (1) = (3\ 6) \cdot \begin{pmatrix}1\\2\end{pmatrix} = B \cdot A$$

  It follows that B is the unique pseudo-inverse $A^\dagger$ of A.

- Over $\mathbb{Z}_5$, the matrix $A^T \cdot A = (1\ 2) \cdot \begin{pmatrix}1\\2\end{pmatrix} = (5) \equiv (0) \pmod 7$ is not invertible, therefore we cannot find the pseudo-inverse matrix of A.

The above example gives an idea to build an algorithm to generate a pseudo-invertible one-dimensional matrix and its pseudo-inverse as below.

Algorithm 1. SimplePseudoMatrix(n, p)- Generating a pseudo-inverse one-column matrix and its pseudo-inverse matrix

```
Input: $n, p$ {$gcd(n, p) = 1$ and $p$ is a prime}
Output: A matrix $A$ in $\mathbb{Z}_p^{n\times 1}$, and its pseudo-inverse $A^\dagger$ in $\mathbb{Z}_p^{1\times n}$
for $i = 1 \to (n-1)$ do
    $A[i,1] = random(p)$ {$random(p)$ is a function that returns a random value $v \in \{0, ..., p-1\}$}
end for
Choose $q \in \{0, 1, ..., p-1\}$ such that $(\sum_{i=1}^{n-1} A[i,1]^2 + q^2) \bmod p \neq 0$
$A[n,1] = q$
$A^\dagger = (A^T \cdot A)^{-1} \cdot A^T$
return $(A, A^\dagger)$
```

The correctness of SimplePseudoMatrix algorithm is due to the fourth statement. Indeed, because $(\sum_{i=1}^{n} A[i,1]^2) \bmod p \neq 0$, $A \cdot A^T = [(\sum_{i=1}^{n} A[i,1]^2) \bmod p]$ is an invertible matrix in $\mathbb{Z}_p^{1\times 1}$. Then the matrix

$$B = (A^T \cdot A)^{-1} \cdot A^T = A^\dagger.$$

We see that, by the uniqueness of the pseudo-inverse matrix as in Theorem 6, if the matrix $B_{1\times n}$ is the pseudo-inverse of the matrix $A_{n\times 1}$, then the transpose of B, $C_{n\times 1} = B^T$, is the pseudo-inverse of the transpose of A, $D_{1\times n} = A^T$. We illustrate this idea by the following example.

Example 8. In the Example 7, the matrix $A = \begin{pmatrix}1\\2\end{pmatrix}$ is pseudo-invertible over $\mathbb{Z}_7$, and its pseudo-inverse matrix is $B = (3\ 6)$. Then the transpose of B, $C = \begin{pmatrix}3\\6\end{pmatrix}$ will be the pseudo-inverse matrix of the transpose of A, $D = (1\ 2)$ over $\mathbb{Z}_7$. We verify this by the properties in Theorem 6

$$C \cdot D \cdot C = \begin{pmatrix}3\\6\end{pmatrix} \cdot (1\ 2) \cdot \begin{pmatrix}3\\6\end{pmatrix} = \begin{pmatrix}3&6\\6&5\end{pmatrix} \cdot \begin{pmatrix}3\\6\end{pmatrix} = \begin{pmatrix}3\\6\end{pmatrix} = C,$$

$$D \cdot C \cdot D = (1\ 2) \cdot \begin{pmatrix}3\\6\end{pmatrix} \cdot (1\ 2) = (1) \cdot \begin{pmatrix}1\\2\end{pmatrix} = (1\ 2) = D,$$

$$(C \cdot D)^T = \left(\begin{pmatrix}3\\6\end{pmatrix} \cdot (1\ 2)\right)^T = \begin{pmatrix}3&6\\6&5\end{pmatrix}^T = \begin{pmatrix}3&6\\6&5\end{pmatrix} = C \cdot D,$$

$$(D \cdot C)^T = \left((1\ 2) \cdot \begin{pmatrix}3\\6\end{pmatrix}\right)^T = (1)^T = (1) = D \cdot C$$

In order to generate a pseudo-invertible multi-dimensional matrix instead of one-dimensional, and its pseudo-inverse matrix, we can use the following algorithm.

