



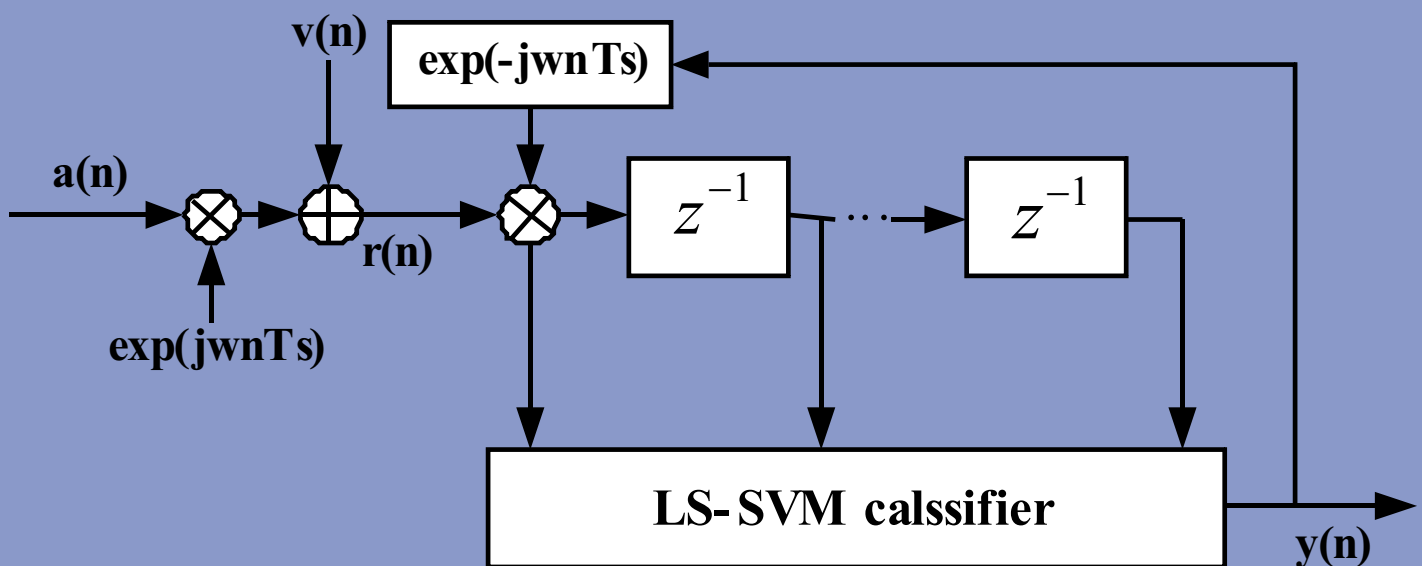
Scientific
Research

International Journal of

Communications, Network and System Sciences

ISSN: 1913-3715

Volume 3, Number 10, October 2010



ISSN: 1913-3715



www.scirp.org/journal/ijcns/

JOURNAL EDITORIAL BOARD

ISSN 1913-3715 (Print) ISSN 1913-3723 (Online)

<http://www.scirp.org/journal/ijcns/>

Editors-in-Chief

Prof. Huaibei Zhou

Wuhan University, China

Editorial Board

Prof. Dharma P. Agrawal

University of Cincinnati, USA

Prof. Eduardo A. Castro

La Plata National University, Argentina

Prof. Hengda Cheng

Utah State University, USA

Prof. Ko Chi Chung

National University of Singapore, Singapore

Dr. Franca Delmastro

Italian National Research Council, Italy

Prof. Mohamed B. El_Mashade

Al_Azhar University, Egypt

Dr. Li Huang

Interuniversity Microelectronics Centre, Netherlands

Prof. Hiroaki Ishii

Kwansei Gakuin University, Japan

Dr. Vladimir A. Masch

Risk Evaluation and Management, Inc., USA

Dr. Nicola Mastronardi

Istituto per le Applicazioni del Calcolo "M. Picone", Italy

Dr. Lim Nguyen

University of Nebraska-Lincoln, USA

Prof. Yi Pan

Georgia State University, USA

Dr. Petar Popovski

Aalborg University, Denmark

Prof. Suresh Rai

Louisiana State University, USA

Dr. Kosai Raoof

Joseph Fourier University, France

Prof. Bimal Roy

Indian Statistical Institute, India

Prof. Shaharuddin Salleh

University Technology Malaysia, Malaysia

Prof. Boris S. Verkhovsky

New Jersey Institute of Technology, USA

Prof. Shi Ying

Wuhan University, China

Editorial Assistant

Vivian QI

Scientific Research Publishing, USA. Email: ijcns@scirp.org

TABLE OF CONTENTS

Volume 3 Number 10

October 2010

A Comparative Analysis of Tools for Verification of Security Protocols

N. Dalal, J. Shah, K. Hisaria, D. Jinwala.....779

Block Layering Approach in TAST Codes

Z. Ahmed, J. P. Cances, V. Meghdadi.....788

Winning Strategies and Complexity of Nim-Type Computer Game on Plane

B. Verkhovsky.....793

A Secure Transfer of Identification Information in Medical Images by Steganocryptography

S. H. Jiao, R. Goutte.....801

Performance Improvement of Wireless Communications Using Frequency Hopping Spread Spectrum

Y. Liu.....805

Solutions for 3 Security Problems and its Application in SOA-FCA Service Components Based SDO

N. N. Wang, Z. Y. Fang, K. Yan, Y. Tang, X. C. An.....811

Reliable Multicast with Network Coding in Lossy Wireless Networks

W. Yan, S. Y. Yu, Y. M. Cai.....816

Using Least Squares Support Vector Machines for Frequency Estimation

X. Y. Teng, X. Y. Zhang, H. Y. Yu.....821

A Turbo Decoder Included in a Multi-User Detector: A Solution to be Retained

S. Kerouédan, M. Touzri, P. Adde, S. Saoudi.....826

The figure on the front cover is from the article published in *International Journal of Communications, Network and System Sciences*, 2010, Vol.3, No.10, pp. 821-825 by Xiaoyun Teng *et al.*

International Journal of Communications, Network and System Sciences (IJCNS)

Journal Information

SUBSCRIPTIONS

The *International Journal of Communications, Network and System Sciences* (Online at Scientific Research Publishing, www.SciRP.org) is published monthly by Scientific Research Publishing, Inc., USA.

