23

Security and Dynamic Encryption System in Mobile Ad-Hoc Network

Peter H. Yu and Udo W. Pooch
Texas A&M University, Department of Computer Science and Engineering
College Station, TX
USA

1. Introduction

Wireless network technology enables computing devices to communicate with each other without any physical medium. Compared with wired networks, wireless communication provides better connectivity and mobility, which allows mobile devices to access other local area networks or the Internet at anytime and anywhere. The benefits of flexible routing, global connectivity and a highly adaptive potential make mobile ad-hoc networks (MANET) suitable for a wide range of applications in both military and commercial environments, such as battlefields, disaster relief operations, mobile device/ personal networking, mobile information sharing and vehicular networks (Kant et al., 2005); (Liu et al., 2007).

However, maintaining security in wireless ad-hoc networks is quite challenging. First, unlike wired networks that at least have some degree of physical protection, wireless communication over radio waves lacks defined and restricted boundaries. Anyone can connect to the network as long as the transmitted signal strength is strong enough to cover the area (Chan et al., 2005), and therefore, security attacks on data communication, such as passive eavesdropping, packet injection or even violations of confidentiality are widespread. Second, the end-to-end communication in MANET cannot rely on any fixed infrastructure, such as a base station or access points (AP); thus, existing security protocols that are based on a centralized or infrastructure-based network environment will not work in this mobile environment (Hubaux et al., 2001).

Third, in order to achieve better network throughput in such a highly dynamic environment, the default routing protocol does not implement any security protection during end-to-end communication. In addition, the trust relationships between each node are very low as a consequence of the frequently changing topology and membership. Because of this, many attacks can be launched against the routing protocol, giving hackers a major opportunity to insert themselves as one of the cooperative nodes in the network. Therefore, the security protection that is used to ensure the integrity of the mobile ad-hoc network should not only repel external attacks, but also prevent internal attacks launched against the network from any compromised node.

Most security mechanisms rely on data encryption, which is a message combined with a secret key to generate a ciphertext that cannot be revived without the original key. This encryption mechanism can prevent any unauthorized user from gaining access to the secured communication. However, a fixed secret key is vulnerable to deciphering by

capturing sufficient packets or by launching a dictionary attack. Therefore, the most efficient way to protect the network from such attacks is to generate the secret key dynamically and replace it periodically (Ramakrishnan et al., 2005). Furthermore, the protocol applied to the mobile ad-hoc wireless network should be sufficiently flexible to adjust to different levels of security protection to fit the needs of applications in different environments and with varied communication speeds. For example, mobile banking and E-commerce require larger encryption keys for stronger protection, while real-time driven applications such as disaster recovery, stream services like VOIP and online video need to preserve data privacy as well as performance to maintain the quality of services (QoS).

In this chapter, we introduced a new, efficient, low-bandwidth cost and security-enhancing data encryption *i-key* protocol for mobile ad-hoc wireless networks via dynamic re-keying during end-to-end communication. Unlike its counterparts, this secret *i-key* is generated using the previous data as the seed and as next packet encryption before delivery; therefore, only the original sender and authorized client are able to decrypt the message using the unique *i-key* in their possession, which ensures the privacy of their communication.

2. Related work

Wired Equivalent Privacy, or WEP, is an encryption protocol designed by the IEEE 802.11 and Home RF group (Lansford & Bahl, 2000) in an attempt to protect link-level data over radio signals for wireless networks, included both Base Station (BS)-oriented and mobile ad-hoc networks, to the security level closer to wired one. The WEP key used to encrypt data sent over wireless networks consists of two parts: the Initialization Vector (IV) and user pre-shared secret key (PSK). The stream cipher, RC4 used in WEP, expands the IV (40 or 104 bits) and PSK into an arbitrary long "key stream" of pseudorandom bits then XOR with the plaintext to obtain the ciphertext. To decrypt it, the receiver side takes the same steps in the reverse order by the same key stream. In addition, a CRC-32 algorithm is applied to check the data integrity for each data packet in WEP encryption.

Many WEP vulnerabilities and security design issues has been discovered and reported by researchers since the IEEE released it as the standard encryption protocol for 802.11 wireless networks (Gast, 2002); (Miller, 2001); (Prasithsangaree & P. Krishnamurthy, 2004); (J. S. Park & Dicoi, 2003). Therefore, wide attention has been paid by many researchers to the design of new protocols to secure the mobile ad hoc network, such as ARIADNE, DSDV, SEAD, ARAN and SPR (Hu et al., 2005); (Perkins & Bhagwat, 1994); (Hu et al., 2003); (Sanzgiri et al., 2005); (Papadimitratos & Haas, 2002) to provide a solutions for the wireless ad-hoc networks.

Hu et al. developed a secure routing protocol called ARIADNE (Alliance of Remote Instructional Authoring and Distributed Networks for Europe) (Hu et al., 2005), which relies on Dynamic Source Routing protocol (DSR) (Johnson et al., 2002) and symmetric cryptography architecture for end-to-end authentication. On the other hand, based on DSDV (Destination-Sequenced Distance Vector Routing) (Perkins & Bhagwat, 1994), Hu and Perrig have proposed the proactive routing protocol SEND (Secure Efficient Ad-hoc Distance vector) (Hu et al., 2003), which runs under a trusted ad-hoc network environment. In order to lower the node's CPU processing time and achieve better performance, SEND uses one-way public-key signed hash functions instead of asymmetric cryptography.

Authenticated Routing for Ad-hoc Network (ARAN) by Sanzgiri et al. (Sanzgiri et al., 2005) detects and protects the ad hoc network against malicious actions with help from its parties'

or peers' nodes by using pre-determined public key cryptography certificates. However, compared with SEND, ARAN requires a higher computational cost in each node to retain the hop-by-hop authentication.

Using a different approach, SRP (Secure Routing Protocol) (Papadimitratos & Haas, 2002) assures correct connectivity information as well as route discovery by rejecting fabricated, compromised or replayed route replies. SRP assumes a security association between the pair of end-points only, without the need for intermediate nodes to cryptographically validate control traffic (Sanzgiri et al., 2005); (Papadimitratos & Haas, 2002).

Those protocols and traditional security approaches, such as authentication, digital certificates and public-key encryption algorithm, still play important roles in achieving data privacy, integrity, non-repudiation and availability of communication in mobile ad-hoc networks (Zhou & Haas, 1999). However, these mechanisms by themselves are not sufficient, either in terms of computational or communication overhead or lack of ability to prevent attacks launched from inside the network. Therefore, there remains a need for a lightweight and reliable security enhancement protocol for mobile ad-hoc wireless network.

3. Routing and dynamic encryption protocol

3.1 Routing

In an ad-hoc wireless network, routing strategies can be classified as proactive or on-demand (reactive). With proactive protocols, such as Destination-Sequenced Distance Vector Routing (DSDV) (Perkins & Bhagwat, 1994) and Optimized Link State Routing Protocol (OLSR) (Clausen et al., 2003); (Clausen & Jacquet, 2003), the packets route information that is periodically exchanged among hosts, allowing each node to build a global routing table without considering the usage of routing information. In the on-demand approach, such as Ad-hoc Network On-demand Distance Vector (AODV) (Perkins & Royer, 1999) and Dynamic Source Routing (DSR) (Johnson et al., 2001), the nodes build and maintain routes as needed and only toward the nodes involved in the routing, instead of continuously calculating routes in the background.

