Key Establishment Algorithms for Some Deterministic Key Predistribution Schemes

Sushmita Ruj and Bimal Roy

Applied Statistics Unit, Indian Statistical Institute
203 B T Road, Kolkata 700 108, India

Abstract. Key establishment is a major problem in sensor networks because of resource constraints. Several key predistribution schemes have been discussed in literature. Though the key predistribution algorithms have been described very well in these papers, no key establishment algorithm has been presented in some of them. Without efficient key establishment algorithm the key predistribution schemes are incomplete. We present efficient shared-key discovery algorithms for some known deterministic key predistribution schemes. Our algorithms run in $O(1)$ and $O(\sqrt[3]{N})$ time and the communication overhead is at most $O(\log \sqrt{N})$ bits, where N is the size of the network. The efficient key establishment schemes make deterministic key predistribution an attractive option over randomized schemes.

1 Introduction

Distributed Sensor Networks (DSN) consist of sensor nodes which are resource constrained. Sensor networks have wide application in military as well as civilian purposes. To ensure secure communication, any two sensor nodes should communicate in an encrypted manner using a common secret key.

The keys are either predistributed in the sensor nodes or online key exchange protocols can be used. Though public key cryptosystems using RSA and ECC have been used in low end devices [1], they are not efficient where several hundred thousand nodes with very limited resources are required. In key predistribution keys are placed in sensor nodes prior to deployment. Any two nodes can communicate with each other if they have some common key. Communication is carried out by encrypting messages using the common key. Key establishment is carried out in the following way. First two nodes find out if they have any common key and the identifier or the value of this common key. This step is called *shared key discovery*. If two nodes do not share a common key then a path key needs to be found. Path key establishment is discussed in Section 4.

The efficiency of key establishment algorithms depends on two factors.

1. The communication overhead - the amount of information that needs to be broadcasted to enable other nodes to find the common keys.
2. Efficient shared key discovery algorithms - algorithms which are efficient in terms of computation and storage.

Ruj S. and Roy B. (2008).
Key Establishment Algorithms for Some Deterministic Key Predistribution Schemes.
In *Proceedings of the 6th International Workshop on Security in Information Systems*, pages 68-77
DOI: 10.5220/0001729900680077

Key predistribution techniques can be randomized, deterministic or hybrid. In randomized technique of key predistribution by Eschenauer and Gligor [2] and Chan Perrig and Song [3], keys are drawn randomly from a key pool and placed in each sensor node. Suppose each sensor contains k keys. In some schemes such as [2, 3], sensor nodes broadcast the entire list of key identifiers. On receiving a list of identifiers a sensor nodes compares it with its own identifier list to find a common identifier or a key is computed from the common identifiers. All encryption and decryption is done using this common key. For a key chains consisting of k keys $O(k \log v)$, bits needs to be sent, where $v =$ number of keys in the key pool. The identifiers may be sorted which requires $O(k \log k)$ time. Then to find a common key identifier it takes $O(k)$ time. This fact was discussed by Lee and Stinson in [4, Section 2.1.2]. Another way is to use Merkle puzzles as done by Eschenauer and Gligor in [2] and Chan, Perrig and Song in [3]. Then to find a common shared key between two nodes, each node has to broadcast a list $\{\alpha, E_{k_i}(\alpha), i = 1, 2, \cdots, k\}$, where α is a challenge. The decryption of E_{k_i} with proper key by the other node would reveal the challenge α and establish a shared key with the broadcasting node. The communication overhead for the schemes [2, 3] will be $O(k \log v)$, where v is the number of keys in the key pool. The calculation of $E_{k_i}(\alpha), i = 1, 2, \cdots, k$ encryption will require $O(k)$ time. However this is not a very efficient way, since communication and computation complexity increases.