Subscription rates:

Print: \$50 per issue.

To subscribe, please contact Journals Subscriptions Department, E-mail: sub@scirp.org

SERVICES

Advertisements

Advertisement Sales Department, E-mail: service@scirp.org

Reprints (minimum quantity 100 copies)

Reprints Co-ordinator, Scientific Research Publishing, Inc., USA.

E-mail: sub@scirp.org

COPYRIGHT

Copyright©2010 Scientific Research Publishing, Inc.

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as described below, without the permission in writing of the Publisher.

Copying of articles is not permitted except for personal and internal use, to the extent permitted by national copyright law, or under the terms of a license issued by the national Reproduction Rights Organization.

Requests for permission for other kinds of copying, such as copying for general distribution, for advertising or promotional purposes, for creating new collective works or for resale, and other enquiries should be addressed to the Publisher.

Statements and opinions expressed in the articles and communications are those of the individual contributors and not the statements and opinion of Scientific Research Publishing, Inc. We assume no responsibility or liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained herein. We expressly disclaim any implied warranties of merchantability or fitness for a particular purpose. If expert assistance is required, the services of a competent professional person should be sought.

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:

E-mail: ijcns@scirp.org

A Comparative Analysis of Tools for Verification of Security Protocols

Nitish Dalal, Jenny Shah, Khushboo Hisaria, Devesh Jinwala

Department of Computer Engineering,

S.V. National Institute of Technology, Ichchhanath, Surat, India

E-mail: dcj@svnit.ac.in

Received July 20, 2010; revised August 21, 2010; accepted September 24, 2010

Abstract

The area of formal verification of protocols has gained substantial importance in the recent years. The research results and subsequent applications have amply demonstrated that the formal verification tools have indeed helped correct the protocols even after being standardized. However, the standard protocol verification tools and techniques do not verify the security properties of a cryptographic protocol. This has resulted in the emergence of the security protocol verifiers to fill the need. In this paper, taking the two popular security verification tools namely Scyther and ProVerif as the basis, we identify a few security protocols and implement them in both Scyther and ProVerif, to aptly evaluate the tools, in terms of the security properties of the selected protocols. In the process, we not only characteristically present a comparative evaluation of the two tools, but also reveal interesting security properties of the protocols selected, showing their strengths and weaknesses. To the best of our knowledge, this is a unique attempt to juxtapose and evaluate the two verification tools using the selected security protocols.

Keywords: Formal Verification, Security Protocols, Attacks

1. Introduction

A protocol is a set of rules that followed the defined conventions to establish semantically correct communications between the participating entities. A security protocol is an ordinary communication protocol in which the message exchanged is often encrypted using the defined cryptographic mechanisms. The mechanisms Symmetric Key Cryptography or Asymmetric Key Cryptography are used to obtain various cryptographic attributes such as *Confidentiality*, *Entity Authentication*, *Message Integrity*, *Non-repudiation*, *Message Freshness*, to name a few [1]. However, merely using cryptographic mechanisms, does not guarantee *security-wise semantically secure operation* of the protocol, even if it is correct. There indeed have been reported breaches in the security protocols, after being published and accepted as a safe protocol [2-4]. In such a scenario, in case of the ordinary communication protocols, recourse has been taken to the *rigorous verification* of the same using appropriate tool for the domain. As for example, the protocol verifier *SPIN* is used to verify the communication protocols for distributed software [5].

Such successful use of the formal methods for verification has led to the upsurge in devising similar tools for verifying the security properties of a cryptographic protocol, too. In order to gain confidence in the cryptographic protocol employed, it has been found desirable that the protocol be subjected to an exhaustive analysis that verifies its security properties. Some of the tools developed for the purpose are Scyther [6], ProVerif [7], Athena [8], Avispa [9], Casper/FDR to name a few. These tools differ in their input language and also in the way they verify the protocols and provide the output.

However, an important fall-out of the emergence of a plethora of such tools is that, it often becomes difficult for a security engineer to identify the appropriateness and suitability of a tool for the protocol under consideration. Motivated with this difficulty, we in this research report, document our attempt at evaluating the two popular cryptographic verification tools namely *ProVerif* and *Scyther*. We use six popular cryptographic protocols to implement the same and then analyze the protocols, using both these tools. In the process, interestingly, we not only comparatively evaluate the tools under consid-

eration, but also identify various interesting properties of the protocols used.

To the best of our knowledge, this is a unique attempt to juxtapose and evaluate the two verification tools using the security protocols namely the *Kao Chow Authentication Protocol* [10], the *3-D Secure Protocol* [11], the *Needham Schroeder Public Key Protocol* [3], the *Andrew Secure RPC Protocol* [12], the *Challenge Handshake Authentication Protocol* [13] and the *Diffie-Hellman Key Exchange Protocol* [14].

The rest of this paper is organized as follows: in section 2, we describe the theoretical background which includes a brief introduction of Scyther and ProVerif tools. In section 3, we formally define the problem and survey the related work. In section 4, we present details of our implementation of various protocols, whereas do a comparative analysis of the two tools, before we draw the conclusion and show probable future work, in the last section.

2. Theoretical Background

2.1. Cryptographic Verification Tools

As mentioned before, one can find a series of tools for the verification of the cryptographic protocols. We have selected Scyther and ProVerif amongst all these, for the comparative evaluation. This decision is largely driven by the popularity of these two tools amongst all, we surveyed. In this section we depict the vital characteristics of these two tools.

2.1.1. Scyther

Scyther is a tool used for security protocol verification, where it is assumed that all the cryptographic functions are perfect. The tool provides a graphical user interface that makes it easier to verify and understand a protocol. In addition, attack graphs are generated whenever an attack is found corresponding to the claim mentioned. The tool can also verify all the possible claims on the protocol. The tool can be used to find problems that arise from the way the protocol is constructed. It can also be used to generate all the possible trace patterns. The verification here can be done using a bounded or an unbounded number of sessions. The language used to write protocols in Scyther is SPDL (Security Protocol Description Language) [6].