Algorithm 2. PseudoInverseMatrix(m, n, p)- Generating a pseudo-invertible matrix and its pseudo-inverse matrix

```
Input: $m, n, p : m < n$
Output: A pseudo-invertible matrix $A$ in $\mathbb{Z}_p^{m\times n}$, and its pseudo-inverse matrix $A^\dagger$ in $\mathbb{Z}_p^{n\times m}$.
1: Generate a random invertible matrix $A' \in \mathbb{Z}_p^{m\times m}$.
2: Generate a random matrix $A" \in \mathbb{Z}_p^{m\times(n-m)}$.
3: Let $A = [A'|A"]$ (combining $A'$ and $A''$ column by column)
4: if $det(A.A^T)$ mod $p = 0$ then
5:     Goto (2).
6: end if
7: Compute $A^\dagger = A^T \cdot \left(A \cdot A^T\right)^{-1}$.
8: return $(A, A^\dagger)$
```

The correctness of PseudoInverseMatrix algorithm is due to the second, third and fourth statements. Indeed, because $det(A \cdot A^T)$ mod $p \neq 0$, there exists $(A \cdot A^T)^{-1}$, hence the pseudo-inverse $A^\dagger$ is determined by $A^\dagger = A^T \cdot \left(A \cdot A^T\right)^{-1}$.

In Algorithm 2, it is necessary to generate a random invertible matrix over $\mathbb{Z}_p$. This can be done efficiently by applying Theorem 9.

Theorem 9.

(i) Upper-triangular matrix U in $\mathbb{Z}_p^{n\times n}$ is invertible iff the product of the diagonal entries of U is not equal to zero, $\prod_{i=1}^{n} u_{ii}$ mod $p \neq 0$.

(ii) Lower-triangular matrix L in $\mathbb{Z}_p^{n\times n}$ is invertible iff the product of the diagonal entries of L is not equal to zero, $\prod_{i=1}^{n} l_{ii}$ mod $p \neq 0$.

(iii) Given a matrix A as a product of two matrices U and V in $\mathbb{Z}_p^{n\times n}$, $A = U \cdot V$. The matrix A is invertible iff both U and V are invertible.

Proof.

(i) Since U is upper-triangular, its determinant is computed by the product of the diagonal entries. We have $det(U) = \prod_{i=1}^{n} u_{ii}$ mod $p \neq 0 \Leftrightarrow U$ is invertible.

(ii) In the same manner, we have that $det(L) = \prod_{i=1}^{n} l_{ii}$ mod $p \neq 0 \Leftrightarrow L$ is invertible.

(iii) We have that $det(A) = det(U \cdot V) = det(U)det(V) \neq 0$ [mod p] iff U and V are invertible.

3.3 Key Exchange Protocol

Now we establish a key exchange protocol based on quasi-inverse. In order to exchange a common key, Alice and Bob do the following steps.

(1) Alice and Bob agree on a same prime p.
(2) Alice generates a secret pseudo-invertible matrix F in $\mathbb{Z}_p^{m\times n}$ and its pseudo-inverse H in $\mathbb{Z}_p^{n\times m}$, $H = F^\dagger$.
(3) Bob generates a secret pseudo-invertible matrix G in $\mathbb{Z}^{n\times m}$ and its pseudo-inverse Y in $\mathbb{Z}_p^{m\times n}$, $Y = G^\dagger$.
(4) Alice computes X as the product of F and H, $X = F \cdot H$, and sends X to Bob.
(5) Bob computes Z as the product of Y and G, $Z = Y \cdot G$, and a middle key $K_b = G \cdot X$, then sends Z and K_b to Alice.
(6) Alice computes a middle key $K_a = Z \cdot F$ and sends K_a to Bob.
(7) Alice computes $K = K_b \cdot F$.
(8) Bob computes $K = G \cdot K_a$.
(9) Alice and Bob now share a secret key K.

Indeed,

$$K = K_b \cdot F = G \cdot X \cdot F = G \cdot F \cdot H \cdot F = G \cdot F = G \cdot Y \cdot G \cdot F = G \cdot Z \cdot F = G \cdot K_a.$$

Example 10.