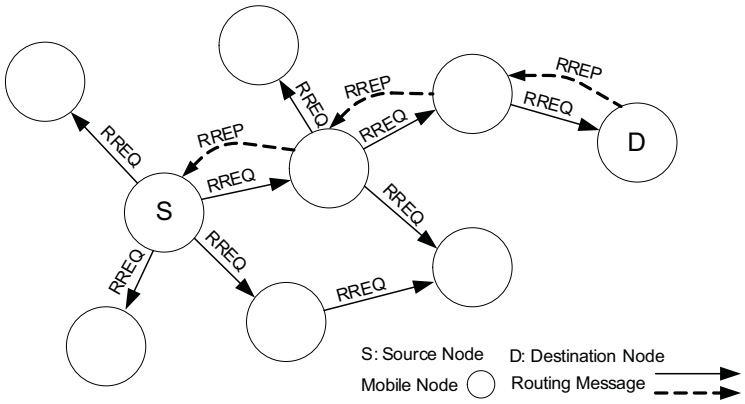


Fig. 1. AODV routing protocol with RREQ and RREP control message

AODV is adapted as the default routing protocol in this dynamic encryption model for the ad-hoc networking because of its high performance and low overhead, which are very important when considering that bandwidth is very limited in wireless communication. In AODV, as shown in Fig. 1. above, the source node first broadcasts a route request (RREQ) message to all adjacent nodes and waits for the corresponding route reply (RREP) message from the destination node to establish routing information. This request and reply query cycle will continue as long as this particular path is not listed in the routing table. Once routes have been built from source to destination, they will continue to be maintained as long as they are needed by the source node. All wireless packets between these two parties will follow the pre-build routing information and will be forwarded node by node until they reach the final destination. When the communication ends, the links will time out and eventually be removed from the table to release space for other routing paths.

3.2 i-key protocol procedures

This *i-key* protocol is primarily based on a dynamic re-keying mechanism that ensures the privacy of communication and prevents unauthorized users from accessing protected data over wireless communication. The key management and cipher stream system in *i-key* architecture is similar to Temporal Key Integrity Protocol (TKIP) used in WPA/ WPA2 and RC4 used in Wired Equivalent Privacy (WEP) (Lansford & Bahl, 2000), in which each encryption key contains a pre-shared key (PSK) and a randomly selected key value from the Initialization Vector (IV) pool. In addition to these two keys, an extra dynamic secret *i-key* is applied to the cipher stream that is used to encrypt every data packet before transmission. Fig. 2. illustrates the key stream that is combined with these three different keys and the block diagram of *i-key* encryption and decryption algorithm. The dynamic *i-key* is generated according to the previous data packet and therefore only the sender and authorized recipient are able to decrypt the cipher text by the key stream that is combined with the dynamic *i-key* and static key to reveal the plaintext in the data packet, which becomes the new seed of the *i-key* used in the next data encryption.

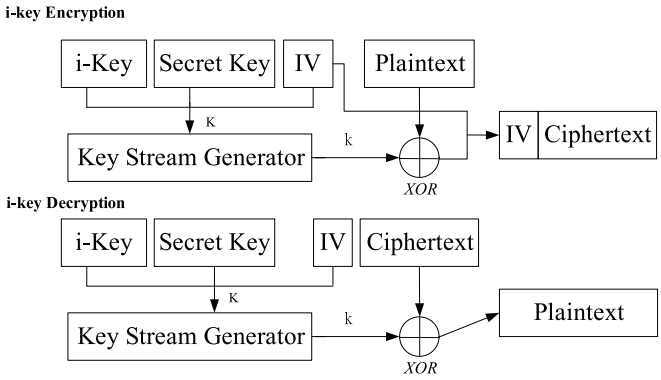


Fig. 2. Block diagram of *i-key* secure protocol

Once routing information and initial handshaking are established for communication between the source mobile node (SMN) and destination mobile node (DMN), the dynamic *i-key* encryption protocol for the wireless ad-hoc network will execute, as seen in Fig. 3.

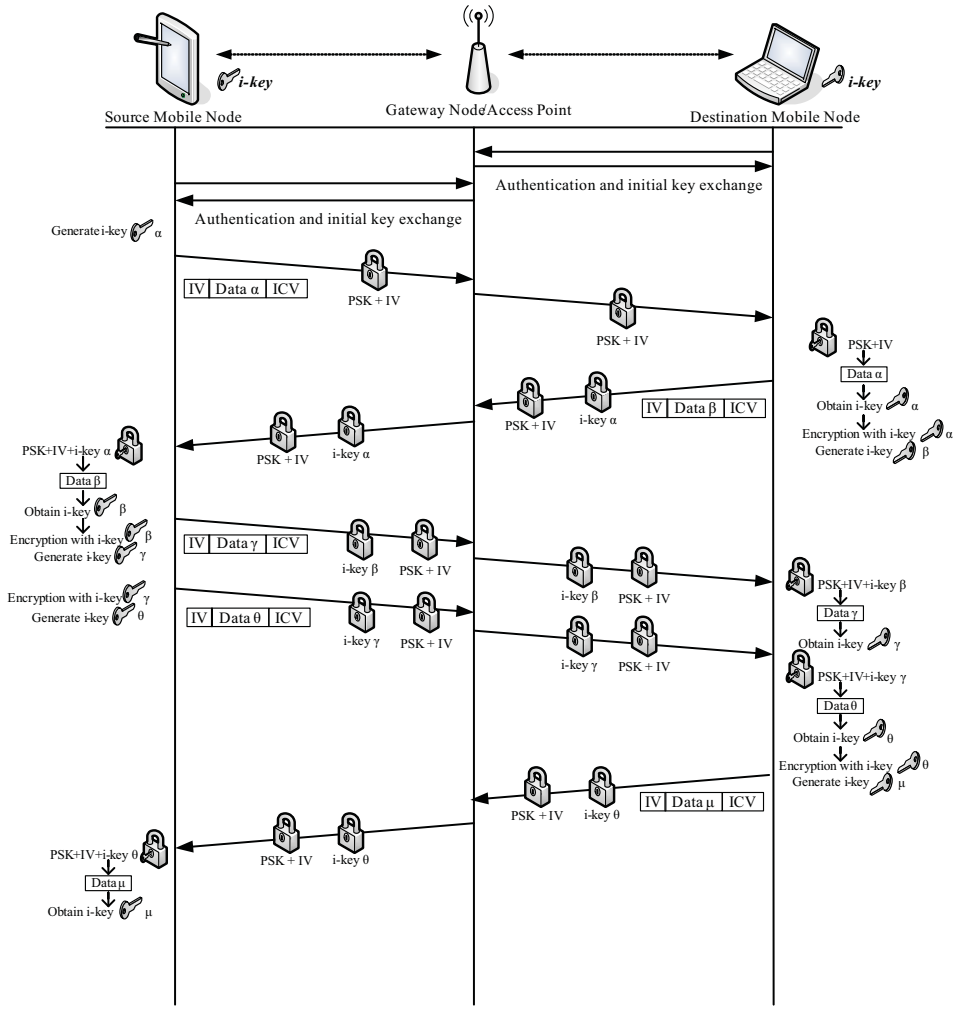


Fig. 3. Dynamic *i-key* encryption and decryption protocol procedures

Step 1. First, the source node S checks the destination node D on its routing information to confirm the proper routing rules been established. Then, source node S generates the secret *i-key*, which is based on the data as the seed contained on the first packet *a*, and keeps this particular secret key to decrypt the next encrypted packet from destination node D. A combination of pre-shared secret key *PSK* and one unique *IV* value is applied for the stream cipher to encrypt the plaintext before routing an adjacent mobile ad-hoc node to relay to the destination node D. Of all the communication between source node and destination node, this is the first and only packet that does not use the dynamic *i-key* for data encryption; however, the security protection remains strong since it needs at least two packets with the identical *IV* value to decode the pre-shard key. Each value in the *IV* pool is

generated randomly and uniquely to strengthen the encryption cipher stream and preventing people from cracking it even they are able to capture those wireless packets.

