CHAPTER 5

PATTERN VIABLE RESTORATION (PVR) TECHNIQUE WITH NOVEL VIABLE KEY (VK) SCRIPT FOR SECURED DATA TRANSMISSION IN WSN

5.1. INTRODUCTION

Wireless Sensor Networks (WSNs) consist of hundreds of tiny devices which have the capability of sensing, processing and communicating the data. These networks are usually deployed in unmanned, unattended and remote places. This Thesis concentrates on monitoring the health of the electrical system and to develop a secured routing scheme for electrical data transmission. The role of WPANs in system health monitoring of electrical grid is sensor signal-based machine signature analysis, remote energy monitoring, power theft monitoring and fault diagnostics of the equipment in the grid. Due to this reason, wireless sensor nodes need to be configured with security mechanism to shield the data or plain text over the wireless medium.

Designing of key management algorithm and development of secure routing protocol are two issues that have been focused by large amount of research. For secure and reliable data transmission, most researchers had developed various encryption algorithms based on crypt analysis. For example, the Reed Solomon (RS) code based security algorithms have been proposed by Wang et al (2010) and key management algorithms by Jankowski & Laurent (2011), Pietro et al (2003). The various security schemes have been proposed in literature such as the design of secure and

efficient routing protocol by Alwan & Agarwal (2013), Liu et al (2012), the design of secure data aggregation protocol by Estrin et al (1999), Bhoopathy & Parvathi (2012) and Roy et al (2012).

Tawalbeh et al (2013) proposed the elliptic curve cryptography for multimedia compression. The selective encryption algorithm was used during compression and perceptual encryption has been achieved by using selective bit-plane encryption which was used in multimedia before the compression process.

This Chapter discusses a novel technique to build secured data transmission using pattern viable restoration of data while transmission and while reception. The patterns are specifically formulated using English alphabet script and termed as *Viable Key (VK) script*. The VK array formulated can be as pattern viable restoration based on recurrent keys or an array formulated as pattern viable restoration based on asymmetric keys. The novelty of the Viable Key (VK) script is that they can be used to formulate 256 script patterns of VK. Since it is pattern based, the security can be achieved through data array formulation.

5.1.1. Outcome of Encryption / Decryption Algorithms

The secure data transmission algorithms are needed to protect the data over the wireless medium from stealing of data in WSN. Most of the secure data transmission algorithms are developed based on encryption. The following are some of the outcomes of data encryption / decryption algorithm (Stallings 2005):

Data confidentiality: The main aspect of confidentiality is the protection of transmitted message from the attacks and other aspect is the protection of traffic flow from analysis. The designed security mechanisms should

guarantee that the transmitted message in the network is not understood by anyone except the intended receiver.

Data Integrity: During data transmission, there may be the possibility to alter the data within the packet or any malicious node may manipulate the data. Loss of data also occurs due to the harsh communication environment. Thus, data integrity ensures that the transmitted message cannot be altered by anyone as it transmits from the sender to the receiver.

Availability: Availability is the property of the node or its resource being accessible by an authorized receiver. Whenever users make a request, the system is readily available and provides services. To achieve this goal, different approaches have been proposed by many researchers. Some approaches have been focused on successful delivery of the messages to its authorized receiver.

Authentication: Any recipient of the message in the network has to verify the identity of the sender whether the received information is coming from the authorized sender or not. Various authentication schemes were proposed by Liang & Shi (2005) and Capkun & Hubaux (2006). Message Authentication Code (MAC) was discussed in the literature and design of shared secret key among the nodes is also one of the authentication scheme used in literature to achieve data authentication.

Data freshness: Data freshness implies that the received data is the recent data and it ensures that no old messages replayed by adversary. When WSN nodes use the shared key for message communication, it is the most essential requirement. The two types of data freshness are weak freshness, which offers partial message ordering, but it does not carry any delay information, and strong freshness, which offers a total order on a request-response pair, and allows for delay estimation. It is also useful for time synchronization within

the network. In addition to these algorithms, some other security requirements are Self-organization, Secure localization, and Time synchronization etc.

5.1.2. Challenges in Secure Data Transmission

- To maximize the level of secure data transmission, it requires maximum resources, but WSNs are resource constrained devices. Developing a secure data transmission algorithm offers a good compromise between maximization of security level and utilization of minimum resources which is the main challenge.
- Depending on the capabilities and constraints of the sensor node, the secure data transmission algorithm is to be developed.
- WSN can be easily attacked by different types of attacks ranging from passive attack to active attack because of the ad-hoc networking topology. Thus, the security algorithm designed by the user should shield the transmitting data between the sender and the receiver.
- The radio is the communication medium between the two devices in WSN. The wired security schemes are not possible in WSN. Thus, design of secured data encryption algorithm and secured routing protocol is still a challenge.

5.2. LITERATURE SURVEY

The Hybrid Multipath Scheme for Secure and Reliable Data Collection (H_SPREAD) protocol proposed by Lou & Kwon (2006) has some advantages and drawbacks. This H-SPREAD protocol had been developed based on the N-to-1 multipath routing protocol. The multiple node disjoint paths were found first. Among these paths M disjoint paths were selected for

delivering the message at the BS. In this protocol the message is first divided into shares and each share has N packets. These shares were transmitted over M paths. Due to the use of N shares among M paths, Lou & Kwon (2006) proposed the H-SPREAD which is a secured protocol. Even though it is a secured protocol, the maximum security can be achieved by compromising M paths for transmission. When there is any node failure or path failure, there may be a packet loss. Thus, there is a difficulty to achieve better data reliability by this protocol. It mainly concentrated on routing protocol based secure transmission and not on data encryption algorithm.

Wang et al (2010) proposed a Reed-Solomon Forward Error Correction (RS-FEC) based selective encryption approach for secure and reliable wireless communication. This algorithm first divide the original data into number of frames and each frame has K symbols and sequence number. Then the original symbol was encoded by RS code, during encryption the K original symbols were encoded into N codeword symbols. In Mode 1, only $N-K+b_1$ symbols were encrypted by the sender, in Mode 2, $K+b_2$ symbol had been encrypted. The b_1 and b_2 were predefined values defined by the users. From the original data only 'w' symbols were encrypted and transmitted. At the receiver end only 'w' symbols were decrypted and decoded. The original data was constructed based on the sequence number. This algorithm uses the partial encryption. The algorithm discussed in this literature is based on the code theory algorithm and it consumes more power for encoding and decoding.

Hybrid key management based security mechanism was discussed by Pan et al (2011) for health care monitoring of elderly people using WSN. Two modified algorithms based on Feistel cipher structure were proposed. This algorithm generated the random 128 bits key for four round functions whereas in Feistel cipher 16 round functions were used. Four keys were

generated by dividing 128 bits into four 32 bits sub-keys named as K_1, K_2, K_3, K_4. The plain text of 64 bit was equally divided into two parts. Each part is 32 bit long. The encryption and decryption were done using these four sub-keys and the round function F. The 64 bit plain text was encrypted by using $K_1K_2K_3K_4$ and decrypted by using $K_4K_3K_2K_1$. This algorithm used the same concept of Feistel cipher structure with reduced number of sub-keys. The encryption was done using Equations (5.1) and (5.2)

$$LE_i = RE_{i-1}; \qquad i = 1,2,3,4 \tag{5.1}$$

$$RE_i = LE_{i-1} \oplus F(RE_{i-1}, K_i); \qquad i = 1,2,3,4 \tag{5.2}$$

and decryption was done using Equation (5.3) and Equation (5.4)

$$LD_i = RD_{i-1}; \qquad i = 1,2,3,4 \tag{5.3}$$

$$RD_i = LD_{i-1} \oplus F(RD_{i-1}, K_{4-i+1}); \qquad i = 1,2,3,4 \tag{5.4}$$

where LE and RE be the 32-bit left parts and 32-bit right parts of encrypted data. Similarly, LD and RD be the 32-bit left parts and 32-bit right parts of decrypted data. This scheme requires more processor power as it has logical operation like XOR.

Shivamurthy et al (2012) proposed a secure Energy Efficient Node Disjoint Multipath Routing Protocol (EENDMRP). The security provided by this protocol was designed using digital signature based crypto system which used the MD5 hash function and RSA algorithm. In this protocol the source node selected the node-disjoint path and *M* messages are transmitted through the selected path. Encryption of these messages had been done by using MD5 hash function. The digital signature given in Equation (5.5) was used by source node to encrypt message digest H(M) with its private key

$$d_{sign} = (H(M))^{d} \bmod n \tag{5.5}$$

where d is the private key, n=p*q and p, q are random number but p≠q.