(1) Alice and Bob agree on the prime $p = 7$.
(2) Alice generates a secret pseudo-invertible matrix $F = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and its pseudo-inverse $H = F^\dagger = (3\ 6)$.
(3) Bob generates a secret pseudo-invertible matrix $G = (3\ 4)$ and its pseudo-inverse $Y = G^\dagger = \begin{pmatrix} 6 \\ 1 \end{pmatrix}$.
(4) Alice computes $X = F \cdot H = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot (3\ 6) = \begin{pmatrix} 3 & 6 \\ 6 & 5 \end{pmatrix}$
(5) Bob computes $Z = Y \cdot G = \begin{pmatrix} 6 \\ 1 \end{pmatrix} \cdot (3\ 4) = \begin{pmatrix} 4 & 3 \\ 3 & 4 \end{pmatrix}$ and a middle key $K_b = G \cdot X = (3\ 4) \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot (3\ 6) = (5\ 3)$, then sends them to Alice.
(6) Alice computes a middle key $K_a = Z \cdot F = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$ and sends to Bob.
(7) Alice computes $K = K_b \cdot F = (4)$.
(8) Bob computes $K = G \cdot K_a = (4)$.
(9) Alice and Bob now share a secret key $K = (4)$.

3.4 Public Key Encryption with Keyword Search Scheme

In this scheme, the prime p is published.

- $\mathsf{KeyGen}()$: Do the following:
  1. Generate a random pseudo-invertible matrix G in $\mathbb{Z}_p^{m\times n}$ and its pseudo-inverse F, $F = G^{\dagger}$,
  2. Compute the public key $K_{pub} = G \cdot F$, and the private key $K_{priv} = G$,
  3. Returns (K_{pub}, K_{priv}).
- $\mathsf{PEKS}(w, K_{pub})$: Given a keyword w in $\{0, 1\}^*$ and the public key K_{pub}, do the following:
  1. Hash the keyword w into a matrix M in $Z_p^{1\times m}$ by using some hash function H_1, $M = H_1(w)$ (H_1 is a hash function that receives a string, and outputs a matrix in $Z_p^{1\times m}$),
  2. Compute $P = M \cdot K_{pub}$ using the public key K_{pub},
  3. Return P.
- $\mathsf{Trapdoor}(w, K_{priv})$: Given a keyword w in $\{0, 1\}^*$ and the private key K_{priv}, do the following:
  1. Let $Q = H_3(w, K_{priv}) \in \mathbb{Z}_p^{n\times n}$ by using some function H_3 that is a hash function that receives a string, a matrix, and outputs an invertible matrix in $Z_p^{n\times n}$,
  2. Computes a pair of values $T_1 = H_2(H_1(w) \cdot K_{priv} \cdot Q)$ and $T_2 = K_{priv} \cdot Q$ using the private key K_{priv} (H_2 is a hash function that receives a vector in $\mathbb{Z}^{1\times m}$, and outputs a binary value),
  3. Return $T = (T_1, T_2)$.
- $\mathsf{Test}(P, T)$: Given a value of PEKS, P, and a trapdoor, $T = (T_1, T_2)$, returns true if $H_2(P \cdot T_2) = T_1$, and false otherwise.

Example 11.

- $\mathsf{KeyGen}()$:
  1. Generates a random pseudo-inverse matrix $G = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ in $\mathbb{Z}_7^{2\times 1}$, and its pseudo-inverse $F = G^{\dagger} = \begin{pmatrix} 3\ 6 \end{pmatrix}$,
  2. Compute the public key $K_{pub} = G.F = \begin{pmatrix} 3 & 6 \\ 6 & 5 \end{pmatrix}$ and the private key $K_{priv} = G = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.
- $\mathsf{PEKS}(w, K_{pub})$: Suppose that $M = H_1(w) = \begin{pmatrix} 2\ 5 \end{pmatrix}$. Then compute $P = M \cdot K_{pub} = \begin{pmatrix} 1\ 2 \end{pmatrix}$.
- $\mathsf{Trapdoor}(w, K_{priv})$: Suppose that $Q = (3)$. Then with $M = H_1(w) = \begin{pmatrix} 2\ 5 \end{pmatrix}$, compute $T = (T_1, T_2)$ as follows:

$$T_1 = H_2\left((2\ 5) \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot (3) \right) = H_2(1),$$

$$T_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot (3) = \begin{pmatrix} 3 \\ 6 \end{pmatrix}.$$