Deterministic key predistribution has the advantage that keys are placed in sensor nodes in a predetermined manner. This helps us to device efficient algorithms for establishing the common keys between sensor nodes. Deterministic key predistribution using combinatorial designs have been studied in [5–9]. Hybrid designs combine the above two approaches and have been studied in [5, 10]. Though key predistribution algorithms have been discussed in details [5, 8, 9], key establishment algorithms have not been given in any of them. Without efficient key establishment algorithms the predistribution schemes are incomplete. In ISPA '07 Ruj and Roy proposed key predistribution scheme using PBIBD (Partially balanced incomplete block designs) and in Inscrypt '07 Dong, Pei and Wang proposed a key predistribution scheme using $3 - designs$. In both these two schemes it was assumed that communication was carried using common shared keys. However no algorithm for key establishment was given in both these papers. In this paper we present efficient shared-key discovery algorithms for the key predistribution schemes given by Roy and Ruj [8] and Dong, Pei, Wang [9]. The algorithms run in $O(1)$ and $O(\sqrt[3]{N})$ respectively. This makes them highly suitable for sensor networks. The communication overhead is also very less, at most $O(\log \sqrt{N})$ bits, where N is the size of the network. Hence our schemes are better than those given in [2, 3]. The design of these algorithms will motivate us towards designing deterministic predistribution schemes.

The rest of the paper is organized as follows. In Section 2 and 3 we present key establishment strategies for the key predistribution schemes given by Dong, Pei and Wang in [9] and Ruj and Roy in [8]. Path key establishment has been represented in 4. We conclude with some open problems in Section 5.

2 Shared-key Discovery for Key Predistribution Scheme given by Dong, Pei and Wang [9]

The key predistribution scheme proposed by Dong et al. in [9] makes use of $3-designs$. In particular they use inversive planes to assign keys in the sensor nodes. We present an algorithm to find a common key between two given nodes or report failure if no common key is present. For completeness we present the key predistribution algorithm using 3-designs. A detailed discussion on $3-$designs can be found in [11, Section 9.2.1].

Let q be a prime. We use an irreducible polynomial $f(x)$ of order 2 to construct a field $F_{q^2} = Z_q/(f(x))$. Let $f(x) = x^2 + f_1'x + f_0'$.

Let the field elements be $f_0 = 0$, $f_1 = 1$, $f_2, \cdots, f_{q^2-1}$. We choose $a, b, c, d \in F_{q^2}$, such that $ad - bc \neq 0$. Let $\infty \notin F_q$. We define a function

$$\pi\begin{pmatrix} a\ b \\ c\ d \end{pmatrix}(x) = \begin{cases} \frac{ax+b}{cx+d} & \text{if } x \in F_q \text{ and } cx + d \neq 0 \\ \infty & \text{if } x \in F_q,\ cx + d = 0 \text{ and } ax + b \neq 0 \\ \frac{a}{c} & \text{if } x = \infty \text{ and } c \neq 0 \\ \infty & \text{if } x = \infty,\ c = 0 \text{ and } a \neq 0 \end{cases}$$

Let $PGL(2, q^2)$ to consist of all distinct permutations $\pi\begin{pmatrix} a\ b \\ c\ d \end{pmatrix}$, where $a, b, c, d \in F_{q^2}$, such that $ad - bc \neq 0$. It can be proved as in [11, Lemma 9.25] that there are $q^6 - q^2$ such permutations. We create blocks in the following way. For each permutation π_i, $(i = 0, 1, 2, \cdots, q^6 - q^2 - 1)$ block B_{π_i} consists elements $\pi_i(j)$, $j = 0, 1, \cdots, q - 1, \infty$. So each block consists of $q + 1$ elements. The resulting design gives rise to a $3 - (q^2 + q + 1, q + 1, 1)$ design.

We consider the distinct blocks and map the blocks to the nodes and preload each node with the keys contained in that particular block. Since the number of distinct blocks is $q^3 + q$, the number of nodes supported by the network is $q^3 + q$. Let the key chain belonging to node i be denoted by $\{k_t^{(i)} : 0 \leq t \leq q\}$. Any two nodes can share at most two keys. Next we describe an algorithm to find the common keys between any two nodes if one exists, or report failure if one doesn't exists.