2.1.2. ProVerif

ProVerif is a software tool for automated reasoning about the security properties found in cryptographic protocols. This was developed by Bruno Blanchet. This tool

verifies the protocol for an unbounded number of sessions, using unbounded message space. The tool is capable of attack reconstruction—wherein if a property cannot be proved, an execution trace which falsifies the desired property is constructed. There are two ways of providing input to this tool—Horn clauses or Pi calculus. In both cases, the output of the tool is essentially the same. Explicit modeling of attacker is not required. It is also possible to state whether an attacker is active or passive [7].

3. Related Work

The main objective of our research is to dissect the two protocol verification tools, Scyther and ProVerif, and to provide a comparative analysis of the two. In order to analyze the tools, suitable standard input protocols are required to be identified. After careful observation of a series of such protocols, we have identified, implemented and analyzed six different security protocols using both these tools. As mentioned earlier, these protocols are namely the Kao Chow Authentication Protocol, the 3-D Secure Protocol, the Needham Schroeder Public Key Protocol, the Andrew Secure RPC Protocol, the Challenge Handshake Authentication Protocol and the Diffie Hellman Key Exchange Protocol.

One can find a few attempts in the literature that concentrate on tools used for protocol verification, whereas very few of them provide a comparative analysis of protocol verification tools as in [15] and in [16]. However, there is no attempt that either focuses on or subsumes a detailed comparative analysis of the tools Scyther and Proverif, using the actual implementation of protocols as the basis. Hence, we believe our attempt here to be a unique one of its kind.

In the next section, we discuss briefly the implementation of each of the protocol in Scyther as well as in ProVerif and analyze the same.

4. Implementation and Analysis

4.1. Kao Chow Authentication Protocol

4.1.1. Definition

The Kao Chow protocol is a mutual authentication and key distribution protocol aiming at strong authentication and low message overhead. A trusted third party, *S*, is used to generate and distribute keys. It is the responsibility of *S* to generate a fresh secret session key *k*. The two communicating parties, *A* and *B*, will use this key to encrypt future message exchanges. Both the parties have

secret keys shared with the server, *Kas* and *Kbs*, which are used to encrypt and decrypt any messages that are exchanged. Further, the protocol aims to authenticate the parties *A* and *B* to each other using their nonces *Na* and *Nb* [10]. The pseudocode of a typical protocol execution run is shown in **Figure 1**.

```

Kao_Chow(Kas, Kbs)

/* nonce = random number,
id_X = identity of X,
A,B = the communicating parties,
S = Server,
Kas = symmetric key between A and S,
Kbs = symmetric key between B and S,
k = session key between A and B */

{
  Call (Sender_A(Kas) | Receiver_B(Kbs) |
        Server_S(Kas,Kbs)).
  /* Call the functions in parallel */
}

Sender_A(Kas)
{
  StepA1:generate nonce Na;
  msg1 = (id_A,id_B,Na);
  send msg1 to S;
  call StepS1.
  StepA2:receive msg3 from B;
  let msg3 = (part1, part2, Nb');
  let (id_A'',id_B'',Na'',k)
  =(sym_decrypt(part1) with Kas) AND
  check_if(id_A''=id_A) AND check_if(id_B''= id_B)
  AND check_if(Na''=Na);

  let (Na''')=(sym_decrypt(part2) with k) AND
  check_if(Na'''=Na);
  msg4 = sym_encrypt(Nb') with k;
  send msg4 to B;
  call StepB2.
}

Server_S(Kas, Kbs)
{
  StepS1:receive msg1 from A;
  let (id_A',id_B',Na') = msg1;
  generate session_key k;
  msg2 = ((sym_encrypt
  (id_A',id_B',Na',k)
  with Kas),
  (sym_encrypt(A',B',Na',k)
  with Kbs));
  send msg2 to B;
  call StepB1.
}

Receiver_B(Kbs)
{
  StepB1:receive msg2 from S;
  let(part1,part2) = msg2;
  let(id_A'',id_B'',Na'',k)=
  sym_decrypt(part2) with Kbs;
  generate nonce Nb;
  msg3 = (part1,(sym_encrypt (Na'')
  with k),Nb);
  send msg3 to A;
  call StepA2.
}

```

```

StepB2:receive msg4 from A;
let(Nb''')=(sym_decrypt msg4 with k)
AND check_if(Nb''' = Nb);
return to Kao_Chow(.
}

```

Figure 1. Kao Chow authentication protocol.

4.1.2. Analysis

When this protocol is verified using Scyther, attacks are found on the sender side as well as on the receiver side. However, we observe that the session key on the sender side (*A*) is secret whereas it is compromised on the receiver side. Because of this, there are synchronization and agreement attacks on both the sides. For the claim to check the secrecy of session key on the receiver (*B*) side, scyther outputs saying the claim is “*Falsified*” and that there is “*At least 1 attack*.” When a similar query is written for the initiator side, scyther gives the output that the claim has been “*Verified*” and that there are “*No attacks*”.

When the same protocol is analyzed using ProVerif, we obtain a similar result. That is, it can also detect that the session key is not secret on the receiver side. The private free variable *var* is encrypted using the session key *k* and published over a public channel *c*, the output obtained is false *i.e.*, the key is not secret on the initiator side. When a similar query is provided for the receiver side, ProVerif gives the output saying that session key is secret on the receiver side. When the parameter “*attacker*” is set to *passive*, we do not obtain any attacks showing that there are no passive attacks on the protocol. The symmetric key *Kas* is secret on *A* side as well as the server(*S*) side. *Kbs* is also secret on the server side as well as *B* side.

4.2. 3-D Secure Protocol

4.2.1. Definition

The 3-D Secure is an XML-based protocol used as an added layer of security for online credit and debit card transactions. Developed by Visa, its aim is to improve the security of Internet payments. It is offered to customers as the *Verified by Visa* service. It has also been adopted by MasterCard, under the name MasterCard *SecureCode*.