– $\mathsf{Test}(P, T)$:

- If $M = (2\ 5)$, then $P = (1\ 2)$. We have

$$H_2\left(P \cdot T_2\right) = H_2\left(\left(1\ 2\right) \cdot \begin{pmatrix}3\\6\end{pmatrix}\right) = H_2(1) = T_1$$

Therefore Test returns ”true”.

- If $M' = (1\ 5) \neq (2\ 5) = M$, then $P' = (5\ 3)$. We have

$$H_2(P' \cdot T_2) = H_2\left((5\ 3) \cdot \begin{pmatrix}3\\6\end{pmatrix}\right) = H_2(5) \neq H_2(1) = T_1$$

Therefore Test returns ”false”.

3.5 Cryptanalysis

In the Key-Exchange Protocol. In order to grasp the common key $K = G{\cdot}F$, a hacker has to solve one of the following two types of problems:

(i) Find F from public key $X = F \cdot H$ (or G from $Z = Y \cdot G$),
(ii) Find G from middle key $K_b = G \cdot X$ (or F from $K_a = Z \cdot F$).

This means that if he can find F or G, he hacks successfully the key-exchange protocol.

Without loss of generality, if one of two following problems, denoted as HP1 and HP2 respectively, is solved, the key-exchange protocol is broken.

HP1 (recovering the private key from the public key):
Given a matrix A in $\mathbb{Z}_p^{m\times m}$ as the product of a pseudo-invertible matrix X in $\mathbb{Z}_p^{m\times n}$ and its pseudo-inverse Y in $\mathbb{Z}_p^{n\times m}$, $A = X \cdot Y$, find X or Y.

HP2 (recovering the private key from the middle-key):
Given a non-invertible matrix B in $\mathbb{Z}_p^{n\times n}$ and a matrix A in $\mathbb{Z}_p^{n\times m}$ as the product of B and a pseudo-invertible matrix X in $\mathbb{Z}_p^{n\times m}$, $A = B \cdot X$, find X.

In the Public Key Encryption with Keyword Search Scheme. We need only consider two cases.

Case 1. If the private key $K_{priv} = F$ can be recovered from the public key $A_{pub} = H \cdot F$, where $F = H^\dagger$, this scheme is broken. This is equivalent to solving the HP1 above.

Case 2. If the the private key $K_{priv} = F$ can be recovered from the trapdoor information $C = Q \cdot K_{priv} = Q \cdot F$, where Q is a secret invertible matrix. Obviously, this problem is as difficult as HP1.

Discussion. It is not difficult to see that HP1 is harder than HP2, because to solve HP1, we are only given one product-matrix and we must find its factor-matrices. Meanwhile, in the HP2, we are given two matrices and one unknown matrix. Consequently, we can assume that the security of the proposed public key algorithms relies on the difficulty of HP2 instead of both HP1 and HP2.

In the HP2, we are given $A = B \cdot X$ in $\mathbb{Z}_p^{n\times m}$, and B in $\mathbb{Z}_p^{n\times n}$, which is not invertible. We need to find X in $\mathbb{Z}_p^{n\times m}$. Suppose that $rank(B_{n\times n}) = r \leq n$, we have $r \times m$ equations with $n \times m$ unknowns. Therefore there are $(n-r) \times m$ free unknowns. In order to find out the exact solution X, a hacker must try to test $p^{m(n-r)}$ times if X is pseudo-invertible and satisfies $A = B \cdot X$. Hence we say that the problem HP2 can be solved with the computational complexity of $\mathcal{O}(p^{m.(n-r)})$. It is infeasible if the prime p or/and $m(n-r)$ are large. For example, if p is a prime of 128 bit length, the hacker has to try $2^{128m(n-r)}$ tests, and it is infeasible even if $m(n-r) = 1$.