2.1 Algorithm to find Common Key

Let node i want to communicate with node j. The node j broadcasts corresponding values of a, b, c, d. Denote these values by a_j, b_j, c_j, d_j. We give below the algorithm to find the common key that i shares with j. When j wants to calculate the common key that it shares with i, it runs the same algorithm and finds x_j and calculates the common key as $ck = \frac{a_j x_j + b_j}{c_j x_j + d_j}$. (See step 25). All calculations are done modulo q.

Algorithm 1. Shared key discovery for the scheme of Dong et al. [9].

Require a_i, b_i, c_i, d_i, a_j, b_j, c_j, d_j
if $c_i = 0$ and $c_j = 0$
  $ck = \infty$
else if $a_i/c_i = a_j/c_j \neq 0$
  $ck = a_i/c_i$
else
  $tag = 0$
  for $s = 0$ to q
    if $k_s^{(i)} = \infty$
      $tag = 1$
    endif
  endfor
  if $tag = 1$
    for $s = 0$ to q
      if $c_j s + d_j = 0$
        $ck = \infty$
      else
        Print : No solution exists
      endif
    endfor
  else
    Solve the equation $\frac{a_i x_i + b_i}{c_i x_i + d_i} = \frac{a_j x_j + b_j}{c_j x_j + d_j}$ for x_i.
    if No solution exists then
      Print : No solution exists
    else
      $ck = \frac{a_i x_i + b_i}{c_i x_i + d_i}$
    endif
  endif
endif

Correctness of Algorithm 1. Suppose $c_i = 0$ and $c_j = 0$, then $\pi_i(\infty) = \pi_j(\infty) = \infty$, hence $ck = \infty$ and Step 1-2 holds. If $\pi_i(\infty) = a_i/c_i$, if $a_i/c_i \neq 0$. So if $a_i/c_i = a_j/c_j \neq 0$, then $\pi_i(\infty) = \pi_j(\infty) = a_i/c_i$ and Step 3-4 holds. If one of the keys in node i is ∞, and $c_j x_j + d_j = 0$, for some $x_j \in F_q \bigcup \{\infty\}$, then ∞ is a common key between the nodes i and j. The only condition that remains if that when $x_i, x_j \neq \infty$ and $c_i x_i + d_i \neq \infty$ and $c_j x_j + d_j \neq \infty$. In such a case we try to find if there exists x_i and x_j, such that $\frac{a_i x_i + b_i}{c_i x_i + d_i} = \frac{a_j x_j + b_j}{c_j x_j + d_j}$. Hence if a solution to this equation exists, then the common key will be $\frac{a_i x_i + b_i}{c_i x_i + d_i}$. By the design we know that any two nodes will share maximum of two keys. We now show that we can find all the values of x_i if they exist, or report failure if no keys are common.

We know that a, b, c d are all one degree polynomial with coefficients in F_q. Let
$a_i = a_{i1}x + a_{i0}$, $b_i = b_{i1}x + b_{i0}$, $c_i = c_{i1}x + c_{i0}$, $d_i = d_{i1}x + d_{i0}$,
$a_j = a_{j1}x + a_{j0}$, $b_j = b_{j1}x + b_{j0}$, $c_j = c_{j1}x + c_{j0}$, $d_j = d_{j1}x + d_{j0}$,

We solve for x_i in the following equation. Note that all calculations are done modulo q.

