Key-Exchange in Peer-to-Peer Radio Networks

Felix K. Akorli
National University of Rwanda
Butare, Rwanda

Dawoud D.S.
University of KwaZulu Natal
Durban, South Africa

Johan Van Der Merwe
University of KwaZulu Natal
Durban, South Africa

Abstract

The paper presents some simple techniques for key establishment over a radio link in pure peer-to-peer network. In such networks, the users do not share any authenticated information in advance and the Man-In-The-Middle (MITM) attack is considered feasible. The proposed approach includes two folds; the first is the selection of a secure key exchange protocol, the second fold is to find ways that guarantee the integrity of the messages that are exchanged between the devices during the phase of establishing the key. To implement the first goal, the paper uses a technique based on a variant of the Diffie-Hellman key agreement protocol. The proposed variant is an appropriate mechanism to avoid MITM and allows mutual authentication. It is also possible to use the scheme to establish shared key between more than two entities. To achieve the second goal the paper proposes the use of a modified I-code that can guarantee the integrity of the messages exchanged while establishing the key. The analyses verified the security of the proposed protocol, and the simulation showed that the modified I-code together with using on-off keying and signal anti-blocking guarantee the integrity of the messages.

1. Introduction

As the popularity of mobile systems such as PDAs, laptops, mobile phones and pocket computers increases every day, users tend to rely more on them in a growing number of situations. It is very common nowadays to get two or more people together (e.g., at a meeting, in a restaurant or a pop) to make use of their devices to communicate with each other. Clearly, the communication between these devices must be properly secured.

In most cases the users want the security between their devices to be peer-to-peer oriented, thus operating independently of any authority. In practice, this means that the mobile devices must run a protocol to authenticate each other and to protect the confidentiality and integrity of the data they exchange. Data protection requires using some form of encryption/decryption and other cryptographic mechanisms. Peer-to-peer communication typically, requires setting up a symmetric shared key. This key can be used to secure immediate communications as well as communications that take place afterwards (e.g., when users exchange e-mail over the Internet). Irrespective of the encryption/decryption techniques used, the two devices have to exchange a number of messages in order to establish the symmetric key. The security of the system starts from this phase; a complete integrity and protection of the messages exchanged between the devices during the phase of establishing the key must be guaranteed.

Although most, if not all, security threats against a wired network are equally applicable to wireless network, the latter possesses a number of additional vulnerabilities that make it more challenging to secure (specially with mobile ad hoc networks):

Open wireless access medium: The security threats of message, eavesdropping and injection, are universal in any network; however, they are particularly severe in wireless networks due to the open and the shared medium.

With off-the-shelf hardware and little effort, an attacker can intercept and inject traffic through a wireless channel. There is no physical barrier separating the attacker from the network, as it is in the case of wired networks.

Limited bandwidth: Wireless networks are particularly vulnerable to Denial-of-Service (DoS) attacks due to their limited bandwidth and in-band signaling. Although the wireless channel capacity is continuously increasing, the spatial contention problem poses a fundamental limitation on the network capacity. One can deploy redundant fibers, but everyone must share the same wireless spectrum.

System Complexity: Generally speaking, wireless networks are far more complex than the wired. This is due to the special needs for mobility support and efficient channel utilization. Each piece of complexity added onto the system can introduce potential security vulnerabilities, especially, in systems with a large user population and a complex infrastructure, such as third–generation (3G) networks.

Recent years have witnessed an influx of efforts of research works on wireless security, with the outcome of a rich body of proposed solutions. Many of the existing literatures cover the security of the three popular wireless networking paradigms: namely, wireless LANs (WLANs) (e.g., [1] - [3]), 3G cellular networks [4], [5], and ad hoc networks [6].

In this paper, we discuss the problem of key agreement (and mutual authentication) in settings where the users do not share any authenticated information in advance. It is our goal to devise mechanisms that prevent the attacker from modifying the exchanging key material without being noticed. We are using a variant of Diffie-Hellman for key exchange together with using modified I-codes.