The source node forwarded the encrypted data packets to the neighbour node. These data packets were decrypted by the neighbour node and the decrypted data was compared with H(M) when it matched the data packets which were forwarded to the next hop node orelse it dropped the packets. In this algorithm, decryption should be done by each and every node in the node disjoint path. Due to this reason each node consumes energy for the decryption and comparison process.

The homophonic substitution and error-correction coding were proposed by Oggier & Mihaljevic (2014) for achieving the cryptographic security. The main focus of this work was the performance analysis of the transmission of the encrypted data over a noisy channel. The data was encoded by the encoder before encryption. The modulo2 addition had been used for encrypting the data. The extra randomness was achieved by using wire-tap channel coding combined with channel noise. The security analysis for passive advisory and active advisory have also been addressed.

This Thesis proposes pattern based encryption algorithm, which does not require any logical or mathematical based processing. The proposed *Pattern Viable Restoration(PVR) technique with Viable Key (VK) Script* in this Thesis consumes very less energy because it uses script based encryption technique. The PVR technique with VK script uses only the scripts for encryption in which there is only the replacement of the bits and does not require any complex computation. The advantage of the proposed technique is encryption and decryption which is done only one time by the source node and destination node respectively. The intermediate node in the routing path

does not have the knowledge about the data. Thus the router nodes consume less energy.

5.3. EXISTING ENCRYPTION ALGORITHMS

Public key encryption is the most commonly used cryptographic method. It uses asymmetric-key pair which consists of public key and private key. A *public key* is known to everyone and a *private* key also known as *secret key* is known only to the receiver of the message. More security related research works have been done based on the public key encryption method. In public key encryption algorithms, both public key and private key need to be designed. In these existing algorithms data or messages are encrypted using private key which is decrypted using public key. Similarly, the data encrypted by public key is decrypted using private key. Symmetric key crypto system is also one type of public key crypto system. Some wireless sensor network based security applications used the symmetric key crypto system for secure data transfer. The symmetric key crypto system uses the same key for both data encryption and decryption. It will reduce the overhead of designing multiple secret key. In literature, most research works designed the mathematical model like mod, power etc., based symmetric keys. Some of the existing encryption standards are discussed in the following section:

5.3.1. Data Encryption Standard (DES)

Data Encryption Standard or DES (Stallings 2005) is a most commonly used cryptosystem. It uses the Feistel network to construct a block cipher. DES applies a 64-bit key to each 64-bit block of data. In DES 64-bit plain text is encrypted using a key which is having a length of 64-bit. The three steps followed for encryption using DES algorithm is as follows:

- A bit stream is constructed based on the fixed initial Permutation. Actually it divides the given plain text of 64-bit into two equal halves (left and right). Each halves has 32-bit.
- It uses 16 round functions to generate 16 keys. These 16 round iterations are executed using the following equations:

$$L_i = L_{i-1} \tag{5.6}$$

$$R_i = R_{i-1} XOR f(R_{i-1}, K_i) \tag{5.7}$$

where f is the round function L_i, R_i are the left and right 32-bit stream.

- The cipher text is obtained by applying inverse permutation to left and right 32-bit string.

5.3.2. RSA (Rivest Shamir Adleman)

RSA cryptographic algorithm was first introduced by Ron Rivest, Adi Shamir, and Leonard Adleman in 1978 (Stallings 2005). The public-key cryptosystem as well as digital signatures were used by RSA algorithm. The public keys and private keys were used for encryption and decryption of the message in RSA algorithm. The receiver transmits the public key to the sender and the private key is kept secret. The public key and private key component were generated using prime numbers and Euler's function.

Whenever the sender wishes to transmit message, first the plain text of message M is converted into integer 'm' using padding scheme. Such that $0 \leq m < n$, where $n = pq$ and p, q are two distinct prime numbers and e is co-prime number. Then the encryption can be done using Equation (5.8)

$$c = m^e \pmod{n} \tag{5.8}$$

Finally encrypted cipher text *c* is transmitted to the receiver.

The decryption has to be done in receiver end. In receiver end, the integer *m* is recovered from the encrypted cipher text *c* using its own private key, integer *m* and *d* the multiplicative inverse of *e*. This can be obtained by using the Equation (5.9)

$$m = c^d \pmod{n} \tag{5.9}$$

After getting the integer 'm' the original message 'M' can be obtained by using reverse padding scheme.

5.3.3. Advanced Encryption Standard (AES)

The Advanced Encryption algorithm discussed by Jankowski & Laurent (2011) is one of the symmetric key algorithm. This means that the encryption and decryption of the data can be achieved by using the same key. AES and DES both are symmetric block cipher. Unlike DES, it uses the Rijndael algorithm which supports key size of 128, 192, 256 bits with data handled in 128-bit blocks. Depending on the key size, AES uses the variable number of rounds for converting the plain text into the cipher text. Commonly used number of rounds is:

- 10 rounds for 128-bit keys.
- 12 rounds for 192-bit keys.
- 14 rounds for 256-bit keys

The AES uses "diffusion" and "confusion" methods to encrypt the plain text. The following are the main functions performed by AES to convert the plain text to cipher text:

- SubBytes()
- ShiftRows()
- MixColumns()
- AddRoundKey()

5.3.3.1. SubBytes()

This function is designed based on the “confusion” method. Each byte is processed by the S-box. The S-box is a substitution table. Based on the substitution algorithm one byte is substituted for another byte.

5.3.3.2. ShiftRows()

This function mixes the data within the rows by using *diffusion* method. In this step, row 1 is not shifted, row 2 is shifted by 1 byte. Similarly row 3 and row 4 are shifted by 2 bytes, 3 bytes respectively. In general, the n^{th} row is shifted by n-1 bytes.

5.3.3.3. MixColumns()

In this step, each column of the bytes is combined (multiplied) with the mixed matrix or mixed polynomial. This function takes the four bytes of each column as input and transforms it into four bytes of output. This function also uses the diffusion method to mix the column bytes.

5.3.3.4. AddRoundKey()

In this step, each state of the byte is bit-wise XOR with the sub key which is the same size as the state. Rijndael's key schedule is used to derive the sub key. This is actually encrypts the data.

In AES, the above four steps are repeated for the specified number of rounds depending on the key size.

5.3.4. RC4

RC4 is a variable key-size stream cipher with byte-oriented operations which can be generated by using Pseudo-random generation algorithm (PRGA). RC4 is commonly used in transport layer protocols. The encryption and decryption both have been done using bit-wise exclusive-OR operator. Permutation of all 256 possible bytes and two 8-bit pointers were used in RC4. The variable key length of 40 to 256 bytes was used in key scheduling algorithm and then the stream of bits was generated using PRGA (Stallings 2005).

Eventhough the algorithms presented in the literature and the existing algorithms have some advantages, there still exist some shortcomings that prevent their security applications in WSNs. The main short coming observed from the literature survey is that, most of the algorithms have been developed based on the mathematical evaluation and coding theory concept. Some of the above algorithms used the complicated mathematical evaluation to develop a shared key or secret key. Thus, these algorithms have more processor overhead and the processor requires more memory and consumes more power for processing the data.

To defeat the disadvantages of these algorithms and to develop a more secure data transmission, this Thesis proposes Pattern Viable Restoration (PVR) Technique for Secured Data Communication in WSN. Generally, security can be developed with various types of existing standards and schemes. Secure communication can only be obtained when the transmitting message between two sensor nodes should be encrypted. Secret algorithms using single shared key is not advisable, because an adversary can easily hold the key.