Step 2. The destination node D obtains the data packet *a* as well as the *i-key a* after running a decryption for this encrypted packet from source node S. It will then apply this dynamic *i-key a* to the next data packet's cipher stream to enhance security (because the source node S is the only one that has the same unique secret *i-key a* in this wireless ad-hoc network). Before sending the response/ reply packet β back to the source node by the same routing strategy, the destination node D will also generate the next *i-key* β based on data in the packet in order to decode the next arrival. From this point forward, every data packet and communication from one side to another in this wireless environment is secured by a dynamic stream cipher that has triple layers of protection: one pre-shared secret key *psk*, one unique *IV* and one dynamic *i-key* possessed only by the original source and destination node.

Step 3. The source node S will use the *i-key a*, generated in Step 1 and which it alone knows, to decode the cipher text along with the pre-shared secret key *psk* and *IV* to acquire the data β in the packet that it receives from destination node D. The encryption procedure with *i-key* in Step 2 will repeat again for the next data packet before node S sends it to the destination node D to enhance the security and maintain the data integrity from malicious modification.

Step 4. In cases when node S has more than one data packet to send before it gets a response, the destination node D will apply the corresponding *i-key* to decode the cipher text in accordance with the order of the arrival packets and update *i-key* based on the sequence number in each packet's header to ascertain that the decrypted cipher stream matches the arrival packet and thus passes the integrity checksum in the payload after decryption.

These *i-key* dynamic encryption/ decryption procedures will continue running and applying to every packet that is transmitted in the mobile ad-hoc wireless network to ensure the integrity and confidentiality of communication. When any wireless packet fails to be delivered to the destination or is lost during ad-hoc routing (which is common in both IEEE 802.1x based-oriented or an ad hoc network wireless network), an ACK-failed (timeout) or AODV routing error RRER message will be triggered and both sides will be alerted to restore the last successfully received data packet and then re-synchronize the dynamic *i-key* and start the communication over again from Step 2 for the next packet transmission.

Furthermore, before confidential data such as medical records or personal financial information are shared through a wireless ad-hoc network to other mobile devices, the source node can verify the authenticity of the destination node by requesting a response to decrypt a challenge message that the source node encrypted with the latest *i-key* holding with its signature. This sharing continues only when the other side passes the identity challenge; otherwise, the source node will mark the destination as invalid node and reject any further conversations to avoid data leaks or session hijacking. This verify-challenge mechanism in the *i-key* protocol can effectively detect any potential intruders and secure the wireless network by blocking both in-coming and out-going communication to prevent additional attacks.

In addition, this encryption protocol is highly flexible. The dynamic secret *i-key* is regenerated every time for each individual data packet; therefore, the secret key-size can also adjust dynamically to fit different needs in different applications. For example, an on-

line streaming system can temporarily increase the key size during the user identity authentication check to strengthen the complexity of ciphertext from eavesdropping by attackers and then lower the encryption/ decryption overhead by reducing the *i-key* size to improve the quality of services (QoS) of real-time live streaming while remaining under solid data protection. Thus, systems with existing security protection, such as SEND and SPR (Hu et al., 2003); (Papadimitratos & Haas, 2002) can still adopt this *i-key* encryption system to enhance data privacy and prevent malicious attacks against the wireless network.

3.3 i-key protocol algorithm

In additional to the RC4 encryption algorithm (Rivest Cipher 4, also know as ARC4 or ARCFOUR) (Rivest, 1992) that also used in WEP and TKIP protocol in IEEE 802.11 wireless networks, dynamic *i-key* protocol also utilizes the stream cipher as the security system model due to its efficiency, reliability and simplicity. Stream cipher takes in one byte to from a stream every time and produces a corresponding but different byte as the output stream, as shown in Fig. 4.

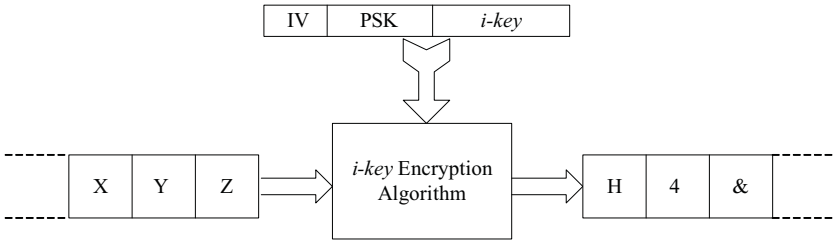


Fig. 4. Dynamic *i-key* encryption stream cipher

Then, this stream cipher combines with the data before transmission over the wireless network by using an exclusive OR (XOR - ⊕) operation. It combines two bytes, one from the cipher and one from the data, and generates a single byte output result as 0 when the values of them are equal, otherwise the result is 1. In general, the strength of an encryption algorithm is primarily measured by how hard it is to decode the ciphertext (Edney & Arbaugh, 2004). Certainly there are stronger encryption procedures than this RC4-like dynamic re-keying algorithm applied in this i-key architecture, however, this simple XOR encryption method is considered very strong among all of the data encryption people use today for both wired and wireless communication (Edney & Arbaugh, 2004).

One of the most important attributes of XOR operation is that if you apply the same value again to the first output result, the original value before the XOR operation is returned:

$$10110010 \oplus 11011001 = 01101011 \quad (1)$$

$$01101011 \oplus 11011001 = 10110010 \quad (2)$$

This characteristic can rewrite as:

$$\text{if } A \oplus B = C, \text{ then } C \oplus B = A \quad (3)$$

This is also how the decryption procedure works in the dynamic *i-key* system:

Encryption: plaintext stream cipher = ciphertext (4)

Decryption: ciphertext stream cipher = plaintext (5)

Compared with other encryption systems, such as AES and RSA, XOR operation is relatively resource friendly and lightweight, ideally suited for mobile and hand-held computing devices since they have limited hardware computing ability and power resources. The only remaining challenge is how to generate a strong cipher stream that ensures the quality of encryption to avoid key deciphering and that protects data integrity over wireless radio communication. Encryption algorithms used in this *i-key* protocol consist of a Key Scheduling Algorithm (KSA) that establishes an initial permutation *S-box* of *{0,1,2,.....,N-1}* of the numbers 0 to 255 from a random key array with the typical size of 40 to 256 bits and an Pseudo-Random Generation Algorithm (PRGA) that utilizes this output permutation *S-box* to generate the pseudo-random output sequence. The pseudocode for these two algorithms is shown in Fig. 5.

```
function KSA(k){
//initalization
  for i=0 to N-1{
    S[i]=i;
  }
  j=0;
//scrambling
  for i=0 to N-1{
    j= (j+S[i]+K[i mod keylength]) mod 256;
    swap(S[i],S[j]);
  }
}



function PRGA(k){
//initalization
  i=0;
  j=0;

//Generation output loop
  while Generationoutput:
  i=(i+1) mod 256;
  j=(j+S[i]) mod 256;
  swap(S[i],S[j]);
  k=(S[i]+S[j]) mod 256;
  r=S[k];
}
```

Fig. 5. Pseudocode of KSA and PRGA Algorithm

The KSA algorithm consists of two N loops of round operations that initialized the permutation array with a sequential number starting with 0 in the first loop and then rearranging the order by swapping each individual value with another byte in the same array with the following computational formula:

$$J(x) = \text{(the value the particular index byte of S-box + the value of the same particular index byte of K-box) with any overflow ignored} \quad (6)$$

The value of *J* is used as an index, as well as the values at that location, and are swapped with the target value in that location in *S-Box*. *Sn* is denoted as the result of the first "n" iterations from the loop of scrambling that represents the process have swapped each of *S[0]...S[n-1]*, with a corresponding value of *S[j]*. The same process will start from the beginning of the initial *S-box* and is continuously repeated until it finishes swapping until the end of the array and produces the final version of *S*, *S256* in our *i-key* system as the output permutation *S-box*.