I-codes is a new security primitive that enables integrity protection of the messages that are exchanged between entities and which, do not hold any shared secrets or mutual authentication material (i.e. public keys or shared secret keys). The construction of I-codes enables a sender to encode any message, such that if its integrity is violated in transmission, the receiver is able to detect it.

In the literature such codes are also known as ***All-Unidirectional Error-Detecting*** (***AUED***) codes. They are used in situations where it is possible to change a bit “0” into a bit “1” but the reverse is not possible (except with a negligible probability) [7, 8]. An all-unidirectional error-detecting code is able to detect any number of unidirectional errors in the given codeword; in other words, for a given error-detection code, no

unidirectional error can transform a (valid) codeword into another (valid) codeword. Unidirectional error detecting codes find application, for example, in the encoding of unchangeable data on digital optical disks [9].

The use of I-codes is proposed in a very recent paper (Feb 2006) [10] to secure the establishment of the secret key between two entities. In order to guarantee the performance of the system, the authors, besides using I-codes, used another two components to construct their protocol: On-Off Keying, and Signal Anti-Blocking. On-Off Keying is a modulation by which the bit "1" is transmitted on the channel as the presence of a signal and the bit "0" is transmitted as the absence of a signal. Signal Anti-blocking means that the energy of the signal (bit "1") cannot be annihilated by an adversary. With these three components, it is possible to ensure that the adversary on the channel can flip only bits "0" but not bits "1" and that if a bit is flipped, this will be detected at the receiver, which is guaranteed by the properties of I-codes. With these three components, the size of the message must be limited in order to achieve reasonable performance. I-codes are used by the authors of [10] to guarantee that the receiver will detect any flipped "1", but can not correct it. The receiver has to discard the message and request for its retransmission.

The proposed modification on the use of I-codes tends to relax the constraints on the modulation, the required transmission power and the message size. At the same time, it guarantees the detection and the correction of any flipped bit.

The mechanisms, which we are using to prevent the attacker from modifying the exchanging key material without being noticed, are based on using the modified I-codes and also a new variant of the Diffie-Hellman (DH) key exchange protocol. The variant of DH key exchange used in this paper allows the initiator (Alice) to communicate with a single party (Bob) or a group of people to establish a shared (conference) key. The first phase of the proposed DH protocol includes the exchange of the protocol participants' public keys. The second phase involves the establishment of the shared symmetric (conference) key. This phase can be delayed to latter stage when the session key is needed. The proposed variant can also be used for authentication based on the established public key infrastructure.

DH key agreement is known to be vulnerable to the man-in-the-middle attack if the two users involved in the protocol do not share any authenticated information (e.g., public keys, certificates, passwords, shared keys, etc.) about each other prior to the protocol execution. We solve this problem by leveraging on I-codes that can enable message integrity protection and thus preventing Man-In-The-Middle attacks. This will improve the performance of the system and cancels the need for retransmission.

The paper is organized as follows. In Section 2, we state our problem and describe the system and attacker model. In Section 3, we introduce our Key-agreement protocol. In Section 4, we introduce the notion, properties of I-codes and our proposed modifications. Section 5, we discuss the security of the proposed protocol and the security of the modified I-codes are discussed and finally, in Section 6, the results of the modified I-codes implementation are presented.

2. Problem Statement and Assumptions

In this paper we consider the following problem: *Assuming that two entities (**A** and **B**), each equipped with a personal device and share a common (radio) communication channel, but they do not share any secrets or authentication material (e.g., shared keys or authenticated public keys).* (The well-known pure Peer-To-Peer (P2P) model is thus applicable). In addition, the two entities can communicate only over a radio channel without any infrared or physical ports available. The Man-In-the-Middle (MITM) attack is considered feasible in these conditions.

The challenge is the following: *How can the entities (the users - **A** and **B**) establish, in the presence of an attacker (**M**) and in the presence of channel noise a shared key in a secure way?*

The users can use the shared key to authenticate and preserve the integrity of the subsequent message exchanges between themselves in the presence of the attackers.