A transaction using Verified by Visa/SecureCode will initiate a redirect to the website of the card issuing bank to authorize the transaction. Each Issuer could use any kind of authentication method. The most common approach is a password-based method. Thus, to effectively buy on the Internet means using a secret password tied to the card. The Verified by Visa protocol recommends the bank’s verification page to load in an inline frame session. In this way, the bank’s systems can be held responsible for most security leaks [11].

The pseudocode of the protocol is shown in **Figure 2**.

```

3-D_Secure(skC,pkC,skM,pkM,skB,pkB)

/* nonce = random number,
C = Customer, M = Merchant, B = Bank,
pkX = public key of X,
skX = secret key of X,
PI = Payment information, with transaction ID,
OI = order information, with transaction ID, PIMD = message
digest of PI,
OIMD = message digest of OI,
POMD = message digest of concatenation of PIMD and OIMD.
*/
{
    Call (Customer_C(skC,pkM,pkB) | Mer-
    chant_M(skM,pkC,pkB) |
    Bank_B(skB,pkC,pkM)).
/* Call the functions in parallel */
}

Customer_C(skC,pkM,pkB)
{
    StepC1: generate nonce Nc;
    msg1 = encrypt(brand,Nc) with pkM;
    send msg1 to M;
    call StepM1.
    StepC2: receive msg2 from M;
    let (Nc',tid) = (decrypt msg2 with skC) AND
    check_if(Nc' = Nc);
    generate session_key Kbc;
    /* between B and C */

    msg3=((sym_encrypt(PI,(encrypt(hash
    cat(hash(PI) AND hash(OI)))) (con-
    with skC),hash(OI)) with Kbc),
    (encrypt Kbc with pkB), OI, hash(PI),
    (encrypt(hash(concat(hash(PI) AND
    hash(OI)))) with skC));
    send msg3 to M;
    call StepM2.

    StepC3: receive msg5 from B;
    let (user) = decrypt msg5 with skC;
    msg6 = encrypt(user,hash(password)) with
    pkB;
    send msg6 to B;
    call StepB2.
}

Merchant_M(skM,pkC,pkB)
{
    StepM1: receive msg1 from C;
    let (brand',Nc') = decrypt msg1 with
    skM;
    generate new tid;
    msg2 = encrypt(Nc',tid) with pkC;
    send msg2 to C;
    call StepC2.
    StepM2: receive msg3 from C;
    Let (part3a,part3b,part3c, part3d,part3e) =
    msg3 AND
    check_if((decrypt(part3e) with pkC)
    =hash(concat(part3d
    AND hash(part3c)))));

    /* compare the received and calculated values of
    POMD */
}

```

```

generate session_key Kbm;

/* between B and M */

msg4 = (part3a,part3b,(sym_encrypt (en-
crypt tid with skM) with Kbm),(encrypt Kbm with
pkB));
send msg4 to B;
call StepB1.
StepM3: receive msg7 from B;
Let (part7a,part7b,part7c,part7d) = msg7;
let (K'mb)=decrypt (part7b) with skM;
let (tid') = (decrypt(sym_decrypt part7a with K'mb)
with pkB) AND check_if(tid'=tid);
msg8 = (part7c,part7d,(encrypt (en-
crypt (tid,amount)
with skM) with pkB));
send msg8 to B;
call StepB3.
StepM4: receive msg9 from B;
let (tid') = (decrypt msg9 with skM)
AND check_if(tid' =tid);
return to 3-D_Secure( ).
}

Bank_B(skB,pkC,pkM):
{
    StepB1: receive msg4 from M;
    let (part4a,part4b,part4c,part4d) = msg4;
    let Kbc = decrypt (part4b) with skB;
    let (PI,encrypted_POMD,OIMD) =
    sym_decrypt(part4a) with Kbc;
    let POMD = (decrypt(encrypted_POMD) with pkC)
    AND check_if(POMD = hash(concat(hash(PI)
    AND OIMD)));

    /* compare the received and calculated values of
    POMD */

    let Kbm = decrypt (part4d) with skB;
    let tid = (decrypt(sym_decrypt
    (part4c) with ks2) with pkM);
    msg5 = encrypt user with pkC;
    send msg5 to C; call StepC3.
    StepB2: receive msg6 from C;
    let (user',password') = (decrypt msg6 with
    skB) AND
    check_if(user' = user);
    generate session_key K'mb;

    /* between M and B */

    generate nonce Nb;
    msg7 = ((sym_encrypt(encrypt tid with skB)
    with K'mb),
    (encrypt K'mb with pkM),
    (sym_encrypt(encrypt Nb with skB) with K'mb),
    (encrypt K'mb with pkB));
    send msg7 to M;
    call StepM3.
    StepB3: receive msg8 from M;
    let (part8a,part8b,part8c) = msg8;
    let (K'mb) = decrypt(part8b) with skB;
    let (Nb') = (decrypt(sym_decrypt (part8a) with
    K'mb) with pkB) AND check_if(Nb' = Nb);
    let (tid',amount) = (decrypt (de-
    crypt(part8c) with skB)
    with pkM) AND
    check_if(tid' = tid);
}

```



```

msg9 = encrypt tid with pkM;
send msg9 to M;
call StepM4.
}

```

Figure 2. 3-D secure protocol.

4.2.2. Analysis

On analyzing this protocol with Scyther, we find that for five runs of the protocol, there are no attacks on the customer(C) and merchant(M) side. However, the attacks are found on the bank(B) side. The session key K_{bc} and the customer's password are secret on C side. On the M side, we find that the session keys K_{bm} , K'_{mb} and K_{bc} are secret. On analyzing the claims on B side, we find that the session key K'_{mb} and the customer's password are secret. But, the keys K_{bc} and K_{bm} can be compromised here. These attacks bring a limitation of the tool Scyther to the fore. In Scyther, we have no provision of comparing the values of two variables. For instance, in step 4, the bank receives POMD, PI and hash (OI). However, there is no way that we can write an "if" condition in this tool to check the equality of the received POMD and the calculated POMD (calculated using the function "hash" and the received value of hash(OI) and PI). The attacks that we see here are the result of such deficiencies in the tool. Had there been a way to compare values here, no attacks would have been found. In order to obtain attacks on session keys, a special "key-compromise" part needs to be added to the protocol. In this module, we first specify who the communicating parties are. Next, we provide the intruder with all the packets that have been used in a session and the values of the session keys for that session in order to verify if a freshness attack is possible.