Once the *S-box*, the so-called state array, is initialized, it will be used as input in the next phase of *i-key* encryption algorithm, called the PRGA. This involves more calculation and swapping to generate the final key stream. A Pseudo-Random Number Generator (PRNG) is an algorithm used to generate a random sequence of numbers, the elements of which are approximately independent. The PRGA in the *i-key* protocol is responsible for creating the cipher stream used to encrypt the plaintext based on the *S-box* value, whish is the output from the KSA in the previous step. It first initializes two indices, *i* and *j* to 0, and then loops over five operations that increase the value of *i* in each loop as the counter, increasing *j* pseudo-randomly by adding one value *S[i]* to it, then swapping the two values of the *S-box* pointed by the value of *i* and *j*, and outputs the values of the *S-box* that is pointed to by *S[i]+S[j]*. Note that every block of *S-box/State* array is swapped at least once, possibly with itself, within each completed iteration loop, and hence the permutation *S-box/State* array evolves fairly rapidly during the generation output loop phase (Fluhrer et al., 2001).

The strength of a cryptographic system primary depends on two components: the algorithm and the encryption key. Since a system is only as strong as its weakest link, both components need to be strong enough to protect the unsecure wireless communication via the radio frequency (Edney & Arbaugh, 2004); (Chandra, 2005). In this *i-key* encryption protocol, first of all, the dynamic re-keying algorithm enormously enhances the level of protection by adding the extra secret *i-key* to the *K-box*. This increases not only the complexity of the secret key array but also effectively prevents key cracking and dictionary attacks. Second, it improves the level of data protection by creating a better initialized *S-box/State* array during the KSA algorithm when swapping the blocks based on the *j* index that are mixed with the value of additional secret *i-key*. Finally, it helps generate a better and stronger pseudorandom number stream in the PRGA algorithm phase that is used to encrypt the data packet sent via the wireless network. Therefore, this dynamic *i-key* encryption protocol strengthens the cryptographic system in both ways and provides a solid protection for both individual stand-alone wireless models as well as for mobile ad-hoc wireless networks.

4. Security analysis

Due to the nature of frequent changes in both topology and membership in mobile ad-hoc networks, the initial design of the wireless routing protocol has mainly focused on the effectiveness of packet forwarding and delivery to the target node, and not on security. Consequently, a number of attacks that take advantage of this weakness have been developed for use against data integrity or routing protocol in wireless communication.

Transmitted data packets may be exposed to unauthorized access at anytime and anywhere due to the nature of radio broadcasting; therefore, it is essential to apply security protection

that prevents the reading or modification of confidential data by anyone who can receive the wireless signal. Using the secret key for data encryption is currently considered the most common way to protect data privacy in all kinds of computer communication; however, one of the static key or pre-shared key (psk) encryption's biggest vulnerabilities is that an attacker can obtain the original secret key by monitoring the packet transmission or conducting a massive dictionary attack between any two nodes in the network. Theoretically, a 64-bit secret key is decipherable with approximately 1 to 2 million data packets (2 to 4 million for 128-bit secret keys) and in a matter of mere hours, attackers can detect enough data packets in an average busy network environment to decode the pre-shared secret key (Chan et al., 2005).

In addition, mobile nodes are often deployed in a wide area with very limited or no physical protection, rendering them very vulnerable to capture or hijacking. Once a single node has been compromised and the secret key revealed, an attacker can launch far more damaging attacks from inside the network without being detected. Hence, the encryption protocol that applies to the mobile ad-hoc network should not only prevent the encryption key from been revealed, but also be flexible enough to be adopted as a security enhancement by other existing routing protocols in such highly dynamic network environment.

With the advanced dynamic encryption mechanism, *i-key* protocol ensures privacy of communication and protects sensitive data from eavesdropping by dynamically changing the secret *i-key*, which allows only the original sender and authorized receiver to decode the encrypted data packet via the secret *i-key* that they own. Therefore, this protocol overcomes the weakness of pre-shared key encryption and protects the wireless network against other attacks in the methods described below.

4.1 WarDriving

WarDriving is the act of scanning and searching for wireless network signals in a moving vehicle by any devices equipped with a wireless interface, such as PDAs or portable computers. Scanning software likes NetStumbler and Airmon-ng can report detailed information, including Service Set Identifier (SSID), MAC address, communication channel, signal strength and most importantly, the encryption protocol applied for each access point and wireless node. It can also record the location by connected to a GPS (Global Position System) receiver.

In addition, there are several online web sites and databases such as WiGLE/ JiGLE, StumbVerter and Google Hotspot Maps where people around the world can report their discovery of each access point's information. In July 2010, WiGLE/ JiGLE alone recorded 23,182,272 pieces of access point data from 1,125,930,947 unique observations, which cover most of the major cities on five continents. Therefore, other people who do not have the proper equipment for doing wardriving can simply locate any near by access point by searching these sites. As an example, take the city of College Station, where Texas A&M University is located. More than six thousand access points have been detected and reported to the WiGLE/ JiGLE database. Fig. 6. demonstrates the distribution in a Google map.

Those scanning tools, access point information sources and online databases are convenient for wireless network studies and research, but they also provide an advantage by letting hackers pick the most vulnerable entry point from an existing wireless network and expected to spend less time and effort to compromise the target node and its local area network. That is also why running a wardriving scan is usually hackers' first step before they start any other kind of wireless attack.

Fig. 6. The distribution of wireless access points in city of College Station, Texas

The dynamic *i-key* encryption protocol can recognize and prohibit wardriving attacks by adding wireless packet pattern analysis to both access point and mobile node. Take NetStumbler for example; this unique pattern can be found in its 802.11 probe request frames (Tsakountakis, 2007). First, LLC encapsulated frames generated by NetStumbler contain the valise 0x00601d for organizationally unique identifier (OID) and protocol identified (PID) of 0x0001. Second, the payload data size is usually 58 bytes with the attached hidden string "Flurble gronk bloopit, bnip Furndletrune!" for version 3.2.0, "All your 802.11b are belong to us" for version 3.2.3 and " intentionally left blank 1" for version 3.3.0. In (Tsakountakis, 2007), authors also illustrate the pseudocode for the above pattern detection in a traditional wireless network and we extended this for dynamic *i-key* protocol used in a mobile ad-hoc wireless network (Fig. 7.). Once the *i-key* system detects the presence of wardriving activities, it generates several false probe requests to prevent any further attacks by misleading attackers with fake MAC address, SSID, channel and encryption protocol. Similar detecting signature parameters and policies shown in Fig. 8 can also add to the intrusion detecting system (IDS) to prevent additional attack on a wireless network.

4.2 Man-in-the-Middle (MITM)

In a Man-in-the-Middle (MITM) attack, as shown in Fig. 9., the hacker places himself in the mid-point of the information flow between sender and recipient, which allows him to access all of the communication between them. If no proper security protection and data encryption protocol are applied to the wireless network, the attacker can effortlessly read the data, inject malicious packets, modify the information integrity or even block the communication from one side to another. In addition, a man-in-the-middle attack is hard to detect and prevent in a wireless network environment since everyone can easily capture the wireless packets transmitted from any mobile device to another or from the base stations.

```
function Detect_Netstumbler{
      sniff for 802.1x wireless packets
      parse into frames and abstract MAC header
      check 802.1x wireless frame type
      if (frame_type == "prob request frame"
      && wlan.fc.type_subtype == "802.1x beacon" (0x08)
      && llc.oui == 0x00601d (netstumbler)
      && llc.pic == 0x0001  (netstumbler)){
        switch (data[4:4]){ //in ASCII code format
          case "466c7572" : NetStumbler detected, version 3.2.0
          case "416C6C20" : NetStumbler detected, version 3.2.3
          case "20202020" : NetStumbler detected, version 3.3.0
          default : NetStumbler not detected
        }
      }
      if(NetStumbler detected){
        log frame and packet content
        reply false probe response frames
        send notice to gateway node and access point to prevent further attack
        repeat function if needed
      }else{ //not detected
        repeat function if needed
      }
  }
```