We assume that the two entities, ***A*** and ***B***, that are involved in the communication do trust each other; otherwise, the problem is not solvable. Whenever we speak of the security of a given protocol, we implicitly assume that the entities involved in the protocol are not compromised. We do assume that the entities know the (public) protocol/security parameters.

Attacker Model: We adopt the following attacker model. Let us assume that the attacker (***M***) controls the communication channel. He can eavesdrop messages and modify transmitted messages by adding his own messages to the channel. It is further assumed that ***M*** can initiate a conversation with any other user but he cannot disable the communication channel (e.g., using Faraday's cage to block the propagation of radio signals) between ***A*** and ***B.*** The attacker can jam the transmission and in a way to prevent the transmission of the information contained in the message. However, the receiver will still receive the message from the sender, superimposed by the messages of the attacker. This model covers the channel noise, which, actually, has the same effect. Finally, we assume ***M*** to be *computationally bounded*.

3. Key Agreement

Key management services should adhere to the following (beside others) generic security attributes:

Confidentiality: Key management schemes should guarantee key secrecy, i.e. ensure the inability of adversaries or unauthorized parties to learn (even partial) key material.

Key Authentication*:* Key authentication is a property whereby a communication entity is assured that only the specifically intended and authenticated communication entity may gain access to the cryptographic key material.

Key authentication, in the context of a communication session between two parties, can be either *unilateral* or *mutual*: unilateral authentication means that only one party's keying material is authenticated, while mutual authentication involves validating both parties' keying material.

Key Confirmation*:* If key confirmation is provided by a key establishment protocol, communication entities prove possession of authenticated keying material. Key authentication with key confirmation yields *explicit* key authentication.

Proposed Key-establishment Protocol:

A variant of Diffie-Hellman key agreement is used in this paper. The proposed variant is an appropriate mechanism to avoid MITM and to provide mutual authentication. We are assuming that Alice ***(A)*** and Bob ***(B)*** start the session with zero

knowledge on each other. We assume also that the generator g and the number n are known globally.

Stage 1: Creating and exchanging public keys:

Alice (***A***) creates her pair of public/private keys (K_A, k_A), chooses a random number N_A and identity ID_A. ***A*** forms the concatenation $N_A \| ID_A \| K_A$ and gets its hash $h(N_A \| ID_A \| K_A)$.

Where, h is a *hash* function satisfying certain security properties and "$\|$" denotes a concatenation.

Message 1*:** ***A sends to ***B*** using modified I-codes, On-Off Keying and Signal Anti-Blocking (see Section 4) the following message:

A → ***B***: I-code { $ID_a \| [\xi_a = h(N_A \| ID_A \| K_A)]$ }

B creates her pair of public/private keys (K_B, k_B), chooses a random number N_B and identity ID_B. (ID_A and ID_B are human readable identifiers belonging to parties ***A*** and ***B*** (e.g., e-mail addresses).)

Message 2*:** ***B sends to ***A*** using modified I-code the following message:

B → ***A***: I-code{ $ID_B \| [\xi_b = h(N_B \| ID_B \| K_B)]$ }

Message 3: ***A*** sends to ***B*** on the Radio channel the message:

A → ***B***: $N_A \| ID_A \| K_A$

B calculates the hash of the received message and compares it with ξ_a received at the first message. If they match, i.e. if $\xi_a = h(N_A \| ID_A \| K_A)$, ***B*** accepts the public key K_A and aborts the protocol if the check fails.

Message 4: ***B*** sends to ***A*** on the radio channel the following message:

B → ***A***: $N_B \| ID_B \| K_B$

A checks if $\xi_B = h(N_B \| ID_B \| K_B)$.

If they match, ***A*** accepts K_B as the public key of ***B***, otherwise she aborts the protocol if the check fails.