When the 3D-secure protocol is analyzed using ProVerif, no attacks are found. That is, the tool says that this protocol is perfectly secure and that there is no way that an intruder can gain knowledge of either the session keys (K_{bc} , K_{bm} , K'_{mb}) or the customer's password. This is because, in this tool we have the provision of writing an "if" condition to check for the equality of two values. For instance, in step B1, we see that the value POMD obtained after decrypting a part of the message can be compared to the calculated value of POMD (using the received values PI and OIMD and the hash function) and the protocol proceeds only if these two values are found to be the same. This way, all the attacks are countered and the protocol becomes completely secure.

Thus, we see that ProVerif provides this advantage over Scyther. Not being able to compare values in Scyther leads to attacks being found whereas in ProVerif, as the values can be compared, these attacks are not found.

4.3. Needham-Schroeder Public Key Protocol

4.3.1. Definition

The Needham-Schroeder Public Key Protocol, based on public-key cryptography, is intended to provide mutual authentication between two parties communicating on a network. It is assumed here that the two parties know the public key of the other. Thus, encrypting the data is possible using the public key of the other party. Here, A and B represent the communicating parties [3].

The pseudocode of the protocol is shown in **Figure 3**.

4.3.2. Analysis

On verifying this protocol using Scyther, attacks are found. It is seen that all the attacks are on the B (receiver) side. Both the nonces, Na and Nb , can be obtained by the attacker. In addition, there are synchronization and agreement attacks. Thus, the protocol is not secure. In addition, Scyther provides the facility of observing all possible trace patterns. For this protocol, a single trace pattern is obtained on the A (initiator) side as there is no intrusion possible here.

Needham_Schroeder(skA,pkA,skB,pkB)

```

/* pkX = public key of X,
skX = secret key of X,
nonce = random number,
id_X = identity of X,
A,B = the communicating parties. */

```

```

{
    Call (Sender_A(skA,pkB) | Receiver_B(skB,pkA)).
    /* Call the functions in parallel */
}

```

Sender_A(skA, pkB)

```

{
    StepA1: generate nonce Na;
    msg1 =(encrypt(Na, id_A) with pkB);
    send msg1 to B;
    call StepB1.
    StepA2: receive msg2 from B;
    let(Na',Nb',X)=(decrypt(msg2) with skA) AND
    check_if(Na' = Na) AND check_if(X = B);

    /* assign the 3 parameters of msg2 on decryption to
    Na',Nb',X respectively */

    msg3 = (encrypt(Nb') with pkB);
    send msg3 to B;
    call StepB2.
}

```

Receiver_B(skB,pkA)

```

{
    Step B1: receive(msg1) from A;
    let(Na',A')=(decrypt(msg1)with skB);
    generate nonce Nb;
    msg2 = (encrypt(Nz,Nb,B) with pkA);
    send(msg2) to A;
    call StepA2.
}

```

```

Step B2: receive(msg3) from A;
let (Nb') = (decrypt(msg3) with skB)
AND check_if(Nb' = Nb);
return to Needham_Schroeder().
}

```

Figure 3. Needham schroeder public key protocol.

On B side, 2 trace patterns are obtained—one showing the normal run of the protocol and the other showing the man-in-the-middle attack. No attacks are found when the protocol is verified for a single run. But with 2 or greater runs, attacks are generated.

In ProVerif, the protocol is run for an unbounded number of times. For checking the secrecy of nonces and keys in ProVerif, a random number is generated and it is encrypted using the nonce and broadcasted over a public channel. The results obtained are similar to those obtained using Scyther. But, there is no way of checking for the synchronization and agreement attacks on either the receiver or the sender side.

4.4. Andrew Secure RPC Protocol

4.4.1. Definition

This protocol is intended to distribute a new session key between two parties A and B. The protocol must guarantee the secrecy of the new shared key k . In every session, the value of k must be known only by the participants playing the roles of A and B. The protocol must guarantee the authenticity of k . In every session, on reception of message 4, A must be able to ensure that the key k in the message has been created by A in the same session. The final message contains $N'b$ which can be used in future messages as a handshake number [12].

The pseudocode of the protocol is shown in **Figure 4**.

```

Andrew_Secure_RPC(Kab)

/* nonce = random number,
id_X = identity of X,
A,B = the communicating parties,
Kab = symmetric key between A and B,
k = session key between A and B. */

{
  Call (Sender_A(Kab) | Receiver_B(Kab)).
  /* Call the functions in parallel */
}

Sender_A(Kab)
{
  StepA1: generate nonce Na;
  msg1 = (id_A, (sym_encrypt(Na) with Kab));
  send msg1 to B;
  call StepB1.
}

```

```

StepA2: receive msg2 from B;
let (Na'',Nb')=(sym_decrypt msg2 with Kab) AND
check_if(Na'' = Na+1);
msg3 = sym_encrypt(Ns'+1) with Kab;
send msg3 to B;
call StepB2.

StepA3: receive msg4 from A;
let (k,Nb1') = sym_decrypt(msg4) with Kab;
return to Andrew_Secure_RPC(). }

Receiver_B(Kab)
{
  StepB1: receive msg1 from A;
  let (id_A',part) = msg1;
  let (Na') = sym_decrypt(part) with Kab;
  generate nonce Nb;
  msg2 = sym_encrypt (Na'+1,Nb) with Kab;
  send msg2 to A;
  call StepA2.

  StepB2: receive msg3 from A;
  let (Nb'') = (sym_decrypt msg3 with Kab)
  AND
  check_if(Nb'' = Nb+1);
  generate nonce Nb1;
  generate session_key k;
  msg4 = sym_encrypt(k,Nb1) with Kab;
  send msg4;
  call StepA3.
}

```

Figure 4. Andrew secure RPC protocol.