Fig. 7. NetStumbler detecting pseudocode

```
Name = "NetStumbler 3.2.0", Ver = 0, Preced= 13, FrmType = data, Pattern =
0:0x0108:0x03FF, Pattern = 27:0x00601d:0xFFFFFF, Pattern = 30:0x0001:0xFFFF, Pattern =
36:0x466c7572:0xFFFFFFFF, Freq = 1, Quiet = 300, Action = report, Desc="NetStumbler 3.2.0"

Name = "NetStumbler 3.2.3", Ver = 0, Preced= 14, FrmType = data, Pattern =
0:0x0108:0x03FF, Pattern = 27:0x00601d:0xFFFFFF, Pattern = 30:0x0001:0xFFFF, Pattern =
36:0x416C6C20:0xFFFFFFFF, Freq = 1, Quiet = 600, Action = report, Desc="NetStumbler 3.2.3"

Name = "NetStumbler 3.3.0", Ver = 0, Preced= 15, FrmType = data, Pattern =
0:0x0108:0x03FF, Pattern = 27:0x00601d:0xFFFFFF, Pattern = 30:0x0001:0xFFFF, Pattern =
36:0x20202020:0xFFFFFFFF, Freq = 1, Quiet = 600, Action = report, Desc="NetStumbler 3.3.0"

Name = "NetStumbler generic", Ver = 0, Preced= 16, FrmType = data, Pattern =
0:0x0108:0x03FF, Pattern = 27:0x00601d:0xFFFFFF, Pattern = 30:0x0001:0xFFFF, Freq = 1,
Quiet = 600, Action = report, Desc="NetStumbler"
```

Fig. 8. NetStumbler signature parameters for CISCO IDS

There are many different ways to interrupt the communication and allow hackers to insert themselves in the middle of the information flow by taking advantage of the protocol's weak security design, for example, by using Address Resolution Protocol (ARP) spoofing (Plummer, 1982); (Wagner, 2001), Domain Name Server (DNS) spoofing (Klein, 2007); (Sax, 2000) or via Border Gateway Protocol (BGP) (Rekhter et al., 2003). Once hackers are able to access the communication channel, the next step is to capture the current session, decode the secret key, decrypt the message and then modify the content and send it back. First, the attacker needs to reveal the secret key before he can successfully alter any data packets and launch an attack on both sender and recipient.

However, due to the natural of this dynamic re-keying protocol, every single packet is secured by a unique and solid cipher stream composed of one hidden pre-shared secret key (psk), one unique IV value and one dynamic *i-key*, which together provide three strong layers of secure enhancement protection for wireless ad-hoc networks. Plaintext messages can only be decoded

by authorized recipients and senders who have the legal and updated *i-key*. Therefore, a real-time man-in-the-middle attack would not succeed against this protocol.

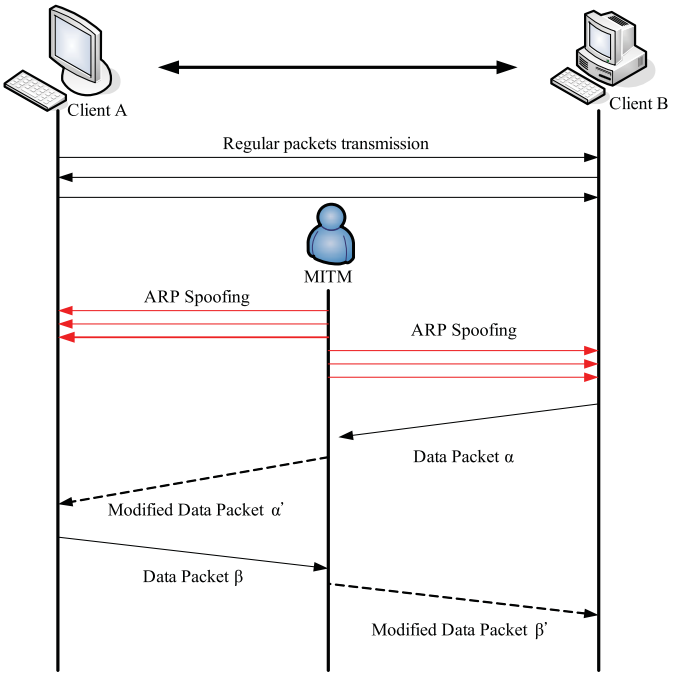


Fig. 9. Wireless man-in-the-middle attack example

4.3 Blackhole attacks

Blackhole attacks (Tamilselvan & Sankaranarayanan, 2008); (Hu & Perrig, 2004); (Chuah & Yang, 2006) (Fig. 10.) are similar to denial of services (DoS) attacks in traditional networks in that a compromised node in MANET participates in a routing protocol and attracts all packets by claiming to have a valid route to all destination nodes, but then drops all received data packets without forwarding them. This attack will not merely prolong the routing delay; in the worst case scenario, it can disrupt the entire network connection.

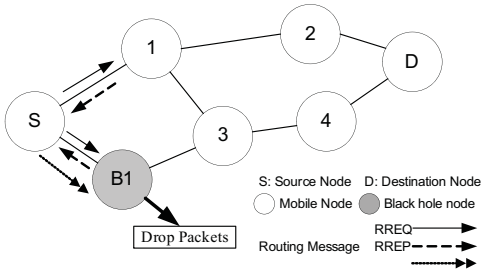


Fig. 10. Black hole attack in MANET

This attack is easily lunched against reactive protocols in a Mobile Ad-Hoc Network such as Dynamic Source Routing (DSR) (Johnson et al., 2001), Temporally Ordered Routing Algorithm (TORA) (V. D. Park & Corson, 1997) and Ad Hoc On-Demand Distance Vector (AODV) (Perkins & Royer, 1999), which assume that all nodes in a given ad-hoc network are trustworthy and that the data packet will forward to the node that first replies to the route reply message (RRM) in routing path discovery. To set in motion a blackhole attack, the attacker needs to decipher not only the pre-shared key (psk) but also the dynamic re-keying secret *i-key*; however, the attacker needs the added advantage of a dynamic re-keying mechanism that provides three different layers of data encryption and unique cipher streams to secure both the data and each mobile host's secret key for every transmitted packet over the mobile ad-hoc wireless network. The *i-key* encryption protocol can easily prevent this form of attack in its very early stages by stopping the node from compromised and controlled by the attacker.

4.4 Wormhole attacks

In wormhole attacks, an adversary establishes a wormhole link by using either in-band or out-of-band communication as illustrated in Fig. 11. This direct link can be set up with a traditional wire, long-range wireless transmission or an optical link. Once this wormhole link is built up, the attacker can receive wireless packets on one end in the network, known as the original point, and then reply to them in a timely fashion at another location, as the destination point.

Using this method, an attacker could relay an authentication exchange to gain unauthorized access without compromising any node or having any knowledge of the routing protocol in use (Chuah & Yang, 2006); (Eriksson et al., 2006). Because a wormhole attack is launched internally against the mobile ad-hoc network, default routing protocols and traditional security protections are unable to effectively detect or prevent this unique attack pattern.