Stage 2: Exchanging the secret shared key and the authentication:

1. ***A*** selects her secret exponent x randomly from the set {1, 2, . . . , q} (q being the order of an appropriate multiplicative group), and calculates the ***secret key***:
   $k = g^x \bmod n$
2. ***B*** selects his secret exponent y and calculates
   $Y = g^y \bmod n$

B forms the concatenation $m_B = 1 \| ID_B \| N_B \| Y$ and signs it by his private key $\delta_{k_B}(m_B)$.

Message 5: ***B*** sends to ***A*** on the radio channel the message:

B → ***A***: $Y \| \delta_{k_B}(m_B)$

3. ***A*** verifies the signature of ***B*** using the received public key K_B, verifies the integrity of the received message by the 1 at the least significant position and also the received value of Y. Verifying the signature with the received public key serves, at the same time, as a proof that ***B*** knows his private key, i.e. it *authenticates* ***B*.**

   A calculates: $X = Y^x \bmod n$

Then forms the concatenation $m_A = 0 \| ID_A \| N_A \| X$ and sign it by her private key $\delta_{k_A}(m_A)$

Message 6: ***A*** sends to ***B*** on the radio channel the message:

A → ***B***: $X \| \delta_{k_A}(m_A)$

4. ***B*** verifies the signature of ***A*** using the received public key K_A, verifies the integrity of the received signal and proof that ***A*** knows his own private key that *authenticates* ***A*.**

   B computes: $z = y^{-1}$

Then calculates the *secret shard key k*: $k = X^z$

Stage 3: Key confirmation

B calculates $i_B = \hat{N}_A \oplus N_B$ (1)

where, $\hat{N}_A$ is the value of N_A as received in message 6.

Message 7: ***B*** uses the secret shared key to encrypt i_B and sends the result to ***A*** using I-code channel (together with on-off keying and anti-blocking)

B → ***A***: I-code $\{E_k(i_B)\}$

5. ***A*** uses k to get i_B and calculates

$$i_A = \hat{N}_B \oplus N_A \quad (2)$$

If $i_A = i_B$, ***A*** will know without any doubt that ***B*** successfully generated the same key.

It is enough in this case for Alice to announce this result by sending a message to Bob confirming the mutual acceptance of all the correspondences and that it is the time to start the session. The confirmation message may be sent on the radio channel or using I-code channel.

A → ***B***: I –code $\{E_k(i_A)\}$

The messages number 3 to 6 are exchanged over a radio link. I-code is used in the first two steps to guarantee the exchange of the public keys. In the last two steps Bob transmits the acceptance confirmation message using modified I-code, which guarantees its integrity, and ensures that Alice receives the message without any possibility of channel-induced error.

4 Integrity-Codes (I-Codes)

I-codes allow a receiver ***B*** to verify the integrity of the message received from the sender ***A***, based solely on message coding. I-codes are used with communication channels for which we can ensure that it is not possible to block emitted

signals without being detected, except with a negligible probability.

The formal definition of integrity code (I-code) considers it either as a *triple* (S, C, e) or as a seven-tuple (S, C, e, P, l, w, e_c), where:

1. S is a finite set of possible source states (plaintext)
2. C is a finite set of binary codewords
3. e is a source encoding rule $e : S \rightarrow C$, satisfying the following:

• e is an injective function

• it is not possible to convert codeword $c \in C$ to another codeword $c' \in C$, such that, $c' \neq c$, without changing at least one bit "1" of c to bit "0".

4. e_c is the channel modulation function.
5. w is the Hamming weight.
6. P is a set of power.
7. l is the length of the codeword.

From the above definitions, we can note that the set C consists of l-bit long codewords, each containing a uniquely ordered w sequence of symbols "1" and $(l - w)$ symbols "0."

To make the above definition more concrete, consider the following two examples of I-codes.

Example 1: Complementary encoding (also Manchester code.)