$\frac{a_i x_i + b_i}{c_i x_i + d_i} = \frac{a_j x_j + b_j}{c_j x_j + d_j}$
$\Rightarrow (a_i x_i + b_i)(c_j x_j + d_j) = (c_i x_i + d_i)(a_j x_j + b_j)$
$\Rightarrow \{(a_{i1}x + a_{i0})x_i + (b_{i1}x + b_{i0})\}\{(c_{j1}x + c_{j0})x_j + (d_{j1}x + d_{j0})\}$
$= \{(c_{i1}x + c_{i0})x_i + (d_{i1}x + d_{i0})\}\{(a_{j1}x + a_{j0})x_j + (b_{j1}x + b_{j0})\}$
$\Rightarrow \{(a_{i1}x_i + b_{i1})x + (a_{i0}x_i + b_{i0})\}\{(c_{j1}x_j + d_{j1})x + (c_{j0}x_j + d_{j0})\}$
$= \{(c_{i1}x_i + d_{i1})x + (c_{i0}x_i + d_{i0})\}\{(a_{j1}x_j + b_{j1})x + (a_{j0}x_j + b_{j0})\}$
$\Rightarrow x\{(a_{i1}x_i + b_{i1})(c_{j0}x_j + d_{j0}) + (a_{i0}x_i + b_{i0})(c_{j1}x_j + d_{j1}) +$
$(q - f_1')(a_{i1}x_i + b_{i1})(c_{j1}x_j + d_{j1})\} +$
$(a_{i0}x_i + b_{i0})(c_{j0}x_j + d_{j0}) + (q - f_0')(a_{i1}x_i + b_{i1})(c_{j1}x_j + d_{j1})\}$
$= x\{(a_{j1}x_j + b_{j1})(c_{i0}x_i + d_{i0}) + (a_{j0}x_j + b_{j0})(c_{i1}x_i + d_{i1}) +$
$(q - f_1')(a_{j1}x_j + b_{j1})(c_{i1}x_i + d_{i1})\} +$
$(a_{j0}x_j + b_{j0})(c_{i0}x_i + d_{i0}) + (q - f_0')(a_{j1}x_j + b_{j1})(c_{i1}x_i + d_{i1})\}$

Equating the coefficients of x and the constant term we get two equations

$$P_1 x_i x_j + Q_1 x_i + R_1 x_j + S_1 = 0 \tag{1a}$$

and

$$P_2 x_i x_j + Q_2 x_i + R_2 x_j + S_2 = 0 \tag{1b}$$

where,
$P_1 = a_{i1}c_{j0} + a_{i0}c_{j1} + (q - f_1')a_{i1}c_{j1} - (a_{j1}c_{i0} + a_{j0}c_{i1} + (q - f_1')a_{j1}c_{i1})$,
$P_2 = a_{i0}c_{j0} + (q - f_0')a_{i1}c_{j1} - (a_{j0}c_{i0} + (q - f_0')a_{j1}c_{i1})$,
$Q_1 = a_{i1}d_{j0} + a_{i0}d_{j1} + (q - f_1)a_{i1}d_{j1} - (b_{j1}c_{i0} + b_{j0}c_{i1} + (q - f_1')b_{j1}c_{i1})$,
$Q_2 = a_{i0}d_{j0} + (q - f_0')a_{i1}d_{j1} - (b_{j0}c_{i0} + (q - f_0')b_{j1}c_{i1})$,
$R_1 = b_{i1}c_{j0} + b_{i0}c_{j1} + (q - f_1')b_{i1}c_{j1} - (a_{j1}d_{i0} + a_{j0}d_{i1} + (q - f_1')a_{j1}d_{i1})$,
$R_2 = b_{i0}c_{j0} + (q - f_0')b_{i1}c_{j1} - (a_{j0}d_{i0} + (q - f_0')a_{j1}d_{i1})$,
$S_1 = b_{i1}d_{j0} + b_{i0}d_{j1} + (q - f_1')b_{i1}d_{j1} - (b_{j1}d_{i0} + b_{j0}d_{i1} + (q - f_1')b_{j1}d_{i1})$,
$S_2 = b_{i0}d_{j0} + (q - f_0')b_{i1}d_{j1} - (b_{j0}d_{i0} + (q - f_0')b_{j1}d_{i1})$.