4.4.2. Analysis

When this protocol is verified using Scyther, attacks are found on the initiator side and none are obtained on the receiver side. The major attack is the one in which the session key is compromised. This is a freshness attack on the protocol. In this, an intruder can replay an old message (the last message) and the party A has no way of knowing that this has come from B or some intruder. Thus, the intruder can establish a session with A using an older session key. Since the communication is not taking place in the proper order, there are attacks of synchronization and agreement as well. For the claim to check the secrecy of session key k on the initiator (A) side, scyther outputs saying the claim is “Falsified” and that there is “Exactly 1 attack”. The attack graph provides a complete flow diagram of the actions of the parties and the intruder.

ProVerif also provides us with a similar result. That is, it can also detect that the session key is not secret on the initiator side. The private free variable var is encrypted using the session key k and published over a public channel c , the output obtained is false *i.e.* the key is not secret on the initiator side. When a similar query is provided for the receiver side, ProVerif gives the output saying that session key is secret on the receiver side. In order to obtain a complete trace pattern, the parameter

“traceDisplay” can be set to “long”. This provides an entire description of how the attack is executed. In addition, when the parameter “attacker” is set to “passive”, we do not obtain any attacks showing that there are no passive attacks on the protocol. It can be verified that the symmetric key *Kab* is secret on both the sides.

4.5. Challenge Handshake Authentication Protocol

4.5.1. Definition

The Challenge Handshake Authentication Protocol (CHAP) uses a 3-way handshake to periodically verify the identity of the party. This is done upon initial link establishment, and may be repeated anytime after the link has been established. Once the link is established, the authenticator sends a challenge message to the party. The party responds with a value calculated using a one-way hash function. The authenticator checks the response against its own calculation of the expected hash value. If the values match, the authentication is acknowledged; otherwise the connection has to be terminated [13].

The pseudocode of the protocol is shown in **Figure 5**.

```

Challenge_Handshake(Kas)

/* challenge = a random value sent by server at irregular intervals,
id = identifier,
A = client,
S = server,
Kas = shared secret between S and A. */
{
  Call (Server_S(Kas) | Client_A(Kas)).
  /* Call the functions in parallel */
}

Server_S(Kas)
{
  StepS1: generate challenge Ns;
  msg1 = (Ns, id);
  send msg1 to A;
  call StepA1.

  StepS2: receive msg2 from A;
  check_if(hash(concat(Ns AND id AND
Kas)) = msg2);
  msg3 = sym_encrypt(id) with Kas;
  send msg3 to A;
  call StepA2.
}

Client_A(Kas)
{
  StepA1: receive msg1 from S;
  let (N's, id') = msg1;
  msg2 = hash (concat(N's AND id' AND Kas));
  send msg2 to S;
  call StepS2.
}

```

```

StepA2: receive msg3 from S;
  let (id'') = (sym_decrypt msg3 with Kas) AND
  check_if(id'' = id').
}

```

Figure 5. Challenge handshake authentication protocol.

4.5.2. Analysis

On analyzing this protocol with scyther, we see that the symmetric key (shared secret) is secret on both the sides, the server as well as the party which needs to be authenticated. But, there are synchronization and agreement attacks on A side but not on the server(S) side. These attacks are caused because the first message from the server is not encrypted. Thus, it can be captured by any intruder. But this does not lead to the shared secret being compromised as the message from the A party to the server is hashed using a one way hash function. Thus, even if the intruder knows the hash value and the hashing algorithm, there is no way to unhash the value and obtain the original message. A single trace pattern is obtained for the server side. For the A side, 2 patterns are obtained—one specifying the normal run of the protocol and the other giving details of how the communication can be disrupted.

When this protocol is verified using ProVerif, the output obtained shows that the symmetric key *Kas* cannot be compromised on either the client side or the server side. Thus, the communication is secure. In addition, when the parameter “attacker” is set to “passive”, no attacks are found suggesting that there are no passive attacks on this protocol.

4.6. Diffie-Hellman Key Exchange Protocol

4.6.1. Definition

Diffie–Hellman key exchange (D–H) is a cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher. There are two publicly known numbers—a prime number *p* and a primitive root of that prime, *g*. Each party then chooses a random number which is less than *p* and using modular arithmetic, the key is calculated. Thus, a key is exchanged between two or more parties over an insecure channel [14].

The pseudocode of the protocol is shown in **Figure 6**.

```

Diffie_Hellman()

/* nonce = random number,
A,B = the communicating parties,
p = a large prime number,
g = a primitive root of p. */

```

```

{
  Call (Sender_A() | Receiver_B()).
  /*Call the functions in parallel */
}

Sender_A()
{
  StepA1: select n0;

  /* 1 <= n0 < (p-1) */

  msg1 = (g^n0 mod p);
  send msg1 to B;
  call StepB1.

StepA2: receive msg2 from B;
  let (part2) = msg2;
  Akey = ((part2^n0)mod p);
  return to Diffie_Hellman().
}

Receiver_B()
{
  StepB1: receive msg1 from A;
  let (part1) = msg1;

```

```

select n1;

/* 1 <= n1 < (p-1) */

Bkey = ((part1^n1)mod p);
msg2 = (g^n1 mod p);
  send msg2 to A;
  call StepA2.
}

```

Figure 6. Diffie Hellman key exchange protocol.

4.6.2. Analysis

When this protocol is verified using ProVerif, we find that the generated key is not secret on the initiator side as well as the receiver side.

This is because of a man in the middle attack. An intruder is able to capture the public values from both the legitimate parties and send its own generated public value to both of them. Thus, an intruder is able to establish a session key with both the parties.

The Diffie Hellman key exchange cannot be modeled using Scyther. The reason is that there is no way to show

Table 1. Scyther and ProVerif characteristic comparison.