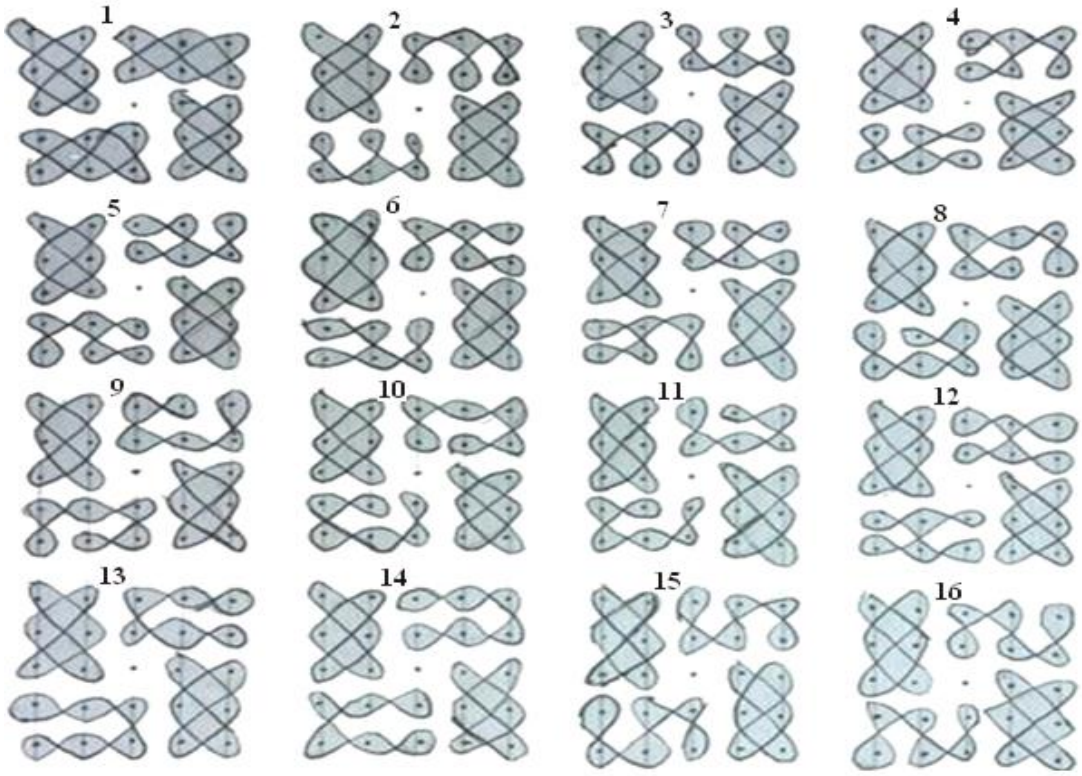
Therefore, the PVR technique is proposed in this Thesis which can develop 16x16 possible VK constructs for designing the secret key. This can also be extended into 256x256 possible secure VK constructs. In the proposed technique, the design of secret key for encryption and decryption is based on the Viable Key constructs and not on the mathematical evaluations. Thus the power required for processing the data is comparatively less.

5.4. VIABLE KEY (VK) CONSTRUCTS BASED ON VK SCRIPT

The various types of sensor nodes such as source node, destination node router node etc. are deployed on the network field for different purposes. The main goal of this Thesis is to communicate the sensed (event) information of electrical data observed from the electrical system to the base station (BS) in a secured manner. For secured data transfer, this Thesis proposes the *Pattern Viable Restoration (PVR)* technique for encrypting the sensed (event) data of the electrical system at source node (cluster member or end device) and decrypting the encrypted data at the destination (BS) node which is deployed at the substation.

The *VK constructs* used for PVR encryption technique is shown Figure 5.1. It shows the 16 different VK constructs which are named with natural numbers from 1 to 16. These 16 VK constructs are formulated from the VK script ‘O’. This VK constructs formation has one symmetric VK constructs ‘O+O’ and fifteen asymmetric VK constructs. The asymmetric VK constructs are numbered as VK constructs 2 through VK constructs 16. Similar to this VK constructs formation; there are 16 different symmetric VK constructs and 240 different asymmetric VK constructs which can be postulated in a similar way. Some of the symmetric and asymmetric VK constructs are shown in Figure 5.2 (Figure 5.2a and Figure 5.2b) and Figure 5.3 (Figure 5.3a and Figure 5.3b) respectively.

The proposed technique has 256 different VK constructs. These VK constructs are framed using English alphabets **E, F, G, H, U, S** and **O**. These alphabets are used as basic shapes to form the original VK constructs for encryption. From these basic alphabets called *Viable Key (VK) scripts*, it is possible to form the derived VK scripts. This Thesis proposes the *derived VK scripts* for **'E' is 'E1'; 'F' is 'F1' 'F2', 'F3'**. Similarly for other VK scripts **G, U and S** is given by '**G1', 'G2', 'G3'; 'U1'; 'S1'** respectively. The remaining VK scripts 'O' and 'U' do not have any derived VK script. Each VK script follows either *2x3 array* or *3x2 array*.



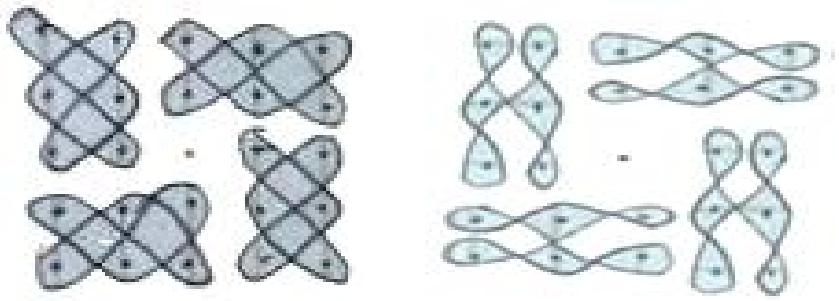
First Row : 'O+O', 'O+E', 'O+E1', 'O+F' Second Row : 'O+F1', 'O+F2', 'O+F3', 'O+G'
Third Row: 'O+G1', 'O+G2', 'O+G3', 'O+H' Fourth Row : 'O+U', 'O+U1', 'O+S', 'O+S1'

Figure 5.1 Formulation of VK Constructs using VK Script 'O' for Pattern Viable Restoration Technique

The various combinations of these VK scripts are used to form the different VK constructs which are unique for data encryption. For example,

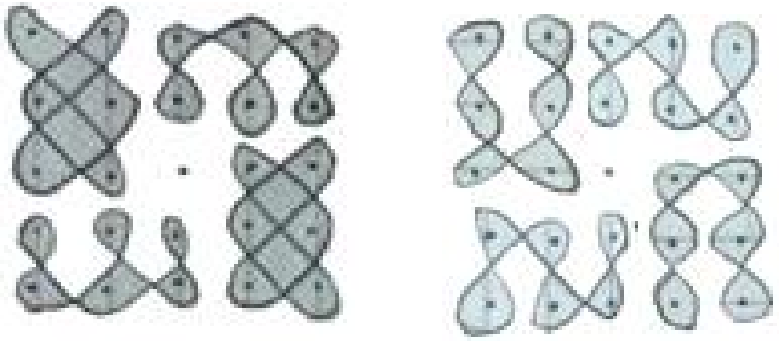
the VK constructs formation using VK script ‘O’ is explained as follows. The VK script ‘O’ itself forms its original symmetric VK constructs called **‘O+O’** which is denoted as number 1 in Figure 5.1. The VK script ‘O’ is combined with other VK scripts E, F, G, H, U, and S form the VK constructs like **O+E, O+F, O+G, O+H, O+U and O+S** and are denoted as 2, 4, 8, 12, 13 and 15. Similarly the derived VK constructs are obtained using the derived VK scripts E1, F1, F2, F3, G1, G2, G3, U1 and S1 are **O+E1, O+F1, O+F2, O+F3, O+G1, O+G2, O+G3, O+U1 and O+S1** and are denoted as 3, 5, 6, 7, 9, 10, 11, 14 and 16. Therefore, *16 different VK constructs* are obtained from a single VK script ‘O’. Similarly the remaining VK scripts E, F, G, H, U, S, E1, F1, F2, F3, G1, G2, G3, U1 and S1 can be used to form rest of *240 different VK constructs*. Among these *256 VK constructs* 16 VK constructs in the combination of E+E, E1+E1, O+O etc., are called as *symmetrical VK constructs* and the remaining 240 are called as *asymmetrical VK constructs*.

The Viable Key (VK) for data encryption is designed based on the symmetrical and asymmetrical VK constructs. The secret key is generated based on the symmetrical VK constructs which is named as *“Pattern Viable Recurrent (PVR) Key”* and the asymmetrical VK constructs named as *“Pattern Viable Asymmetric (PVA) Key”*. Some example of VK constructs used for PVR key and PVA key are given in the Figure 5.2 and Figure 5.3 respectively.