Eliminating the term $x_i x_j$, we get

$$(\frac{Q_1}{P_1} - \frac{Q_2}{P_2})x_i + (\frac{R_1}{P_1} - \frac{R_2}{P_2})x_j = \frac{S_2}{P_2} - \frac{S_1}{P_1}$$

$$x_j = U + V x_i \tag{2}$$

where, $U = (\frac{S_2}{P_2} - \frac{S_1}{P_1})(\frac{R_1}{P_1} - \frac{R_2}{P_2})^{-1}$ and $V = q - (\frac{Q_1}{P_1} - \frac{Q_2}{P_2})(\frac{R_1}{P_1} - \frac{R_2}{P_2})^{-1}$

Substituting the value of x_j in (1a) we get

$$P_1(Vx_i + U)x_i + Q_1(Vx_i + U) + R_1(Vx_i + U) + S_1 = 0$$
$$\Rightarrow P_1 V x_i^2 + x_i(UP_1 + VQ_1 + VR_1) + UP_1 + UQ_1 + UR_1 + S_1 = 0$$

where

The above equation can have either one or two or no solutions which can be calculated easily. Hence, we obtain a maximum of two values for x_i. Then the common key

will be $\frac{a_i x_i + b_i}{c_i x_i + d_i}$. Thus the algorithm gives all the common keys or reports failure if none is present.

Time Complexity of Algorithm 1. Steps 7 to 10 are executed at most q times. All the other steps require constant time. Since the number of nodes N is $O(\sqrt[3]{N})$, the time complexity is $O(\sqrt[3]{N})$. Only the four values of a, b, c and d need to be broadcasted. Hence the communication overhead is $O(\log q)$ bits, which is quite efficient compared to algorithms proposed in [2, 3].

3 Shared-key Discovery for Ruj and Roy Schemes of Key Predistribution [8]

Two key predistribution schemes have been discussed in [8]. Both the schemes make use of Partially balanced incomplete block designs for key predistribution.

Ruj and Roy Scheme [8] I. The authors use a triangular association scheme to predistribute the keys in the sensor network. The design can be found in [8, Section 3]. We now give an algorithm to find at least one common key between two given nodes.

3.1 Algorithm to find Common Key

Let nodes P and Q want to communicate with each other. For this purpose we store the location of the node in the array A. The nodes broadcast their position in the array A. We need to calculate a simple function which will give the identity of one or more common key between any two nodes. Given the position (x, y) of a node P the value in the matrix at position (x, y) is given by

$$f(x,y) = \begin{cases} *, \text{ for } x = y \\ y - x, \text{ for } x = 1, x < y \\ x - y, \text{ for } y = 1, x > y \\ n + y - x - 1, \text{ for } x = 2, x < y \\ n + x - y - 1, \text{ for } y = 2, x > y \\ (x-1)n - (x+1)(x-2)/2 + (y - x - 1), \text{ for } x < y, x > 2 \\ (y-1)n - (y+1)(y-2)/2 + (x - y - 1). \text{ for } x > y, y > 2 \end{cases}$$

Given any node P it can find the ids of the keys in common with another node Q at position (x', y') in the following way.

1. If $x = x'$, then $a_{f(x,t)}$ and $a_{f(y,y')}$ are the common keys between P and Q for $t = 1, 2, \cdots, n$ and $t \neq x, y, y'$.
2. If $y = y'$, then $a_{f(t,y)}$ and $a_{f(x,x')}$ are the common keys between P and Q for $t = 1, 2, \cdots, n$ and $t \neq x, y, x'$.
3. If $x \neq x'$ and $y \neq y'$, then the keys $a_{f(x,x')}$, $a_{f(x,y')}$, $a_{f(y,x')}$ and $a_{f(y,y')}$ are common between P and Q.

Since there are more than one keys in common, the nodes can choose any of the common keys. Since $f(x, y)$ can be calculated in constant time, key agreement can be done in $O(1)$ time. Also the memory overhead is $O(\log n) = O(\log \sqrt{N})$ bits, since only the position of the node in the array is sent.