The encoding rule (e) is the following:

$1 \rightarrow 10$

$0 \rightarrow 01$

Assume now that we want to encode messages from the set $S = \{00, 01, 10, 11\}$ using the above encoding rule.

Then, $C = \{0101, 0110, 1001, 1010\}$, i.e., $e(00) = 0101$, $e(01) = 0110$, $e(10) = 1001$, and $e(11) = 1010$. This encoding rule is clearly injective. Note further that each codeword $c \in C$ is characterized by the equal number of "0"s and "1"s. Therefore, it is not possible to convert one codeword $c \in C$ to a different codeword $c' \in C$, without flipping at least one bit "1" to bit "0". For example, to convert $c = 0110$ into $c' = 0101$, the third bit of c has to be changed to 0. By Definition, the triple (S, C, e) is an I-code.

Example 2: Codes with fixed Hamming weight.

We encode each source state $s \in S$ into a binary sequence (codeword) of the fixed length l and fixed Hamming weight w.

For binary sequences, Hamming weight, is the number of bits "1" in the binary sequence. As in the previous example, suppose $S = \{00, 01, 10, 11\}$. Let $l = 4$ and $w = 1$ (accordingly, each sequence has $l - w = 3$ symbols "0").

Then the number of possible binary sequences of length l and with Hamming weight w is $\binom{l}{w} = \binom{4}{1} = 4$; *i.e.* $\{0001, 0010, 0100, 1000\}$. Let us define the set of codewords C as follows: $C \equiv \{0001, 0010, 0100, 1000\}$. Suppose further the following source encoding rule e: $00 \rightarrow 0001$, $01 \rightarrow 0010$, $10 \rightarrow 0100$ and $11 \rightarrow 1000$. Clearly, e is injective. Moreover, no codeword $c \in C$ can be converted into a different codeword $c' \in C$, without flipping at least one bit "1" of c to bit "0". Therefore, by Definition, the triple (S, C, e) is an I-code.

4.1 I-codes on the Radio Channel

Let us assume that m denotes the message for which integrity should be checked.

Using a given I-code (e.g. the complementary encoding rule), the sender first encodes m into the corresponding I-code codeword c. Due to the injective property of I-codes, it is possible to recover unambiguously message m from the codeword c. Any technique proposed for transmitting c over a given radio channel must guarantee the fulfillment of the basic concept of I-codes; I-codes are to be used with communication media (channels) for which we can ensure that it is not possible to block emitted signals without being detected, except with a negligible probability. The use of suitable modulation technique, suitable waveforms and suitable signal energy level can achieve this goal.

A. Modulation Technique: The sender uses, at the physical layer, the following *On-Off Keying* modulation that satisfies the following conditions:

- P is a set of power. It consists of two power levels 0 and p with $p > 0$, *i.e.,* $P \equiv \{0, p\}$;
- e_c is a channel modulation function satisfying the following rules: a) the symbol "1" is transmitted using power level p and b) the symbol "0" is transmitted using power level 0.

For each symbol "1" of c, the sender emits some signal (waveform) during the period T_s (the *symbol period*). For each symbol "0" of c, however, the sender emits nothing during period T_s. The waveforms that are transmitted do not carry any information, but it is the *presence* or *absence* of energy in a given time slot of duration T_s that conveys information "1".

As an example, if it is required to send the source state "10" to a designated receiver, the transmitter emits a signal described by the following sequence $0p00$ (using the fixed Hamming weight code explained above) . This sequence is interpreted as follows: only during the period of the second symbol, the sender emits some signal (energy) over the used channel.

B. Power levels: For the proper decoding of I-codes, the users have to ensure that the average power received, $(p_r(i))$, by a designated receiver during the period T_s corresponding to a symbol "1" is greater or equal to a pre-defined threshold power level p_1 (i.e., $p_r(i) > p_1$, for all i corresponding to symbol "1").

This condition can be satisfied with high probability by using relatively high transmission powers and ensuring that the distance between the sender and the designated receiver is relatively short.