4 Conclusion

We studied the regular monoids and quasi-inverse concept in the sense of von Neumann, and used them to create two new public key cryptosystems. Such a quasi-inverse element y has good properties, that are $(y \cdot x)^n = y \cdot x$ and $(x \cdot y)^m = x \cdot y$, for all positive integers m, n.

Besides, we provide the instance of the two models through using pseudo-inverse matrices. The security of the instances relies on the hardness of the matrix factorization problem.

In the future lines of research, we aim to investigate in greater detail about the cryptanalysis security of the proposed models, and/or find out other under-monoid.

A Proof for the Full Factorization

Let $B_{m\times r} = [A_{*b_1} A_{*b_2} \cdots A_{*b_r}]$ contain the basic columns of A, and let $C_{r\times r}$ contain the non-zero rows of E_A, where E_A denotes the unique reduced row echelon form derived from A by means of row operations. If A_{*k} is basic, means $A_{*k} = A_{*b_j}$, then $C_{*k} = e_j$, and

$$(B \cdot C)_{*k} = B \cdot C_{*k} = B \cdot e_j = B_{*j} = A_{*b_j} = A_{*k}.$$

If A_{*k} is non-basic, then C_{*k} is non-basic and has the form

$$\begin{aligned} C_{*k} &= \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_j \\ \vdots \\ 0 \end{pmatrix} \\ &= \mu_1 \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \mu_2 \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \cdots + \mu_j \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \\ &= \mu_1 e_1 + \mu_2 e_2 + \ldots + \mu_j e_j, \end{aligned}$$

where e_i's are the basic columns to the left of C_{*k}.

Because $A\ B$, the relationships that exist among the columns of A are exactly the same as the relationships that exist among the columns of E_A.

In particular,

$$A_{*k} = \mu_1 A_{*b_1} + \mu_2 A_{*b_2} + \ldots + \mu_j A_{*b_j},$$

where A_{*b_i}'s are the basic columns to the left of A_{*k}. Therefore,

$$\begin{aligned} (B \cdot C)_{*k} &= B \cdot C_{*k} \\ &= B \cdot (\mu_1 e_1 + \mu_2 e_2 + \ldots + \mu_j e_j) \\ &= \mu_1 B_{*1} + \mu_2 B_{*2} + \ldots + \mu_j B_{*j} \\ &= \mu_1 A_{*b_1} + \mu_2 A_{*b_2} + \ldots + \mu_j A_{*b_j} \\ &= A_{*k}. \end{aligned}$$

□

References

1. Diffie, W.: The First Ten of Public Key Cryptography. Proceedings of the IEEE 76(5), 560–577 (1988)
2. Rivest, R.L., Shamir, A., Adleman, L.M.: Cryptographic Communications System and Method. U.S. Patent #4,405,829 (1983)
3. Steinberg, B.: A Theory of Transformation Monoids: Combinatorics and Representation Theory. The Electronic Journal of Combinatorics 17 #R164 (2010)

4. Waters, B., Balfanz, D., Durfee, G., Smetters, D.: Building an Encrypted and Searchable Audit Log. In: The 11th Annual Network and Distributed System Security Symposium, NDSS 2004, San Diego, California (2004)
5. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public Key Encryption with Keyword Search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)
6. Campell, S.L., Meyer, C.D.: Generalized Inverse of Linear Transformations. Dover Publications, New York (1979)
7. Menezes, A.J., Oorschot, P.C.V., Vanstone, S.A.: Handbook of Applied Cryptography, p. 33. CRC Press (1997)
8. ANSI X9.30 (PART 2), American National Standard for Financial Services - Public key cryptography using irreversible algorithms for the financial services industry - Part 2: The secure hash algorithm (SHA), ASC X9 Secretariat - American Bankers Association (1993)
9. Rivest, R.: The MD5 Message-Digest Algorithm. RFC 1321 (1992)
10. Gorenstein, D.: The Classification of Finite Simple Groups Vol. 1. Groups of Noncharacteristic 2 Type. The University Series in Mathematics. Plenum Press (1983) ISBN 978-0-306-41305-6, MR 746470