Ruj and Roy Scheme [8] II. The second scheme given in [8] is an extension of Scheme I. Here a second array A' is used in conjunction to the array A given above. Array A' is given in [8, Section 5.1]. The first $n(n-1)/2$ are loaded as given in the Scheme I. For the next $n(n-1)/2$ nodes, keys are chosen according to the pattern in array A'. For the $n(n-1)/2+i$th node, the ids of the keys are the elements in the row and the column in which i belongs. The element in the position (x, y) in the matrix A' is given by $f'(x,y) = \begin{cases} *, \text{ for } x = y \\ (x-y-1)(2n-x+y)/2+y, \text{ for } x > y \\ (y-x-1)(2n-y+x)/2+x \text{ otherwise} \end{cases}$

3.2 Algorithm to find Common Key

Let the nodes i and j want to communicate with each other. Any node j broadcasts the following information.

1. Array s from which j was derived. $m[s] = 0$ if j is derived from A and $m[s] = 1$ if j is derived from A'. This requires one bit.
2. Position (x_j, y_j) of j in the array from which it has been derived. This requires $O(log\ \sqrt{N})$.

Given the above information node i can calculate the common keys using Algorithm 2.

3.3 Proof of Correctness and Time Complexity of Algorithm 2

If both the nodes i and j are derived from the same array, then we follow the algorithm similar to that given in Section 4.1. We will consider the case when i and j are derived from arrays A' and A respectively. The case where i and j are derived from arrays A and A' respectively will follow similarly.

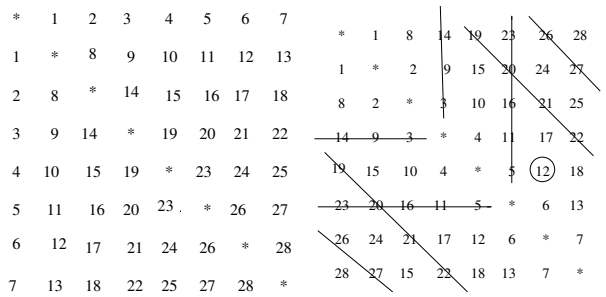
We consider the following example

Example. Suppose position of $i = (5, 7)$ in array A' and that of $j = (4, 6)$ in array A. Ids of keys belonging to j are $3, 9, 14, 19, 21, 22$ and $5, 11, 16, 23, 26, 27$. These have been marked in array A' as below. We mark four diagonal lines and two vertical lines and two horizontal lines. We find all the crossed elements that lie in the 7th column. These are the elements common between i and j which occur along the two diagonals. These are 26 and 21. Similarly, all the crossed elements that lie in the 5th row are the common elements between i and j. These are 19 and 5 in the above example. So the common keys have identifiers 5, 19, 21 and 26.

Algorithm 2. Shared key Discovery for Scheme II.