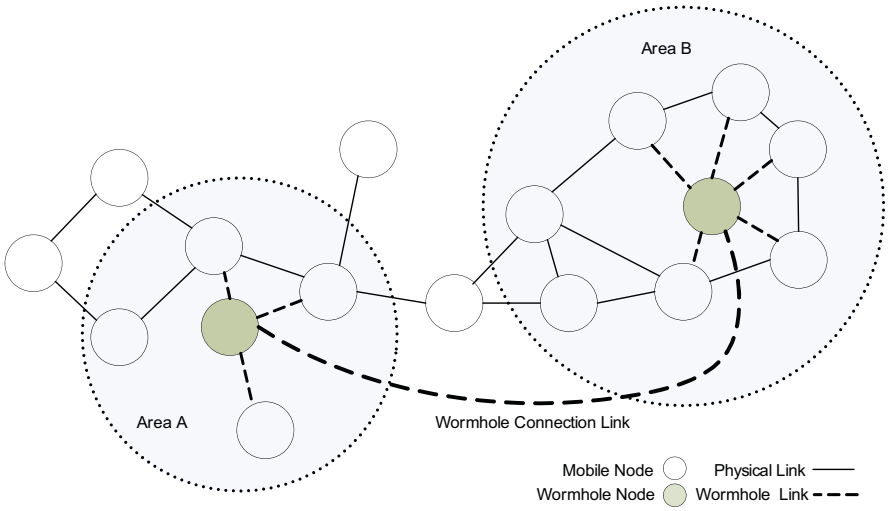


Fig. 11. Wireless wormhole attack

Under the protection of the *i-key* encryption protocol, however, only the original sender and authorized receiver are able to decrypt the cipher text, by using the unique secret key in their possession, ensuring continued confidentiality and integrity for the data communication, as well as the authentication information between source and destination node. Therefore, even if wormhole attacks are launched inside the network, the cryptographic key that is used for both encryption and decryption during each node-to-node communication still remains secret and the authentication information is still valid only to original node as well.

4.5 Session hijacking

In session hijacking, attackers take an authorized and authenticated session away from its owner and use it to establish a valid connection with the peer node, then snoop or modify the secret data. To successfully execute session hijacking, the attacker must accomplish two tasks: He first needs to stop the target node from continuing the session and then disguise himself as one of the legal client nodes (Welch & Lathrop, 2003).

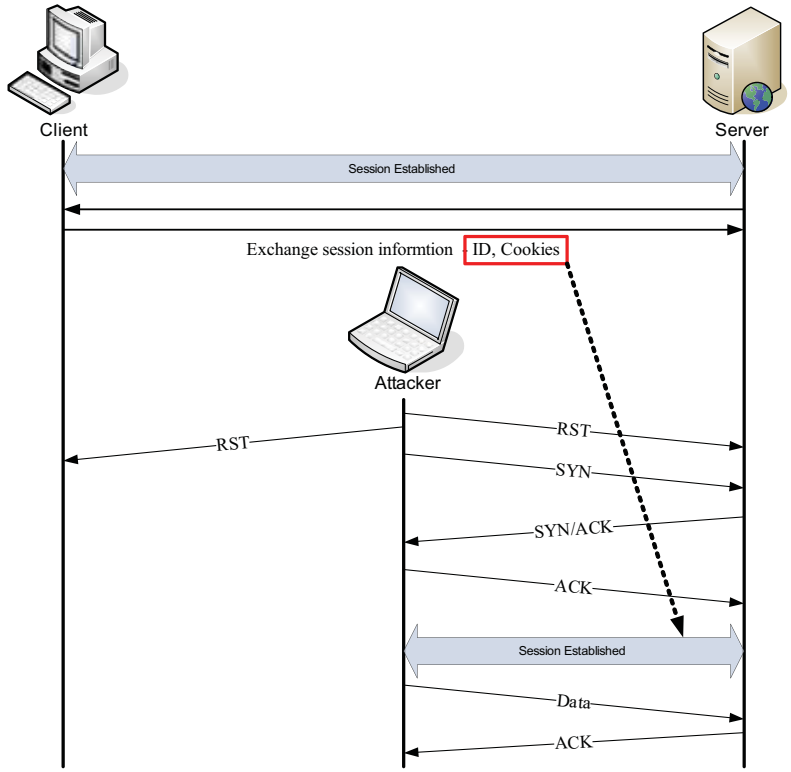


Fig. 12. Session hijacking attack example in IEEE 802.11 wireless network

The attacker can take the advantage of using Denial of Services (DoS) or a flood attack to achieve his first task for the session hijacking to temporarily interrupt the target's session connection; however, in order to masquerade himself as the target, he also needs to obtain

the original secret key to maintain communication with the peer node. Because the *i-key* is dynamic re-keys for every packet, the secure key stream remains secret even if the session connection is interrupted. In this protocol design, described in the previous chapter, when communication is stopped or interrupted, the two parties will be notified by an IEEE 802.11 ACK-failed (timeout) or AODV routing error RRER message to restore the last successfully received data packet and the secret *i-key*. Therefore the security protection remains even when consistency session connections are lost.

4.6 Key cracking and dictionary attacks

Any encryption system using only static pre-shared key (psk) or lacking well-defined re-keying mechanisms are vulnerable to key cracking through the capturing of sufficient packets. Also, when choosing passwords for authentication or encryption system, many users select from a small domain and end up with a weak password. Those weak security systems and passwords enable adversaries to launch dictionary attacks that attempt to login into accounts by trying all possible password combinations. Once the correct password is discovered, attackers can crack the ciphertext easily and even carry out other attacks effortlessly (Pinkas & Sander, 2002). Fig. 13. below illustrates the key cracking attack with Aircrack-ng software.

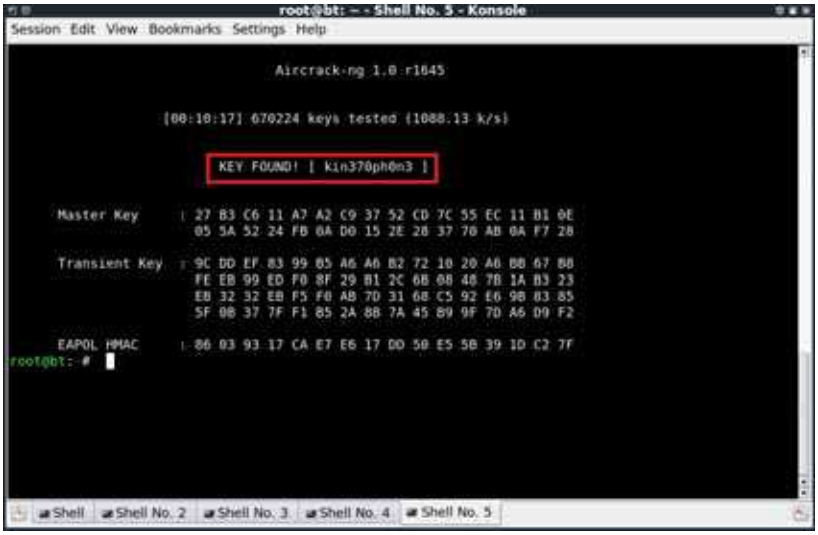


Fig. 13. Key cracking by Aircrack-ng

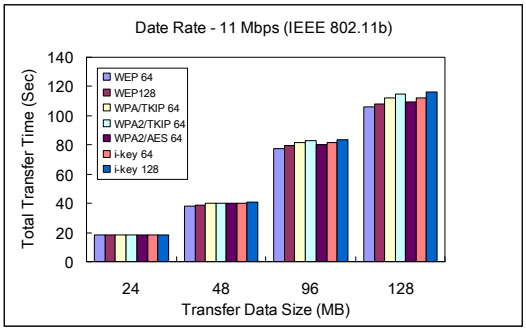
Dynamic re-keying in the manner used in *i-key* protocol is advantageous because not only is every stream cipher unique for each packet, but also the *i-key* system provides the wireless ad-hoc network with an innovative and solid security protocol of up to 18,432 bits, the maximum for the data packet size in IEEE 802.1x wireless communication (Borsc & Shinde, 2005), in key size. Therefore, attackers are unlikely to take the time required to capture enough packets before they can start to crack them or launch dictionary attacks against the system, because they know the longer they stay, the more likely their detection by a monitor system or firewall will be.