(a) ‘O+O’ VK constructs **(b) ‘H+H’ VK constructs**

Figure 5.2 Symmetrical VK Constructs used for PVR Key



(a) 'O+E1' VK constructs **(b) 'U+S' VK constructs**

Figure 5.3 Asymmetrical VK Constructs used for PVA Key

These VK constructs can be represented by *5x5 array* and is shown in Figure 5.4. $R_{11,}$ R_{12} ... R_{55} are the variables to indicate the bit position of the sensed data during encryption and decryption. For example R_{23} means the position of the bit is second row third column. Figure 5.5 shows the bit representation of the pattern used for encryption and decryption. Using bit representation, the corresponding bit in the 'R_{23}' is 'b_7'.

R_{11}	R_{12}	R_{13}	R_{14}	R_{15}
R_{21}	R_{22}	R_{23}	R_{24}	R_{25}
R_{31}	R_{32}	R_{33}	R_{34}	R_{35}
R_{41}	R_{42}	R_{43}	R_{44}	R_{45}
R_{51}	R_{52}	R_{53}	R_{54}	R_{55}

Figure 5.4 Formulation of 'O+O' using 5x5 Array

b_0	b_1	b_2	b_3	b_4
b_5	b_6	b_7	b_8	b_9
b_{10}	b_{11}	b_{24}	b_{12}	b_{13}
b_{14}	b_{15}	b_{16}	b_{17}	b_{18}
b_{19}	b_{20}	b_{21}	b_{22}	b_{23}

Figure 5.5 Bit Pattern Representation

5.4.1. Conversion of Plain Text→Cipher Text

The conversion of plain text to cipher text consists of 4 steps, which are

- **ColMat**(): Plain text in single Column - 24 Rows format

  The sensed electrical data observed from the electrical system is converted into 24-bit column matrix. The padding scheme is used to add 0's to make the observed data to 24-bit if data length is less than 24-bit. The 24-bit column array is given by,

$$
\begin{matrix}
b_0 \\
b_1 \\
b_2 \\
b_3 \\
. \\
. \\
. \\
b_{23}
\end{matrix}
$$

- **ConArr():** Convert into 5x5 array format

  The above 24 bit column matrix is converted into 5x5 array which is called **ConArr** which is given by,

$$
\begin{bmatrix}
b_0 & b_1 & b_2 & b_3 & b_4 \\
b_5 & b_6 & b_7 & b_8 & b_9 \\
b_{10} & b_{11} & b_{24} & b_{12} & b_{13} \\
b_{14} & b_{15} & b_{16} & b_{17} & b_{18} \\
b_{19} & b_{20} & b_{21} & b_{22} & b_{23}
\end{bmatrix}
$$

In this array the middle bit b_{24} indicates the *Recurrent Check Bit (RCB)*.

- **SubPos**(): Substitute position for the bits

  In the third step, the bits in 5x5 array is replaced with the position of the bits in terms of rows and columns and is given by,

R_{11}	R_{12}	R_{13}	R_{14}	R_{15}
R_{21}	R_{22}	R_{23}	R_{24}	R_{25}
R_{31}	R_{32}	R_{33}	R_{34}	R_{35}
R_{41}	R_{42}	R_{43}	R_{44}	R_{45}
R_{51}	R_{52}	R_{53}	R_{54}	R_{55}

- **IdenPat()** : Generate keys

  In the last step, the VK constructs-id is first selected. Then based on the VK constructs-id the 24 bit PVR/PVA key is generated.

5.4.2. Conversion of Cipher Text → Plain Text

The above four steps are repeated in the reverse to obtain the plain text from the cipher text. The steps for decrypting the cipher text at the receiver end are as follows:

- **InvIdenPat()**

  From cipher text, first to identify the type of key is either PVR or PVA using RCB bit, which is the last bit of the cipher text, used for encrypting plain text during the encryption process and then to identify the unique VK constructs-id.

- **InvSubPos()**

  The position of the bit is reconstructed from the PVR/PVA key. The 5x5 array is constructed with position of bit.

- **InvColArr()**

  Using the PVR/PVA key the bit-positions in 5x5 array are replaced with bits.

- **InvColMat()**

  The 5x5 array of bits is converted into the 24 bit column matrix. Now, the plain text is reconstructed from column matrix by reverse permutation.

5.5. PATTERN VIABLE RESTORATION WITH ENCRYPTION/ DECRYPTION USING VK CONSTRUCTS

In cryptography, *encryption* is generally defined as the process of encoding messages or information in such a way that it can be read only by the authenticated node. Design of efficient encryption algorithms are used to secure the data which are encrypted and communicated over the wireless medium. The original message or information also known as plaintext is encrypted using the designed encryption algorithms and then changed into an unreadable text called *cipher text*. This is usually done with the use of an encryption key, which specifies how the message is to be encoded. Any adversary can see the *cipher text* but do not have the knowledge about the original message. An authorized party can only be able to decode the *cipher text* using a decryption algorithm. This decoding process usually requires a secret decryption key that only the authorized person has access to it.

The block diagram for encryption and decryption algorithm based on the PVR or PVA key is shown in Figure 5.6 and Figure 5.7 respectively. In

encryption paradigm, the VK script based secret PVR key/PVA key is generated using VK constructs. The unreadable *cipher text* is generated from the readable *plaintext* by encrypting the plain text using secret PVR key/PVA key. In decryption paradigm, the unreadable *cipher text* is decoded using secret PVR key/PVA key which is developed by using Pattern Viable Restoration Technique.

5.5.1. Pattern Viable Restoration (PVR) Technique for Encryption

The sensed data is encrypted using PVR/PVA Key which converts the *plain text* to *cipher text*. The steps involved in the encryption process shown in Figure 5.6 are as follows:

Step 1: Sensed data (*plain text*) in the form of 24 bit column array

Step 2: Convert the 24 bit column array of *plain text* into 5x5 array

Step 3: Select the RCB bit and place it in the middle of the 5x5 array

Step 4: Replace the bits in 5x5 array into its positions

Step 5: Depending on the RCB, the 24-bit PVR key or PVA key for encryption is used to encrypt the plaintext (payload). The *cipher text* is obtained.

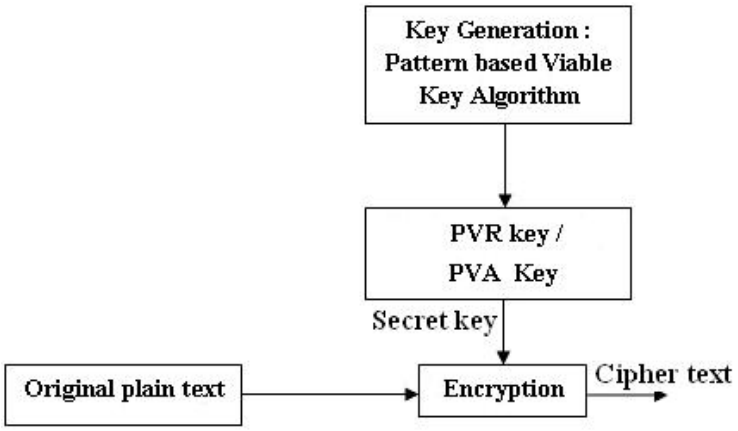


Figure 5.6 Block Diagram for PVR Technique during Encryption Paradigm

5.5.2. Pattern Viable Restoration (PVR) Technique for Decryption

The sensed data is encrypted using PVR secret key or PVA secret key depending on the VK constructs used and the decryption algorithm is shown in Figure 5.7. The encrypted data is communicated to the receiver end through the communication protocols. To reconstruct the plain text, the following steps need to be done during decryption at the receiver end.

Step 1: The encrypted payload (cipher text) is received by the receiver at the receiver end.

Step 2: The bit of RCB field is checked, based on this bit, the type of VK constructs used is identified either symmetrical (PVR) or asymmetrical (PVA).

Step3: After the identification of the VK constructs, select the corresponding Viable Key. If it is symmetrical do step 4 otherwise do step 5.

Step 4: From the content of pattern viable restore field the corresponding VK constructs-id is identified and then using the identified symmetrical VK constructs the payload bits are converted into array form using PVR key then go to step6.

Step 5: From the content of pattern viable restore field the corresponding VK constructs-id is identified and then using the identified asymmetrical VK constructs the payload bits are converted into array form using PVA key.

Step 6: The 5x5 bit pattern is converted to the one column array

Step 7: Finally the sensed data is retrieved from the column array.