Characteristics of SCYTHER	Characteristics of PROVERIF
<ul style="list-style-type: none"> The protocol is modeled using the “spdl” language. It is possible to run the protocol for either bounded or unbounded number of sessions. Tool comes with its own graphical user interface. It generates the following possible outputs namely Property holds for n runs, Property is false and attack trace is shown, Property holds for all traces. Attack graphs are generated which give a visual flow graph of a trace and are self explanatory. All possible trace patterns are generated depicting protocol execution. The communicating parties need to be modeled as roles. It doesn’t provide any option to check for equality of different variables. Tool by its own discretion checks for secrecy of all possible variables, no explicit “claims” are necessary. The anticipated intruders along with the legitimate communicating parties have to be specified as agents. In case of protocols which may suffer from a freshness attack, we have to put a key compromise module in the code which specifies that a complete session has been captured and the intruder also knows the session key. There is no concept of channels. 	<ul style="list-style-type: none"> The protocol is modeled using horn clauses or pi calculus. Tool has to be run through command line interface. It generates the following possible outputs namely Property is true, Property is false and attack trace is generated, Property cannot be proven when false attack is found, Tool might not terminate. Step by step trace is generated explaining the run and attack. Trace is generated only for the property which is checked. The communicating parties need to be modeled as processes. Equality can be checked by using “if..then” or “let..in”. It checks only those attacks for which the “query” has been specified in the code. ProVerif does not require any such specification. No special code for a freshness attack needs to be given in ProVerif. Channels need to be specified for communication. It is possible to run the protocol only for an unbounded number of sessions.

the equivalence of 2 exponential operations. That is, a rule that states $(\exp(\exp(g,x),y) \bmod p)$ and $(\exp(\exp(g,y),x) \bmod p)$ are the same cannot be specified. Thus, it is not possible to handle such exponentiations which need equivalence conditions to be explicitly mentioned. Hence, the tool is not able to know that the keys that have to be generated on both sides are ideally the same.

5. Conclusions and Future Work

Based on the implementation and evaluation as described above, we summarize the comparative analysis of Scyther and ProVerif in the Table-I.

From this, it can be observed that applying formal methods to verify security protocols is an interesting and challenging research area. Using the tools Scyther and ProVerif, it is possible to model many security protocols in standard format, verify them and know the attacks they are susceptible to. We modeled six characteristic protocols and verified them using these tools. The tools vary in regards like their input language, manner in which the output is provided, the way in which traces of attacks are generated. Moreover, both the tools have a few limitations. Using these tools, it is easy to know what flaws these protocols suffer from so that the flaws can be rectified. Although verification using these tools does not ensure that the protocols once verified by these tools are flawless, still they provide a means to know many of the flaws easily. As a future work in this area, we plan to extend this comparative evaluation using other interesting protocols namely to have a better understanding of the differences in the capabilities of both. Our work can also be extended to model the same protocols using other tools to have a wider evaluation.

6. Acknowledgements

We are grateful to all those anonymous reviewers for their useful suggestions, to help give the paper the shape, it is now, in.

7. References

- [1] W. Stallings, "Cryptography and Network Security: Principles and Practices," 4th Edition, Pearson Education, ISBN-10: 0131873164 ISBN-13: 9780131873162, 2006.
- [2] D. Denning and G. Sacco, "Timestamps in Key Distribution Protocols," *Communications of the ACM*, Vol. 24, No. 8, 1981, pp. 533-536.
- [3] R. Needham and M. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," *Communications of the ACM*, Vol. 21, No. 12, 1978, pp. 993-999.
- [4] R. Needham and M. Schroeder, "Authentication Revisited," *ACM SIGOPS Operating Systems Review*, Vol. 21, No. 1, 1987, p. 7.
- [5] G. J. Holzmann, "Software Model Checking with SPIN," *Advances in Computers*, Vol. 65, 2005, pp. 78-109.
- [6] C. J. F. Cremers, "Scyther-Semantics and Verification of Security Protocols," Ph.D. Thesis, Eindhoven University of Technology, 2006.
- [7] B. Blanchet, "An Efficient Cryptographic Protocol Verifier Based on Prolog Rules," *Proceedings of the 14th IEEE Computer Security Foundations Workshop (CSFW)*, Cape Breton, IEEE Computer Society, 2009, pp. 82-96.
- [8] D. Song, "Athena: A New Efficient Automatic Checker for Security Protocol Analysis," *Proceedings of the 12th IEEE Computer Security Foundations Workshop (CSFW)*, IEEE Computer Society, 1999, pp. 192-202.
- [9] A. Armando et al., "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications," *Proceedings of Computer Aided Verification'05 (CAV)*, Vol. 3576 of Lecture Notes in Computer Science, Springer, 2005, pp. 281-285.
- [10] I. Kao, Lung and R. Chow, "An Efficient and Secure Authentication Protocol Using Uncertified Keys," *SIGOPS Operating Systems Review*, Vol. 29, No. 3, 1995, pp. 14-21.
- [11] Verified By Visa 3-D Secure Protocol. [Online] Available: <https://usa.visa.com/personal/security/vbv/index.html>, last retrieved on 2nd June 2010.
- [12] M. Satyanarayanan, "Integrating Security in a Large Distributed System," *ACM Transactions on Computer Systems*, Vol. 7, No. 3, 1989, pp. 247-280.
- [13] W. Simpson, "PPP Challenge Handshake Authentication Protocol (CHAP)," August 1996 <http://www.ietf.org/rfc/rfc1994.txt>, last retrieved on 2nd June 2010.
- [14] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, Vol. 22, No. 6, 1976, pp. 644-654.
- [15] C. J. Cremers, P. Lafourcade and P. Nadeau, "Comparing State Spaces in Automatic Security Protocol Analysis," *Formal to Practical Security: Papers Issued from the 2005-2008 French-Japanese Collaboration*, Springer-Verlag, 2009, pp. 70-94.
- [16] C. J. Cremers, "Unbounded Verification, Falsification, and Characterization of Security Protocols by Pattern Refinement," *CCS'08: Proceedings of the 15th ACM conference on Computer and communications security*, ACM Press, 2008, pp. 119-128.