1 if $m[i] = m[j] = 0$
2 if $x_i = x_j$
3 Ids of the common keys are $a_{f(x_i,t)}$ and $a_{f(y_i,y_j)}$, for $t = 1, 2, \cdots, n$ and $t \neq x_i, y_i, y_j$
4 else if $y_i = y_j$
5 Ids of the common keys are $a_{f(t,y_i)}$ and $a_{f(x_i,x_j)}$, for $t = 1, 2, \cdots, n$ and $t \neq x_i, y_i, x_j$
6 else
7 Ids of the common keys $a_{f(x_i,x_j)}$, $a_{f(x_i,y_j)}$, $a_{f(y_i,x_j)}$ and $a_{f(y_i,y_j)}$.
8 endif
9 else if $m[i] = m[j] = 1$
10 if $x_i = x_j$
11 Ids of the common keys are $a_{f'(x_i,t)}$ and $a_{f'(y_i,y_j)}$, for $t = 1, 2, \cdots, n$ and $t \neq x_i, y_i, y_j$.
12 else if $y_i = y_j$
13 Ids of the common keys are $a_{f'(t,y_i)}$ and $a_{f'(x_i,x_j)}$, for $t = 1, 2, \cdots, n$ and $t \neq x_i, y_i, x_j$.
14 else
15 Ids of the common keys $a_{f'(x_i,x_j)}$, $a_{f'(x_i,y_j)}$, $a_{f'(y_i,x_j)}$ and $a_{f'(y_i,y_j)}$.
16 end if
17 else if $m[i] = 1$ and $m[j] = 0$
18 Ids of the common keys as calculated by i will be $a_{f'(a,b)}$ where
1. $(a, b) = (y_i - x_j, y_i), (y_i - y_j, y_i), (y_i + x_j, y_i), (y_i + y_j, y_i), (x_i, x_i - x_j), (x_i, x_i - y_j), (x_i, x_i + x_j), (x_i, x_i + y_j)$, such that $0 < a, b \leq n$ and $a \neq b$.
2. $(a, b) = (x_i, x_j), (x_i, y_j)$ if $a < b$, $x_i \neq x_j$
3. $(a, b) = (x_j, y_i), (y_i, y_i)$ if $a > b$, $y_i \neq y_i$
4. $(x_i, 1), (x_i, 2), \cdots, (x_i, x_{i-1})$ if $x_j = x_i$.
5. $(1, y_i), (2, y_i), \cdots, (y_{i-1}, y_i)$ if $y_j = y_i$.
19 else
20 Ids of the common keys will be calculated as above except that the $f_{(a,b)}$ will be calculated instead of $f'_{(a,b)}$.
21 endif

Proceeding as in the example above there will be at most four diagonal lines and two vertical lines and two horizontal lines. The position of the elements along the marked diagonals that lie on the same column as i will be given by, $(a, b) = (y_i - x_j, y_i), (y_i - y_j, y_i), (y_i + x_j, y_i), (y_i + y_j, y_i)$ such that $0 < a, b \leq n$ and $a \neq b$.
Similarly, all the positions of the elements along the marked diagonals that lie on the same row as i and is given by $(a, b) = (x_i, x_i - x_j), (x_i, x_i - y_j), (x_i, x_i + x_j), (x_i, x_i + y_j)$, such that $0 < a, b \leq n$ and $a \neq b$.
If both i and j belong to the same row (ie, $x_i = x_j$), then the position of the common elements will be $(a, b) = (x_i, 1), (x_i, 2), \cdots, (x_i, x_{i-1})$. These elements lie on one of the marked rows. If both i and j belong to the same column (ie, $y_i = y_j$), then the position of the common elements will be $(a, b) = (1, y_i), (2, y_i), \cdots, (y_{i-1}, y_i)$. These elements lie on one of the marked columns. If i and j do not belong to the same row



(a) array 1 (b) array 2

Fig. 1. Array 1 and array 2.

or column, then the positions will be given by $(a, b) = (x_i, x_j), (x_i, y_j)$ if $a < b$ and $(a, b) = (x_j, y_i), (y_i, y_i)$ if $a > b$. So the ids of the common keys are given by $f'_{(a,b)}$. To communicate, the nodes can choose any of the common keys. All the steps take $O(1)$ to be done. Hence the overall time complexity is $O(1)$.

Each node broadcasts the array to which it belongs (this requires just one bit) and its position in the array from which it is derived. Since the order of each array is $O(\sqrt{N})$ (where N is the number of nodes), $O(log\ \sqrt{N})$ bits have to be broadcasted.

4 Path Key Establishment

Where shared key exists between nodes, a secure channel is created and all communications between the nodes are performed using the common key. However there may exist situations where nodes may not share common keys (as in the scheme of [9] which uses $t - designs$) or when common shared keys are exposed because of node compromise. In such cases a path needs to be established between the nodes. Suppose u and v having no common key need to communicate with each other. u establishes communication with some node n_1 through some common key which further establishes communication with n_2 and so onwards. Let $u, n_1, n_2, \cdots, n_l, w$ be the path between u and v. Let u share a common key k_1 with n_1. Similarly, let n_1 share a common key k_2 with n_2, and n_{l-1} share a common key k_l with n_l and n_l share a common key k_{l+1} with w. u generates a random key K, encrypts with k_1 and sends it to n_1. n_1 decrypts K using k_1 and encrypts it using k_2 and sends it to n_2 and the process continues. Ultimately K reaches v using k_{l+1}. So v can decrypt using k_{l+1} and obtain K. K is the path key and communication between u and v is done using K. This approach has been taken in [12]. The path is found in a breadth first manner.