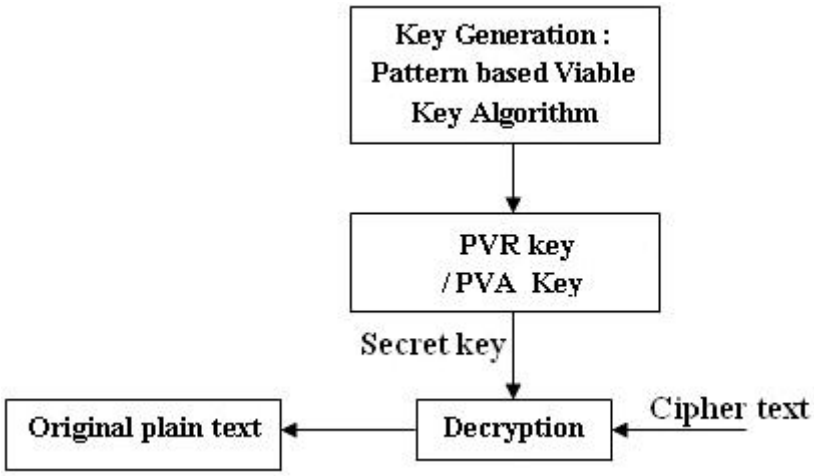


Figure 5.7 Block Diagram for PVR Technique during Decryption Paradigm

After completion of the above six steps, the receiver can get the decrypted sensed data value (plain text) from the encrypted data (cipher text). This technique shows that the PVR and PVA secret keys are used for the secure and reliable data transfer. The design of PVR technique using VK constructs improves the data confidentiality.

5.6. DESIGN OF VK SCRIPT BASED VIABLE KEY

The Viable keys are designed using VK constructs. Different VK scripts are combined to form VK constructs. The design of Viable Keys (PVR/ PVA) is explained in the next section using one symmetrical VK constructs ‘O+O’ for PVR key and one asymmetrical VK constructs ‘O+E1’ for PVA key.

5.6.1. Design of Viable Key based Restoration Technique

The Figure 5.8 shows 256 possible VK constructs. Every numbered VK constructs is used to design PVR due to symmetrical pattern of VK script and the rest would be PVA due to asymmetrical pattern of VK script.

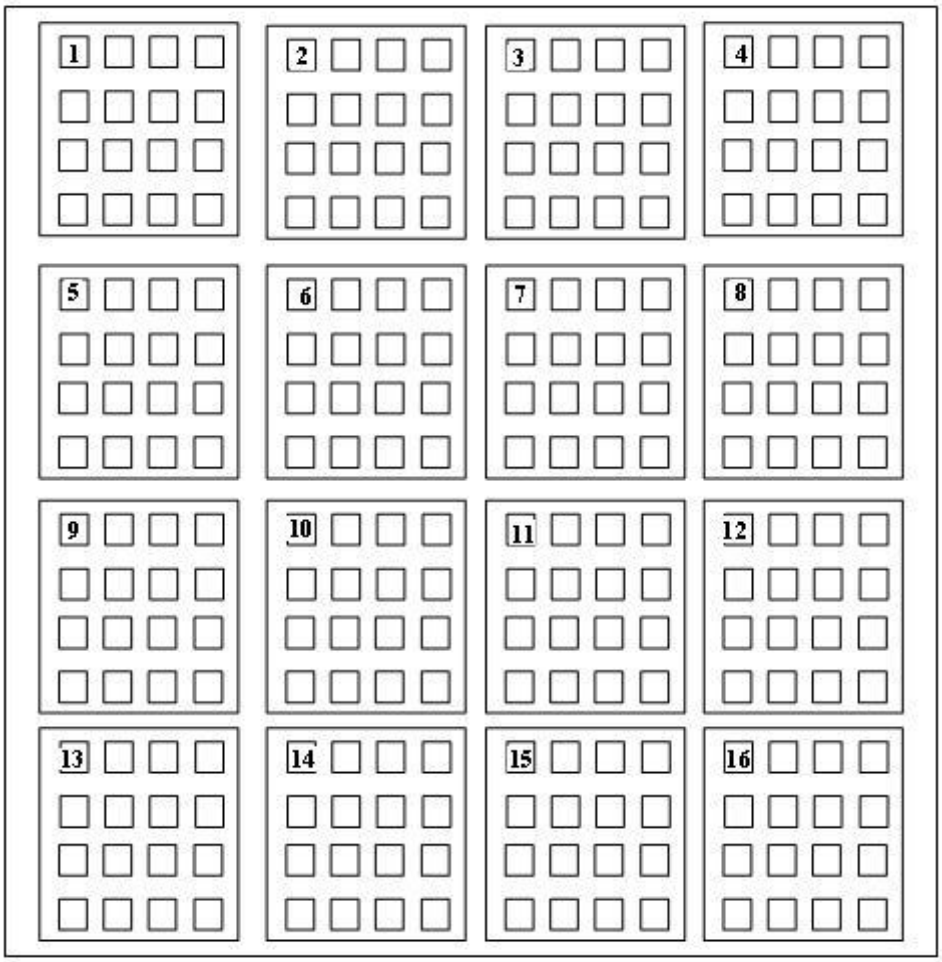


Figure 5.8 Possible Combination of 256 VK Constructs using VK Scripts

The sensors placed on the sensor nodes which are connected to the electrical system continuously monitor the system health by sensing the parameters of the system such as temperature, current, voltage, and acceleration etc., of the electrical system. These sensors transmit the data only when an event (abnormality in system) occurs, the sensed data is first converted into 25- bit data. This means that the first 24 bits represent the 3-Byte payload bit and the last bit b_{24} represents the Recurrent Check Bit. The 25-bit representation of the plain text is the sensed data D(S) and is given by

$$Plain\ text = D(s) = b_0 | b_1 | b_2 | b_3 | b_4 | b_5 | \ldots\ldots | b_{20} | b_{21} | b_{22} | b_{23} | b_{24} \qquad (5.10)$$

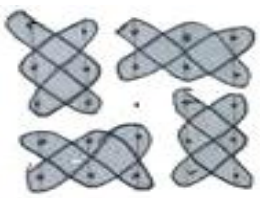
The *Recurrent Check Bit (RCB)* b_{24} is used to identify whether the encryption is done based on symmetrical VK constructs or asymmetrical VK constructs. If it is identified, then it can easily identify the secret key which is used for encryption at the transmitter. The secret key is either Pattern Viable Recurrent (PVR) key or Pattern Viable Asymmetric (PVA) key. If the VK constructs used for encryption is symmetrical, the decryption can be done using PVR key and otherwise if asymmetrical the PVA key would be the suitable key for decryption.

During encryption at the transmitter end, the D(s) is converted to the array. The first five bits b_0, b_1, b_2, b_3, b_4 are placed in the first row of the array and next five bits b_5, b_6,b_7, b_8, b_9 are placed in the second row. The middle row consists of the b_{10}, b_{11} in the first two columns and b_{12}, b_{13} are in the last two columns. The third column of the middle row has the bit b_{24} called RCB bit. The remaining bits b_{14}-b_{23} are placed in the last two rows. Therefore the bit formation of the array BM(S) is given by,

$$BM(S) = \begin{bmatrix} b_0 & b_1 & b_2 & b_3 & b_4 \\ b_5 & b_6 & b_7 & b_8 & b_9 \\ b_{10} & b_{11} & b_{24} & b_{12} & b_{13} \\ b_{14} & b_{15} & b_{16} & b_{17} & b_{18} \\ b_{19} & b_{20} & b_{21} & b_{22} & b_{23} \end{bmatrix} \tag{5.11}$$

The symmetrical VK constructs used for the secret key PVR(S) is given by,

$$PVR(S) = \text{`}O+O\text{'} = \tag{5.12}$$



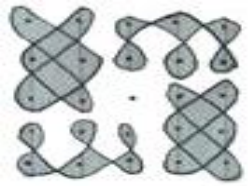
The dot (.) at the center bit is called Recurrent Check Bit (RCB)

The generated PVR key using Equation (5.12) is given by,

$$PVR(S) = R_{11}|R_{22}|R_{31}|R_{12}|R_{21}|R_{32}|R_{23}|R_{14}|R_{25}|R_{13}|R_{24}|R_{15}|R_{43}|R_{52}|R_{41}|R_{53}| \\ |R_{42}|R_{51}|R_{55}|R_{44}|R_{35}|R_{54}|R_{45}|R_{34} \tag{5.13}$$