5. Performance evaluation

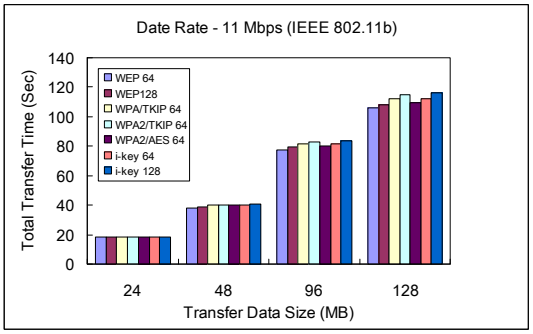
In these experiments, both 25 and 50 mobile nodes with 2 access points randomly located over an area of 600m x 600m and 1100m x 1100m are simulated with different settings of the size of the secret *i-key* that correspond to other security protocols. Each simulation ran for 200 simulated seconds with a radio transmission range set to 250 meters. Nodes coved by this range can receive the wireless signal and establish communication directly to the nodes within its ad-hoc range, while others rely on packets relayed by adjacent mobile nodes to deliver the message to the destination node. The physical and MAC layer setting is following the standard of IEEE 802.11 protocol with the data rate set from 1 to 20 MB/ s.
The kernel of this test bed is based on Fig. 3. and Fig. 5. for the *i-key* dynamic encryption protocol with the rewrite extension from CMU Monarch (Monarch Project, 1998) to support this dynamic re-keying architecture model for AODV routing in mobile ad-hoc network.

5.1 Protocol throughput

In the throughput experiment, two mobile nodes are randomly selected in the deployed area and measured the average of total complete time for four different sizes of data transferred between them. This protocol throughput test allowed us easily to compare the performance of *i-key* with WEP, WPA and WPA2 system, which are the most popular and adopted security protocols in today's wireless networking. As seen in Fig. 14, there is almost no



(a) 25 mobile nodes over 600mx600m area



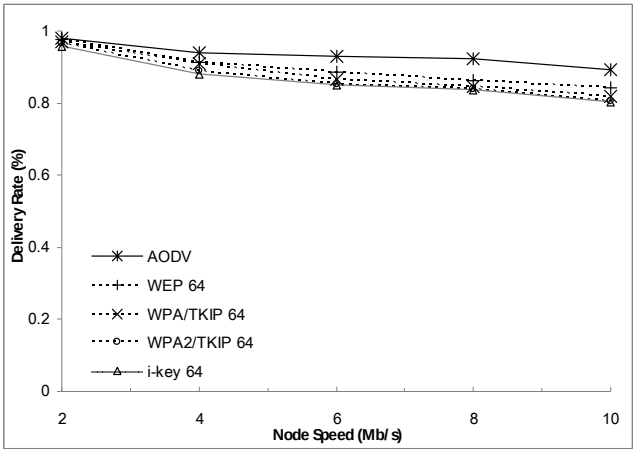
(b) 50 mobile nodes over 1100mx1100m area

Fig. 14. Average total data transfer time for *i-key* encryption protocol

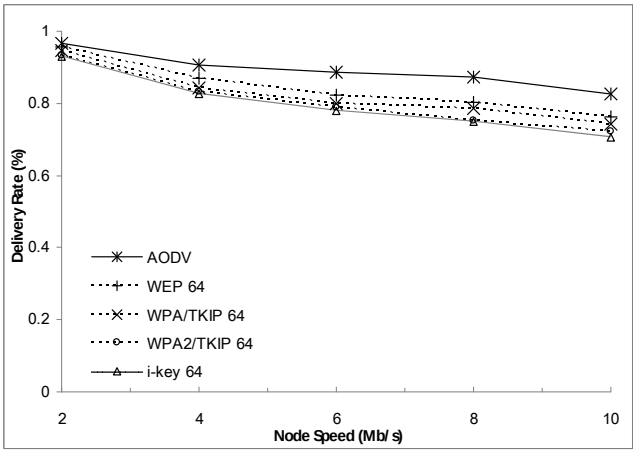
difference between each encryption approach in the lower transfer data size (24 and 48 MBs) and only a very small gap from the quickest WEP protocol with 64 bits to the slowest dynamic *i-key* 128 bits security system while transferred over 96 MBs of data. However, regarding data security, *i-key* encryption protocol not only strengthened the cipher by doubling the secret key size to provide a higher level of protection, but also dynamically re-keying during the end-to-end communication to defend the network from unwanted intrusion and guarantee the privacy of wireless data exchange.

5.2 Protocol delivery rate

The simulation results for protocol average delivery rate are shown in Fig. 15. The percentage of successfully delivered packets is measured from the source to the destination



(a) 25 mobile nodes over 600mx600m area



(b) 50 mobile nodes over 1100mx1100m area

Fig. 15. Average end-to-end delay for AODV and *i-key* protocol

node in five different data rate setting: 2, 4, 6, 8 and 10 MB/ s. As expected, delivery rates dropped as the result of a greater number of lost packets and collisions in the wireless environment caused by the increased number of mobile nodes and data transfer speed. The nature of radio communication makes packet loss and collisions during transmission unavoidable. When this happens to the *i-key* dynamic encryption protocol, it only needs to retrieve the secret key from the most recently received data packet and then re-synchronize with both sides to continue the conversation. Consequently, the cost of time and overhead for packet loss and collision in the *i-key* protocol is quite low. This also is why the differences between *i-key* with other secure protocols are minimal.

Both the complexity of the encryption system and the size of the ad-hoc network have a negative effect on performance. Obviously, AODV alone had the best delivery rate in all of the simulations, a result of the trade-off between security and performance. However, the relatively small gap between them also underscores that this *i-key* protocol can perform as efficiently as a non-security protection such as an AODV routing protocol while providing stronger data privacy through the dynamic *i-key* encryption system.

Those results from throughput and end-to-end delay experiments also indicate that the *i-key* security mechanism has very low computational overhead and power consumption during both data encryption and decryption procedure, which is very critical, especially when most mobile nodes in the wireless network depend on limited processing ability and the finite energy provided by batteries (Wang & Chuang, 2004).

6. Conclusion and future research

Data integrity and privacy are the two most important security requirements in wireless communication today. Most mechanisms rely on pre-share key (psk) data encryption to prevent unauthorized users from accessing confidential information. However, maintaining security in the highly dynamic ad-hoc wireless network is full of challenges due to the complexity of data routing and the nature of the wireless transmission medium.

In this chapter, we introduced a novel, efficient and lightweight encryption protocol that fulfils the need for security protection in wireless ad-hoc networks. This protocol ensures the privacy of communication from node to node and prohibits the modification of sensitive data by dynamically changing the secret key for data encryption during packet transmission. Under the protection of this protocol, only the original sender and authorized recipient are able to decode the cipher text using the secret key that is in their possession only. Therefore, the weakness of pre-shared key encryption is overcome and other wireless attacks are prevented. Experiment results with different network configurations and key sizes have been simulated. They indicate that this *i-key* protocol design is efficient, with low commutation overhead, while providing better and stronger data protection compared with other common security protocols in IEEE 802.11 wireless network. Furthermore, the dynamic encryption and decryption architecture in *i-key* protocol is flexible; other secure systems can also adopt it as a secondary security enhancement without compromising system performance.

The future works include the integration of this existing work with the intrusion detection and locating system. This integration provides another layer of defense by effectively pinpointing the location of an attacker and helps the wireless secure system to react correctly and instantly. Also, the implementation of advanced dynamic secure protection for large-scale wireless communication, such as IEEE 802.16 WiMAX network and the 4G (4th

generation) of the cellular wireless network is also recommended, with evaluation of protocol performance in both lab software simulations and real-world experiments.