5 Conclusions

Various deterministic key predistribution have been studied in literature. However efficient key establishment has not been discussed for many key predistribution schemes. We present key shared-key discovery algorithms for the key predistribution schemes

given by Ruj and Roy in [8] and by Dong, Pei and Wang in [9] , which had not been presented in these papers. The algorithms run in $O(1)$ and $O(sqrt[3]N)$ respectively. Also communication requires at most $O(log\ \sqrt{N})$ bits, where N is the size of the network. Randomized key predistribution algorithms lack efficient key management strategies because there is no underlying pattern. The efficient key establishment strategies of deterministic schemes as given in this paper motivates us to use deterministic schemes for key predistribution. We are working towards devising algorithms for shared key discovery for other known key predistribution schemes. One interesting problem will be to design efficient key establishment schemes for randomized key predistribution schemes.

References

1. Gura, N., Patel, A., Wander, A., Eberle, H., Shantz, S.C.: Comparing elliptic curve cryptography and rsa on 8-bit cpus. In Joye, M., Quisquater, J.J., eds.: CHES. Volume 3156 of Lecture Notes in Computer Science., Springer (2004) 119–132
2. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In Atluri, V., ed.: ACM Conference on Computer and Communications Security, ACM (2002) 41–47
3. Chan, H., Perrig, A., Song, D.X.: Random key predistribution schemes for sensor networks. In: IEEE Symposium on Security and Privacy, IEEE Computer Society (2003) 197–
4. Lee, J., Stinson, D.R.: On the construction of practical key predistribution schemes for distributed sensor networks using combinatorial designs. ACM Trans. Inf. Syst. Secur. **11** (2008)
5. Çamtepe, S.A., Yener, B.: Combinatorial design of key distribution mechanisms for wireless sensor networks. In Samarati, P., Ryan, P.Y.A., Gollmann, D., Molva, R., eds.: ESORICS. Volume 3193 of Lecture Notes in Computer Science., Springer (2004) 293–308
6. Lee, J., Stinson, D.R.: Deterministic key predistribution schemes for distributed sensor networks. In Handschuh, H., Hasan, M.A., eds.: Selected Areas in Cryptography. Volume 3357 of Lecture Notes in Computer Science., Springer (2004) 294–307
7. Lee, J., Stinson, D.R.: A combinatorial approach to key predistribution for distributed sensor networks. In: IEEE Wireless Communications and Networking Conference, WCNC 2005, New Orleans, LA, USA. (2005)
8. Ruj, S., Roy, B.K.: Key predistribution using partially balanced designs in wireless sensor networks. In Stojmenovic, I., Thulasiram, R.K., Yang, L.T., Jia, W., Guo, M., de Mello, R.F., eds.: ISPA. Volume 4742 of Lecture Notes in Computer Science., Springer (2007) 431–445
9. Dong, J., Pei, D., Wang, X.: A key predistribution scheme using 3-designs. In: INSCRYPT. (2007)
10. Chakrabarti, D., Maitra, S., Roy, B.K.: A key pre-distribution scheme for wireless sensor networks: Merging blocks in combinatorial design. In Zhou, J., Lopez, J., Deng, R.H., Bao, F., eds.: ISC. Volume 3650 of Lecture Notes in Computer Science., Springer (2005) 89–103
11. Stinson, D.: Combinatorial Designs: Constructions and Analysis. Springer-Verlag, New York (1987)
12. Du, W., Deng, J., Han, Y.S., Varshney, P.K.: A key predistribution scheme for sensor networks using deployment knowledge. IEEE Trans. Dependable Sec. Comput. **3** (2006) 62–77