Similarly, asymmetrical VK constructs used for the secret key PVA(S) is given by,

$$PVA(S) = \text{`}O+E1\text{'} = \tag{5.14}$$



The asymmetrical VK constructs based crypto key is known as PVA key and is given by

$$PVA(S) = R_{11} | R_{22} | R_{31} | R_{12} | R_{21} | R_{32} | R_{13} | R_{23} | R_{14} | R_{25} | R_{15} | R_{24} | \\ | R_{53} | R_{43} | R_{52} | R_{41} | R_{51} | R_{42} | R_{55} | R_{44} | R_{35} | R_{54} | R_{45} | R_{34} \tag{5.15}$$

5.6.2. Pattern Viable Restoration using PVR Key for Encryption/Decryption

The PVR key is designed based on the symmetrical VK constructs. There are 16 symmetrical VK constructs which are formed based on the VK scripts. The PVR key generates 16 different encryption/decryption key format using 16 symmetrical VK constructs for assembling data in the payload field. Among these 16 different PVR key, the following section explains the design of one PVR key based on the symmetrical VK constructs ‘O+O’. The PVR key designed from the ‘O+O’ symmetrical VK constructs for assembling data into payload field is shown in Table 5.1.

Table 5.1 Format for PVR Key

Bits	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{24}
Position of bits	R_{11}	R_{22}	R_{31}	R_{12}	R_{21}	R_{32}	R_{23}	R_{14}	R_{25}	R_{13}	R_{24}	R_{15}	R_{33}
Position of bits	R_{43}	R_{52}	R_{41}	R_{53}	R_{42}	R_{51}	R_{55}	R_{44}	R_{35}	R_{54}	R_{45}	R_{34}	
Bits	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}	b_{17}	b_{18}	b_{19}	b_{20}	b_{21}	b_{22}	b_{23}	

The sensed data is first obtained from the sensor and RCB is added and is converted to the 25-bit format. In this 25 bit, first 24-bit from 1 to 24 indicates the payload reading of the sensor and the last bit (25^{th} bit) indicates the Recurrent Check Bit (RCB).

5.6.3. Pattern Viable Restoration using PVA Key for Encryption/Decryption

The PVA Key based Encryption method is suitable for asymmetrical VK constructs. The proposed technique can generate 240 asymmetric VK constructs. Thus this key can generate 240 different PVA key. From these 240 PVA key this section explains design of one PVA using asymmetrical VK constructs ‘O+E1’ and its format is shown in Table 5.2. Similar to the PVR key, PVA key is also having 25-bit including RCB. In PVA key RCB is the last bit like PVR key.

Table 5.2 Format for PVA Key

Bits	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{24}
Position of bits	R_{11}	R_{22}	R_{31}	R_{12}	R_{21}	R_{32}	R_{13}	R_{23}	R_{14}	R_{25}	R_{15}	R_{24}	R_{33}
Position of bits	R_{53}	R_{43}	R_{52}	R_{41}	R_{51}	R_{42}	R_{55}	R_{44}	R_{35}	R_{54}	R_{45}	R_{34}	
Bits	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}	b_{17}	b_{18}	b_{19}	b_{20}	b_{21}	b_{22}	b_{23}	

5.7. SECURED DATA COMMUNICATION SCHEME

The event information which occurs in sensor node is encrypted using PVR technique with VK script and is communicated to the receiver end using Optimized Multipath Routing for Balanced Load Distributing (OMR BLD) protocol which is explained in Chapter 4.

The same assumptions which are used for OMR-BLD protocol is considered here for communicating the sensed event information of the system. Some of the assumptions are made in PVR technique with VK constructs for communicating the event information and is given by:

- It uses the homogeneous system model. All the nodes in the network that contain the same initial energy. They are static in nature and every node has a unique node-ID.
- Whenever an event occurs the source (Cluster Member-CM) node encrypts the data and transmits the encrypted data to Base Station.
- The source node and the destination node only know the VK constructs based secret key. The intermediate router nodes do not have the knowledge about data.

5.7.1. Implementation of PVR Technique using VK Constructs with IRIS Motes

The Pattern Viable Restoration technique is implemented on the TINY-OS based cross-bow IRIS motes to make the transmission is secured and the performance of Pattern Viable Restoration (PVR) technique is compared with standard encryption algorithms such as DES, AES and SHA algorithms.

5.7.1.1. Configuration of IRIS MOTE- XM2110

The IRIS-XM2110 (IRIS Data Sheet information) shown in Figure 5.9 is the sensor board used as wireless sensor node or mote. The XM2110 is developed based on the Atmel ATmega1281 micro controller. It includes a processor that operates on Tiny-OS operating system environment. The ATMEGA 1281 is a low-power microcontroller which executes the designed technique from its internal flash memory. It has ZigBee protocol based RF 230 transceiver for wireless communications. It uses IEEE 802.15.4 protocol standard and operates on 2.4GHz frequency band. The single XM2110 board can be configured to perform various sensor applications, processing of the sensed data and the network/radio communications etc. The IRIS 51-pin expansion connector supports analog inputs, digital I/O, I2C, SPI and UART interfaces. These interfaces make it easy to connect to a wide variety of external peripherals.

Figure 5.9 IRIS Mote View - XM2110

The prototype testbed for PVR technique is shown in Figure 5.10. In this testbed, one IRIS mote is configured as Base Station (BS), one mote is configured as source node that acts as transmitter. The router nodes are deployed between BS and source node. The Hall Effect sensor probe is connected to the transformer for health monitoring. The disturbances or

abnormalities occurred in the transformer like voltage sag and voltage swell are detected as events (Vijayalakshmi & VanajaRanjan 2013) by the sensor. When an event (sag/swell) occurs, the event information is communicated to the BS through the routers. In this experiment the sensed data (event information) is encrypted by PVR technique before the data or event transmission starts. It means that the encryption is done by the transmitter or source node. In the proposed PVR technique, the PVR/PVA key is only known to the transmitter and BS motes. The event information received by the base station is shown in Figure 5.11.

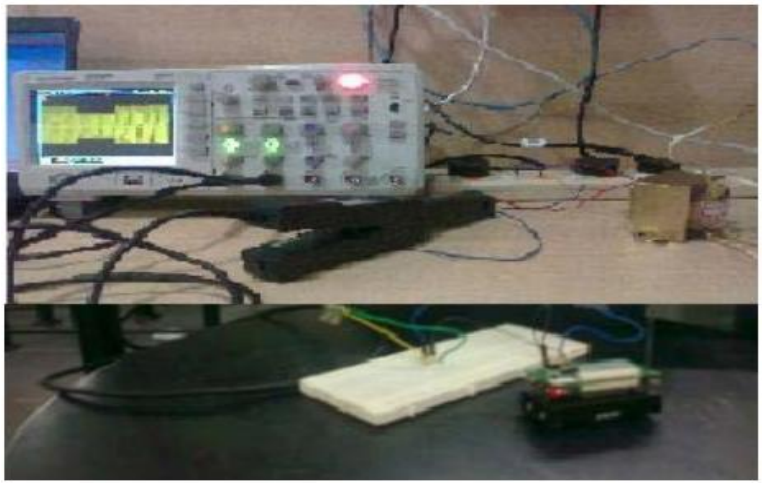


Figure 5.10 Experimental Testbed for Voltage Sag detection-At Transmitter

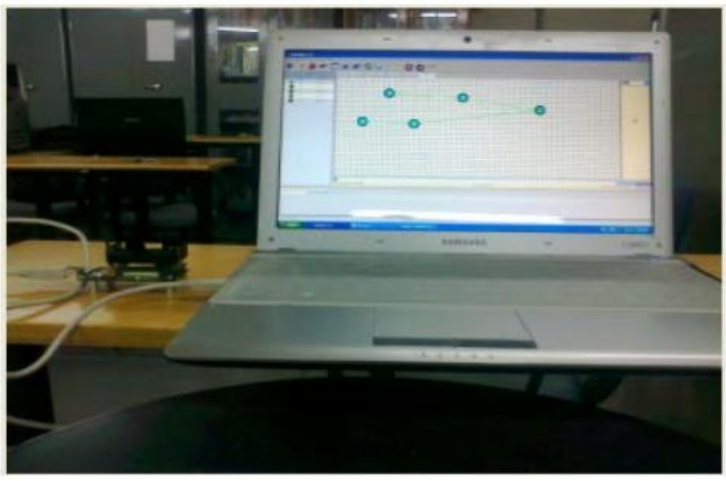


Figure 5.11 Experimental Testbed for Voltage Sag Detection-At Receiver

The IRIS motes are configured to perform as source node, router, Base Station and are deployed on the network field to identify the electrical disturbances. The router nodes do not have the knowledge about the data and encryption techniques. The performance analysis of the various standard encryption algorithms using IRIS motes also has been done. The Network view of the deployed network for event (electrical disturbances) driven application is shown in Figure 5.12 and the graphical observations is shown in Figure 5.13. The TINY-OS based program implementation using IRIS Mote is shown in Appendix 3 (Figure A3.1 to Figure A3.7).