Based on that, in order to retrieve the codeword transmitted, the receiver simply measures the energy in the corresponding time slots of duration T_s. For the given time slot, the receiver decodes the received signals as follows: (1) if $p_r \geq p_1$, output symbol "1", and (2) if $p_r < p_1$, output symbol "0". We note here that p_r represents the average power received in the interval duration T_s.

Finally, the receiver uses the inverse of the used encoding rule (i.e., $01 \rightarrow 0$, $10 \rightarrow 1$) to retrieve the emitted message m.

C. Use of large number of waveforms: To further increase the robustness of I-codes, the transmitter can have a large number of symbol "1" waveforms, one of which the transmitter chooses randomly for each symbol "1" transmitted. In this way, the attacker does not know what waveform to try

to cancel. At the same time, the receiver only measures the energy he receives during intervals of duration and so any of the "1" waveforms are equally good for this purpose. Note that the receiver does not have to know which waveform the transmitter uses in a given time interval as shown in Fig.1.

All that the receiver has to know is the frequency band used by the sender; the receiver can be thought of as being a bank of radiometers measuring the energy in the given frequency band.

It is important to note here that this measure can be used only if the designated receiver can demodulate the signal approximately at the same speed as the attacker.

Finally, the on-off keying modulation implies that the adversary has to delete (cancel) at least one signal (waveform) emitted on the channel.

However, according to our assumption, the adversary can delete the signal emitted on the used radio channel only with a negligible probability. Also, the transmitter should transmit signals using the power level high enough so that the average power as measured by the receiver is above the threshold p_1.

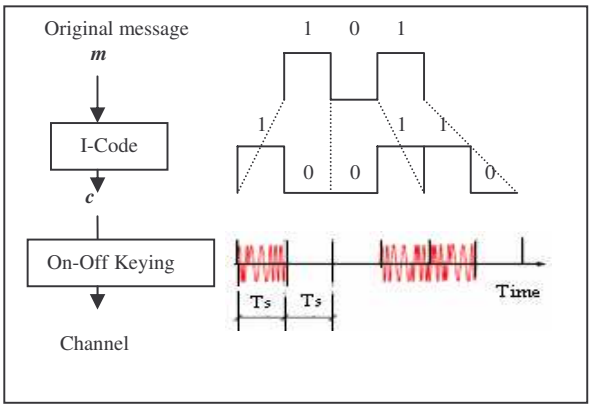

Figure 1. I-coding An example of I-coding at the sender using the complementary encoding rule: 1 → 10 and 0 → 01

I-Codes as Error Detecting and Correcting Code

I-codes, as defined above, can be considered as an error detecting code. In case of detecting an error (bit-flipping) the message must be retransmitted. To reduce the possibility of error (induced by attacker or due to channel noise), the above discussion proposes the use of a high value for the pre-defined threshold power level p_1 to make it difficult to be compromised by attacker and at the same time, it uses large number of waveforms to represent binary "1". Increasing the power threshold reduces the lifetime of the battery and using a large number of waveforms adds complexity to the system. Even with that, the possibility of error is there and the need of retransmission exists. These factors put a limit on the size of the message to be transmitted in order to get a suitable performance.

It is possible to convert the I-code into an error correcting and detecting code. The modified code can detect any bit in error. It splits the message into frames of 4-bit each and can correct any one bit in the frame.

Consider the complementary encoding; the detection of one bit being in error within a codeword is possible by calculating the XOR of the two bits forming the codeword, i.e. to calculate $t_i = a_{2i} \oplus a_{2i+1} (i = 0, 1,\ldots)$ with $t_i = 0$ means one of the two bits is in error. It is not possible to identify the bit in error in order to correct it. Assume now that every two successive codeword (four binary bits) form one frame. Let a_0 a_1 a_2 a_3 represent the bits forming any arbitrary frame. We concatenate to each frame a parity bit p. The value of p is given by:

$$p = a_0 \oplus a_2$$

From the definition of the complementary I-code, if the coding is correct, the following relation is correct:

$$p = a_0 \oplus a_2 = a_1 \oplus a_3$$

It is possible to prove the following equations:

$$\begin{aligned} a_0 &= (a_2.\overline{p} + a_3.p).\overline{t_1} \\ a_1 &= (a_3.\overline{p} + a_2.p).\overline{t_1} \\ a_2 &= (a_0.\overline{p} + a_1 p).\overline{t_2} \\ a_3 &= (a_0.p + a_1\overline{p}).\overline{t_2} \end{aligned} \quad (3)$$