7. References

Borsc, M., & Shinde, H. (2005). Wireless security & privacy. In *2005 IEEE International Conference on Personal Wireless Communications, 2005. ICPWC 2005*, pp. 424-428, 2005

Chan, F., Ang Hee Hoon, & Issac, B. (2005). Analysis of IEEE 802.11b wireless security for university wireless LAN design. In *Networks, 2005.* doi:10.1109/ ICON.2005.1635688

Chandra, P. (2005). *Bulletproof wireless security: GSM, UMTS, 802.11 and ad hoc security.* Elsevier, 0750677465

Chuah, M., & Yang, P. (2006). Comparison of Two Intrusion Detection Schemes for Sparsely Connected Ad Hoc Networks. In *Military Communications Conference, 2006. MILCOM '06*, pp. 1–7, 2006

Clausen, T., & Jacquet, P. (2003). RFC3626: Optimized Link State Routing Protocol (OLSR). *RFC Editor United States*.

Clausen, T., Jacquet, P., Adjih, C., Laouiti, A., Minet, P., Muhlethaler, P., Qayyum, A., et al. (2003). Optimized link state routing protocol (OLSR).

Edney, J., & Arbaugh, W. A. (2004). *Real 802.11 security: Wi-Fi protected access and 802.11 i*. Addison Wesley Publishing Company, 0321136209

Eriksson, J., Krishnamurthy, S. V., & Faloutsos, M. (2006). Truelink: A practical countermeasure to the wormhole attack in wireless networks. In *Proceedings of the 2006 14th IEEE International Conference on Network Protocols, 2006. ICNP'06*, pp. 75–84, 2006

Fluhrer, S., Mantin, I., & Shamir, A. (2001). Weaknesses in the Key Scheduling Algorithm of RC4. In *Selected Areas in Cryptography*, pp. 1-24

Gast, M. (2002). Wireless LAN security: A short history. Available online at: http:/ / www.oreillynet.com/ pub/ a/ wireless/ 2002/ 04/ 19/ security.html

Hu, Y. C., Johnson, D. B., & Perrig, A. (2003). SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. *Ad Hoc Networks*, *1*(1), pp. 175–192

Hu, Y. C., & Perrig, A. (2004). A survey of secure wireless ad hoc routing. *IEEE Security and Privacy magazine*, *2*, pp. 28–39.

Hu, Y. C., Perrig, A., & Johnson, D. B. (2005). Ariadne: A secure on-demand routing protocol for ad hoc networks. *Wireless Networks*, *11*(1), pp. 21–38.

Hubaux, J. P., Buttyán, L., & Capkun, S. (2001). The quest for security in mobile ad hoc networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*

Johnson, D. B., Maltz, D. A., Broch, J., & others. (2001). DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad hoc networking*, *5*, pp. 139–172

Johnson, D. B., Maltz, D. A., Hu, Y. C., & Jetcheva, J. G. (2002). *The dynamic source routing protocol for mobile ad hoc networks*. Internet-Draft

Kant, L., Demers, S., Gopalakrishnan, P., Chadha, R., LaVergne, L., & Newman, S. (2005). Performance modeling and analysis of a mobile ad hoc network management system. In *MILCOM*, Vol. 5

Klein, A. (2007). *BIND 9 DNS cache poisoning*. Available online at http:/ / citeseerx.ist.psu.edu/ viewdoc/ download?doi=10.1.1.86.4474&rep=rep1&type=pdf

Lansford, J, & Bahl, P. (2000). The design and implementation of HomeRF: A radio frequency wireless networking standard for the connected home. *Proceedings of the IEEE*, *88*(10).

Liu, X., Fang, Z., & Shi, L. (2007). Securing Vehicular Ad Hoc Networks. In *2nd International Conference on Pervasive Computing and Applications, 2007. ICPCA '07*, pp. 424-429

Miller, S. K. (2001). Facing the challenge of wireless security. *Computer*, *34*(7), pp. 16–18

Monarch Project (1998). Rice Monarch Project and Wireless Mobility Extension to ns-2

Papadimitratos, P., & Haas, Z. J. (2002). Secure routing for mobile ad hoc networks. In *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, Vol. 31

Park, J. S., & Dicoi, D. (2003). WLAN security: current and future. *IEEE Internet Computing*, *7*(5), pp. 60–65.

Park, V. D., & Corson, M. S. (1997). A highly adaptive distributed routing algorithm for mobile wireless networks. In *IEEE Infocom*, Vol. 3, pp. 1405-1413

Perkins, C. E., & Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the conference on Communications architectures, protocols and applications*

Perkins, C. E., & Royer, E. M. (1999). Ad-hoc on-demand distance vector routing. In *2nd IEEE Workshop on Mobile Computing Systems and Applications,*New Orleans, LA.

Pinkas, B., & Sander, T. (2002). Securing passwords against dictionary attacks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 161-170

Plummer, D. C. (1982). RFC-826 An Ethernet Address Resolution Protocol. *Network Working Group*.

Prasithsangaree, P., & Krishnamurthy, P. (2004). On a framework for energy-efficient security protocols in wireless networks. *Computer Communications*, *27*(17), pp.1716–1729.

Ramakrishnan, K., Balasubramanian, A., Mishra, S., & Sridhar, R. (n.d.). Wireless Security Protocol using a Low Cost Pseudo Random Number Generator, 2005.

Rekhter, Y., Li, T., Hares, S., & others. (2003). *RFC-1771 A border gateway protocol 4 (BGP-4)*. RFC 1771, March 1995.

Rivest, R. L. (1992). The RC4 Encryption Algorithm. RSA Data Security. *Inc., March*, *12*.

Sanzgiri, K., LaFlamme, D., Dahill, B., Levine, B. N., Shields, C., & Belding-Royer, E. M. (2005). Authenticated routing for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, *23*(3), pp. 598–610.

Sax, D. (2000). DNS spoofing (malicious cache poisoning). *November*, *12*. Available online at http:/ / www. sans. org/ rr/ firewall/ DNS_spoof. php

Tamilselvan, L., & Sankaranarayanan, V. (2008). Prevention of co-operative black hole attack in MANET. *Journal of networks*, *3*(5), 13.

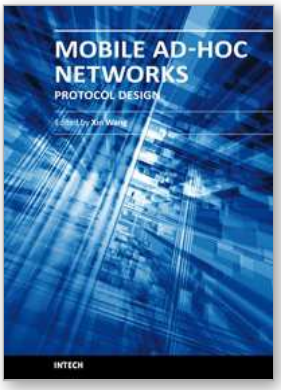
Tsakountakis, A., Kambourakis, G., & Gritzalis, S. (2007). Towards effective Wireless Intrusion Detection in IEEE 802.11i. *Third International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, 2007

Wagner, R. (2001). Address resolution protocol spoofing and man-in-the-middle attacks. *The SANS Institute*.

Wang, Y. H., & Chuang, C. C. (2004). Ad hoc on-demand backup node setup routing protocol. *Journal of Information Science and Engineering*, *20*(5), pp. 821–843.

Welch, D., & Lathrop, S. (2003). Wireless security threat taxonomy. In *Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society*, pp. 76–83

Zhou, L., & Haas, Z. J. (1999). Securing ad hoc networks. *IEEE network*, *13*(6), pp. 24–30.



Mobile Ad-Hoc Networks: Protocol Design

Edited by Prof. Xin Wang

ISBN 978-953-307-402-3
Hard cover, 656 pages
Publisher InTech
Published online 30, January, 2011
Published in print edition January, 2011

Being infrastructure-less and without central administration control, wireless ad-hoc networking is playing a more and more important role in extending the coverage of traditional wireless infrastructure (cellular networks, wireless LAN, etc). This book includes state-of-the-art techniques and solutions for wireless ad-hoc networks. It focuses on the following topics in ad-hoc networks: quality-of-service and video communication, routing protocol and cross-layer design. A few interesting problems about security and delay-tolerant networks are also discussed. This book is targeted to provide network engineers and researchers with design guidelines for large scale wireless ad hoc networks.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821