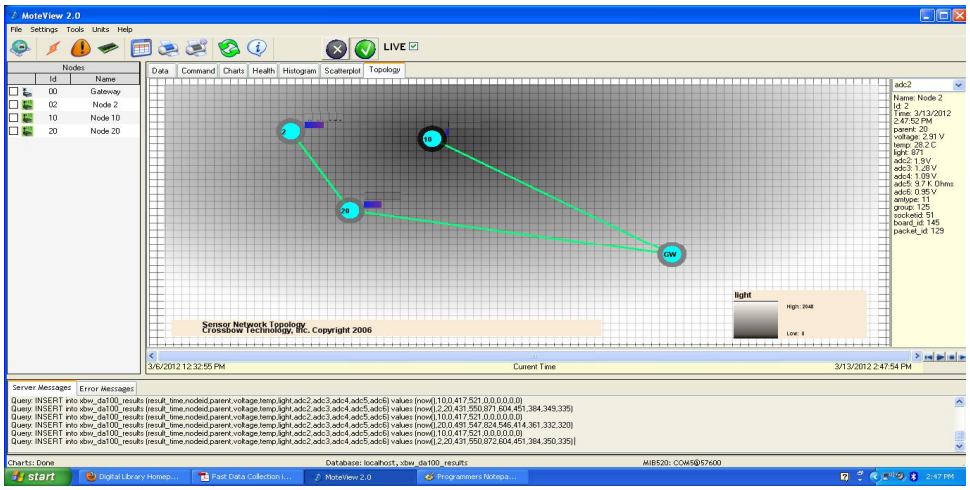


Figure 5.12 Network View of the Deployed Network for Event-driven Application

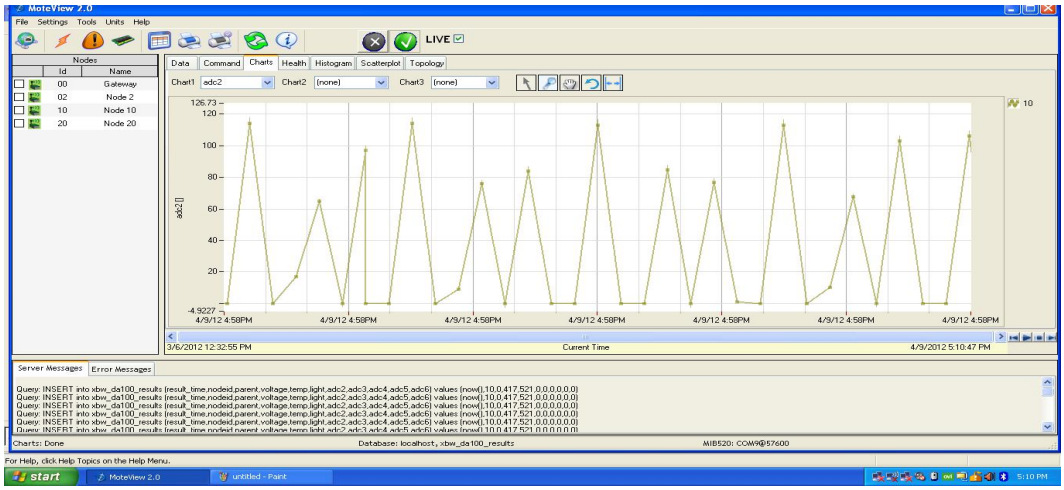


Figure 5.13 Graphical Observations of Electrical Disturbances

5.7.2. Secured Data (SD) Packet Format

The source node transmits the secured data packet which includes Node-id, Length which is equal to the number of bytes between Length and hash sum, Encrypted Pay Load. The payload field of the data packet carries the sensed data (event information) of the electrical system. The length of the payload is variable. In PVR technique, the Viable Key (VK) based encrypted event information is included in the payload field of the packet before the data transmission starts. This Thesis considers the length of the payload field which is defined as 3 bytes which includes 24 bit PVR technique with VK based encrypted payload data. The next field is Recurrent Check Bit (RCB) which is used to identify the VK constructs type (symmetrical and asymmetrical). The last two fields are Hash Sum for error detection and Pattern Viable Restore field which is used to identify the VK constructs like O+O', 'E+E' or 'S+O' etc. The format of the secured data packet is shown in Table 5.3.

Table 5.3 Secured Data (SD) Packet Format

Node-id (2 bytes)	Length (2 bytes)	Encrypted Pay Load (Variable)	RCB (1bit)	Hash Sum (2 bytes)	Pattern Viable Restore (4 bits)

5.7.2.1. Recurrent check bit (RCB)

One bit field in SD packet is Recurrent Check Bit (RCB). The Pattern Viable Restoration (PVR) technique uses PVR key and PVA Key for encryption or decryption depending on the type of VK constructs used. The RCB bit is the bit which is used to identify the type of key used for encryption or decryption. This bit has the value either '0' or '1'. When RCB is '1', the sensed data is encrypted or decrypted based on the symmetric VK constructs and it uses the PVR key of the corresponding VK constructs. When RCB is '0', the asymmetric VK constructs based PVA key is used for encryption or decryption.

5.7.2.2. Hash sum

The next field of the packet format is ‘Hash Sum’. This field is used for detecting transmission errors in transmitted data. The source node calculates the Hash Sum using Equation (5.16)

$$Hash\ Sum\ =\ FF\ -\ \left[Lower\ 8-bit\ of\ \ \left(LP_i + RP_i + RCB_i\right)\right] \qquad (5.16)$$

where LP_i indicates the Left Part of payload bits (b_0-b_{11}) of the packet ‘i’ and RP_i Right Part of payload bits (b_{12}-b_{23}) of the packet ‘i’ and RCB is the Recurrent Check Bit.

The calculated Hash sum is included in the SD packet format before transmitting. At the receiver end the Lower 8-bit of (LP_i+RP_i+RCB) and the Hash Sum are added and checkd whether the sum is equal to ‘FF’. If it is ‘FF’, the receiver decides that there is no transmission error in the received data. For example, if the sensed data has three bytes of 03H, 18H, A5H and RCB bit is 1 means the calculated hash sum is 3EH [FF-lower 8 bit of (03H+18H+A5H+01H)]. These values are included in the secured data packet format and are transmitted to the receiver. At the receiver end the values 03H, 18H, A5H, 01H and 3EH are added. If the sum is equal to FF then the received data has no transmission error or else there is an error. In this case the sum of lower 8 bit of (03H, 18H, A5H, 01H) and 3EH is equal to FFH. This means that the data is received without any transmission error.

5.7.2.3. Pattern viable restore

It is the last field in the SD packet format. It has 1 byte length having a value which ranges from 00000000 to 11111111 (0 to 256). The value included in this field can be used to identify the VK constructs-id for decryption at the receiver end. Generally user can name the VK constructs with unique VK constructs-id like 1, 2, 3,…, 16 for symmetrical VK

constructs shown in Table 5.4 and 1, 2, 3,…, 240 for asymmetrical VK constructs with unique VK constructs-id assignment during encryption process. The user can give any VK constructs-id for any VK constructs for their convenience. This means that the VK constructs-id differ from one user to another user. The unique VK constructs-id assignment for some asymmetrical VK constructs is shown in Table 5.5. This Thesis has designed the encryption technique for one symmetrical VK constructs ‘O+O’ with VK constructs-id assigned 0001 (1) and fifteen asymmetrical VK constructs ‘O+E’ to ‘O+S1’ with assigning VK constructs-id 0001 (1) to 1111 (15). For example, the encryption done in the transmitter end is based on ‘E+F1’ VK constructs. The transmitter transmits the SD packets with value 0011 (3) in the Pattern Viable Restore field and the RCB bit is set to 0. This ‘0’ indicates the asymmetrical VK constructs.