At the receiving end, the receiver calculates t_1 and t_2 and if any of them is "0", thus detecting an error, the system uses equation (3) to calculate the correct bit.

The same approach can be used with I-codes that uses fixed Hamming weight.

5. Security Analysis

The key establishment protocol proposed in Section-3 executes in three stages. In the first stage the two protocol participant, ***A*** and ***B***, exchange their public/private key pairs (*K/k*). In terms of an MITM attack the adversary want to replace any party's public key *K* with its won public key *K'*.

In the *msg* 1 and 2, ***A*** and ***B*** exchange integrity parameters ξ_a and ξ_b as commitments to their keying material. The primary reason for exchanging these commitments, rather than the public keys themselves, is to reduce the amount of data send using I-codes, on-off keying and signal anti-blocking. It should be clear that using I-codes, on-off keying and signal anti-blocking prevent any adversary from jamming or modifying ξ_a and ξ_b. Once the commitment are exchanged ***A*** and ***B*** send to each other (over the radio interface) a unique random number (*N*), identity (*ID*) and self-generated public key (*K*). The random number eliminates a possible replay attack. The only option left to the adversaries is to modify the contents of *msg* 3 and 4, which are sent over the radio link. Based on the properties of a collision free one-way *hash* function, such modification will be detected by Alice and Bob when they check that the commitment received via I-codes match the *hash* of the keying material received over the radio interface.

Once ***A*** and ***B*** exchanged their authentic public keys they continue to establish a shared secret key in stage two, *msg* 5 and 6. The security analysis on the second stage is trivial since the protocol participants can use their private keys to ensure the integrity of the Diffie-Hellman parameters via a digital signature scheme. Note that verifying the signature also implies authentication of the signer, hence Alice and Bob can positively identify each other.

The third and final stage serve as a key confirmation phase. Bob sends to Alice an encryption of [$i = N_A \oplus N_B$] using the shared key k. Decryption of $E_k(i)$ authenticates Bob to Alice and serves as proof that Bob has access to k. Since this exchange is performed via I-codes, On-Off Keying and Signal Anti-Blocking and since the message is encrypted, an adversary cannot jam the signal nor modify the content of *msg* 7 without being detected.

In another paper [11] we proved that if the length of i_A and i_B (equations 1 and 2) is L, the probability that the attacker can generate $\tilde{N}_A$ and $\tilde{N}_B$ such that $\tilde{N}_A \oplus \tilde{N}_B = i_A = i_B$ is $< 2^{-L}$

***Security of I-codes*:** The security of I-codes relies on the property mentioned before: *Assume that we can ensure,* $p_r(i) > p_1$ *for all i corresponding to symbol "*1*," except with a negligible probability. Then, an adversary can neither add nor remove a symbol "*1*" from the communicated sequence without being detected.*

In other words, if we can ensure that any symbol "1" cannot be completely blocked (deleted, annihilated, canceled) once transmitted over a public channel, except with a negligible probability, then an adversary cannot change an "I-coded" message without being detected. In other words, we need to ensure that $p_r(i) > p_1$, for all corresponding to symbol "1."

In the context of a radio channel, canceling a radio signal in such a way that $p_r(i) < p_1$, where p_1 is set to the background noise level, requires sending out the inverted signal that will have exactly the same characteristics (signal level, phase) on the receiver's side. This could be quite challenging to achieve for an adversary. The use of a large number of symbol "1" waveforms by the transmitter increases the robustness of the I-code and increases challenge for the attacker. The attacker does not know what waveform to try to cancel. At the same time, the receiver only measures the energy he receives during intervals of duration Ts and so any of the "1" waveforms are equally good for this purpose. Note that the receiver does not have to know which waveform the transmitter uses in a given time interval.

6. Implementation, Results, and Conclusions

We implemented I-codes using Matlab. We used 100 MHz frequency, FSK modulation spectrum shaping, complementary encoding, White Gaussian noise channel and 0dB. We simulated the two cases of normal I-code and the error detecting and correcting version. We changed the message size from 8 bits to 1024 bits. In each case, the transmission success ratio is calculated.

In case of using normal I-code, the results showed that the transmission success ratio decreases quickly as the message size increases in case of using normal I-code. These results mean that I-codes are best suited for reasonably short messages. For longer messages, we would need to transmit them multiple times in order for one of the messages to be transmitted correctly. The results showed that using the modified I-code gave better performance. With messages size of 1024, the performance of the modified I-code was better than the performance of I-codes with much less size.

7. References

[1] Wi-Fi Protected Access (WPA). [Online]. Available: http:// www.wi-fi.org/wpa/

[2] W. Arbaugh, N. Shankar, Y. Wan, and K. Zhang, "Your 802.11 wireless network has no clothes," *IEEE Wireless Commun.,* vol. 9, no. 6, pp. 44–51, Dec. 2002.

[3] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: the insecurity of 802.11," in *Proc. Int. Conf. Mobile Computing and Networking (MOBICOM)* 2001, pp. 180–189.

[4] G. Koien, "An introduction to access security in UMTS," *IEEE Wireless Commun.,* vol. 11, no. 1, pp. 19–25, Feb. 2004.

[5] M. Shin, J. Ma, A. Mishra, and W. Arbaugh, "Wireless network security and interworking," *Proc. IEEE (Special Issue on Cryptography and Security Issues),* vol. 94, no. 2, pp. 455–466, Feb. 2006.

[6] ——, "Packet leashes: a defense against wormhole attacks in wireless networks," in *Proc. Annu. Joint Conf. IEEE Computer and Communications Societies (INFOCOM)* 2003, pp. 1976–1986.

[7] J. M. Berger. A Note on Error Detecting Codes for Asymmetric Channel. *Information and Control*, 4:68–73, 1961.

[8] J. M. Borden. Optimal Asymmetric Error Detecting Codes. *Information and Control*, 53:66–73, 1982.

[9] E. L. Leiss. Data Integrity on Digital Optical Discs. *IEEE Transactions on Computers*, 33:818–827, 1984.

[10] M.Cagalj, S.Capkun, J.P. Hubaux, "Key Agreement in Peer-to-peer Wireless Networks" Proc. IEEE, vol. 94, no. 2, February 2006, pp. 467-478.

[11] D.S. Dawoud, F.K. Akorli, "Use of I-Codes for Message Integrity Protection and Authentication over Insecure Channels" Submitted for SAIEE 2006

Dr. F. K. Akorli: Received D.EE from the Pontifical Catholic University of Rio de Janeiro. Presently, he is seconded from Kwame Nkrumah University of Science and Technology to National University of Rwanda as the Director of M.Sc in ICT Program. His research interests are mainly in Mobile and wireless communication networks, Network security, Radio propagation.

Dr. D. S. Dawoud: Coordinator of Computer Engineering Program, Faculty of Engineering, University of KwaZulu Natal. His researches are mainly in Computer hardware, Arithmetic Units, Embedded systems, Digital Signal Processing, Network and ad hoc network security, Software Defined Radio.

Mr. J. Van Der Merwe: Has BSc. (2003), and MSc (2005) from University of KwaZulu Natal. He is preparing now his PhD. His main field of interest is ad hoc mobile networks. During the last three years he published 2 IEEE papers and 3 International conferences papers, all in the field of mobile ad hoc networks.