Table 5.4 Pattern Viable Restore for Symmetrical VK Constructs

Sl No.	VK constructs	VK constructs-id
1	E+E	1 (0001)
2	E1+E1	2 (0010)
3	F+F	3 (0011)
4	F1+F1	4 (0100)
5	F2+F2	5 (0101)
6	F3+F3	6 (0110)
7	G+G	7 (0111)
8	G1+G1	8 (1000)
9	G2+G2	9 (1001)
10	G3+G3	10 (1010)
11	H+H	11 (1011)
12	O+O	12 (1100)
13	U+U	13 (1101)
14	U1+U1	14 (1110)
15	S+S	15 (1111)
16	S1+S1	16(10000)

Table 5.5 Pattern Viable Restore for Asymmetrical VK Constructs

Sl. No.	VK constructs	VK constructs –id
1	E+E1	1 (0001)
2	E+F	2 (0010)
3	E+F1	3 (0011)
4	E1+F2	19 (10011)
5	E1+F3	20 (10100)
6	E1+G	21 (10101)
7	F+G1	37 (100101)
8	F+G2	38 (100110)
9	F+G3	39 (100111)
10	F1+H	55 (110111)
11	F1+O	56 (111000)
12	F1+U	57 (111001)
13	F1+U1	58 (111010)
14	S1+U1	239 (11101111)
15	S1+S	240 (11110000)

At the receiver end the data is decrypted using the 3rd VK constructs in the asymmetric form (E+F1) and PVA key is used because of the asymmetric form. In contrast, if it transmits the encrypted data with RCB bit is 1, the receiver uses the 3rd VK constructs of the Symmetric form ‘F+F’ and PVR key is used to decrypt the data. The 3rd VK constructs of symmetric form represents the ‘F+F’ and the 3rd VK constructs of asymmetric form follows ‘O+F’. In this way, the data can be encrypted and decrypted using 16 symmetrical VK constructs and 240 asymmetrical VK constructs. The encryption and decryption technique done in this Thesis are only for 16 VK constructs. The acquired data is encrypted by inserting a new key for, every time slot. The key is known only to the source and the destination. The pattern viable key encryption technique is adaptable for 256 VK constructs.

5.7.3. Construction of Pattern Viable Key

The VK constructs key used for the encryption of data at the transmitter end can be easily decrypted at the receiver end using the VK constructs based keys. The encrypted payload field bits are positioned in the array based on the decryption key. It may be either PVR or PVA key. The key has the position of the bits (row, column format). The bits position in PVR key using 'O+O' VK constructs is shown in Table 5.6. In this construction, the position of the 1st bit (b_0) is same as 19th bit (b_{19}) but the row and column numbers are interchanged. For example, in PVR key the position represented in 1st bit is R_{11} and 19th bit is R_{55}. It indicates that the first bit of the payload received at the receiver end is placed in the 1st row 1st column of the array and the bit received at 19th bit position is placed in the 5th row 5th column of the array. Similarly 2nd bit is R_{22} and 20th bit is R_{44} which means that 2nd bit is placed in 2nd row 2nd column of the array and 20th bit is placed in the 4th row 4th column of the array. From these examples it can be clearly understood that the row/column number are interchangeably used like1←→5, 2←→4 and 3←→3.This is clearly listed in the Table 5.6 for PVR key using 'O+O' VK constructs and Table 5.7 for PVA key using 'O+E1' VK constructs.

Table 5.6 Bits Position in PVR Key using 'O+O' VK Constructs

Bits → Position		**Bits → Position**			**Bits → Position**		**Bits → Position**	
b_0	R_{11}	b_{18}	R_{55}		b_6	R_{23}	b_{12}	R_{43}
b_1	R_{22}	b_{19}	R_{44}		b_7	R_{14}	b_{13}	R_{52}
b_2	R_{31}	b_{20}	R_{35}		b_8	R_{25}	b_{14}	R_{41}
b_3	R_{12}	b_{21}	R_{54}		b_9	R_{13}	b_{15}	R_{53}
b_4	R_{21}	b_{22}	R_{45}		b_{10}	R_{24}	b_{16}	R_{42}
b_5	R_{32}	b_{23}	R_{34}		b_{11}	R_{15}	b_{17}	R_{51}

Table 5.7 Bits Position in PVA key using 'O+E1' VK constructs

Bits → Position		Bits → Position			Bits → Position		Bits → Position	
b_0	R_{11}	b_{18}	R_{55}		b_6	R_{13}	b_{12}	R_{53}
b_1	R_{22}	b_{19}	R_{44}		b_7	R_{23}	b_{13}	R_{43}
b_2	R_{31}	b_{20}	R_{35}		b_8	R_{14}	b_{14}	R_{52}
b_3	R_{12}	b_{21}	R_{54}		b_9	R_{25}	b_{15}	R_{41}
b_4	R_{21}	b_{22}	R_{45}		b_{10}	R_{15}	b_{16}	R_{51}
b_5	R_{32}	b_{23}	R_{34}		b_{11}	R_{24}	b_{17}	R_{42}

5.8. PERFORMANCE ANALYSIS

The impact of message length on processor energy consumption for four different encryption schemes is shown in Figure 5.14. The result shows that the processor energy consumption increases with the increase in message length in all four algorithms. The SHA has maintained constant processor energy consumption with the increase in message length but it is high energy consumption due to its more complex encryption algorithms. The standard encryption algorithms AES and DES have relatively more energy consumption than the proposed technique. These standard encryption algorithms use ten times and sixteen times permutation respectively.

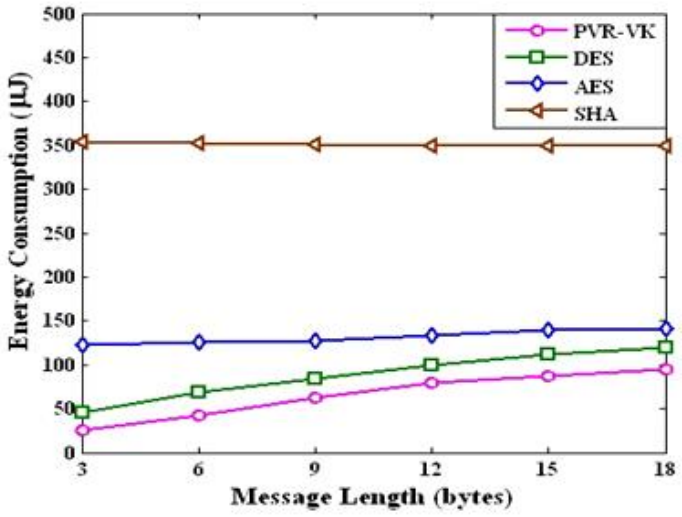


Figure 5.14 Impact of Message Length on Energy Consumption

Therefore these algorithms consume more energy than the proposed technique. As compared to the other three encryption algorithms, the energy consumption of the proposed PVR technique has 2 times lesser than DES, 5 times lesser than AES and 14 times lesser than SHA algorithm. This improved performance is obtained by using PVR technique because of its simple VK script based encryption technique and its concept does not require complex computation calculation.

5.9. CONCLUSION

This Chapter has proposed a novel technique with VK constructs based data encryption key called Viable Key. It is designed specifically for resource constrained devices like wireless sensor nodes. The proposed Pattern Viable Restoration technique using VK script has mainly concentrated on secured, confidential and protected data transmission. This secured data transmission technique is suitable for the system health monitoring of any electrical system like solar panel. In this security technique the data is encrypted only by source node and is decrypted only by destination node. The main contribution of this work is,

- The proposed technique can incorporate various keys among the 256 VK constructs to encrypt data because of which the proposed PVR technique with VK script based secured data technique provides higher security strength.
- The novelty of the proposed technique is: it consumes less processor power as its concept does not have any complex computation calculations. The implementation of Pattern Viable Restoration (PVR) technique on IRIS mote shows that the energy consumed for the proposed technique is 2 times, 5 times

and 14 times lesser than the standard encryption algorithms DES, AES and SHA respectively.

- The VK constructs used for encryption and decryption is chosen by the user. Thus, the PVR and PVA key differ from one user to the other user. Therefore, PVR technique with VK constructs has achieved higher data security and data confidentiality during data transmission.