Block Layering Approach in TAST Codes

Zahoor Ahmed, Jean Pierre Cances, Vahid Meghdadi

Université de Limoges - Ecole Nationale Supérieure d'Ingénieurs de Limoges (ENSIL)

XLIM-Dept. C2S2, UMR CNRS 6172 16, Rue Atlantis Parc ESTER-BP 6804-87068 Limoges cedex, France

E-mail: zahoor.ahmed@ensil.unilim.fr, cances@ensil.unilim.fr, meghdadi@ensil.unilim.fr

Received July 26, 2010; revised August 20, 2010; accepted September 21, 2010

Abstract

Threaded Algebraic Space Time (TAST) codes developed by Gamal *et al.* is a powerful class of space time codes in which different layers are combined and separated by appropriate Diophantine number ϕ . In this paper we introduce a technique of block layering in TAST codes, in which a series of layers (we call it Block layers) has more than one transmit antenna at the same time instant. As a result we use fewer layers (Diophantine numbers) for the four transmit antennas scheme, which enhances the coding gain of our proposed scheme. In each block layer we incorporate Alamouti's transmit diversity scheme which decreases the decoding complexity. The proposed code achieves a normalized rate of 2 symbol/s. Simulation result shows that this type of codes outperforms TAST codes in certain scenarios.

Keywords: TAST Code, Block Layer, Space Time Coding

1. Introduction

It is well known that wireless communications systems over Rayleigh fading channels can benefit from the simultaneous use of multiple antennas at both the transmitter and receiver to convey information either more reliably or at higher rates than would be possible for single antenna system. The remarkable paper of Alamouti [1] which is considered a benchmark in space time coding, is based on orthogonal design for two transmit antennas offer full diversity and simple linear maximum likelihood (ML) detectors that decouple the transmitted symbols.

Unfortunately, the Hurwitz-radon theorem showed that square complex linear processing orthogonal designs cannot achieve full diversity and full rate simultaneously for more than two antennas. Later on such type of proof has also been shown in [2]. In [3] Jaffarkhani *et al.* has generalized the scheme of orthogonal STBC codes construction for more than two transmit antennas by compromising either the diversity or coding gain. Some researchers have also introduced codes with higher rates and better performances by sacrificing the simplicity of ML decoding and thus orthogonality. In [4] a layering concept in STBC, called vertical Bell Lab layered space-time (V-BLAST), was introduced, but the main drawback of this code was its inflexibility with number of antennas.

Extending the work of layering concept of [4], H. Gamal *et al.* [5] introduced a new architecture in STBC codes,

known as Threaded Space Time (TST) codes. In this architecture, independent codes streams are distributed throughout the transmission resource array in different threads. Of course the efficient separation of individual layers from one another was the primary objectives in the design of such codes. The main drawback of this type of code is the complexity of ML decoder which rises exponentially with number of transmit antennas.

Threaded Algebraic codes [5] based on Diophantine approximation theory and number field were further generalized in [6,7] for arbitrary number of transmit and receive antennas, retaining full rate and maximum diversity. Such types of high rate STBC codes have also been constructed using division algebras [8,9].

In this paper we propose a technique of construction TAST codes within the framework of [6]. The proposed codes are flexible both in term of usage of antennas (at both ends) and Diophantine numbers. We use term AF TAST code for being flexible in term of antennas and DNF TAST code for being flexible in term of Diophantine numbers. As a result the DNF TAST code for four transmit antennas scheme provides higher coding gain and higher code rate retaining maximum diversity as that of original layered codes. This framework is based on TAST code with a slight modification in the definition of layer that in this scheme we may use more than one transmit antenna for transmitting same block or a series of layers

The rest of paper is organized as follow:

A brief review of previous work on TAST code is outlined in Section 2. In Section 3 we present the new approach of flexible TAST code construction in term of antennas' flexibility. In Section 4 we discuss flexible TAST codes in term of Diophantine numbers. The decoding is presented in Section 5 and finally Section 6 presents our conclusion.

2. Preliminaries

As our proposed framework is based on the threaded space time architecture [6], so for sake of completeness we review some notation from [6].

A layer in an $N_T \times T$ (where N_T denote the number of transmit antennas) transmission resource array is identified by an indexing set $l \subset I_{N_T} \times I_T$ where $I_{N_T} = \{1, 2, \dots, N_T\}$ and the t -th symbol interval on antenna a belongs to the layer if and only $(a, t) \in l$. This indexing set must satisfy the requirement that if $(a, t) \in l$, then either $t \neq t'$ or $a = a'$ (i.e., that a is a function of t). The definition will be clearer from **Table 1** given below, which depicts a view for four transmit antennas having four layers.

Where for layer l , $l = 1, \dots, n$ of the codeword, the set of matrix entries in positions are given by

$$(t, (l+t-1) \bmod(n)+1), \quad \text{for } k=1, \dots, n$$

With an arbitrary number of threads, the TAST codes are constructed by transmitting a scaled DAST code [10] in each thread, i.e.,

$$\mathbf{x}_l = \phi_l \mathbf{x}_l = \phi_l \mathbf{M}_l \mathbf{s}_l \quad (1)$$

is transmitted over thread l_l . Where \mathbf{x}_l are encoded symbols, \mathbf{M}_l is an $N_T \times N_T$ real or complex rotation matrix, $\mathbf{x}_l = \mathbf{M}_l \mathbf{s}_l$ are rotated complex information symbol vectors and ϕ_l , $l=1, \dots, L$ are the Diophantine numbers chosen to ensure full diversity and maximize the coding gain of the component codes. In [6] ϕ_l is given by

$$\phi_l = \phi^{(l-1)/N_T} \quad (2)$$

where $\phi = e^{i\lambda}$ ($\lambda \neq 0$) is an algebraic number

3. AF TAST Codes

In some communication systems (for example UMTS), the number of antennas varies among base stations and

mobile devices, so it is vital to design a flexible MIMO transmission scheme supporting various multi-element antennas. As a minimum requirement, the mobile station might only be informed about the number of transmit antennas at the base station. Based on its own number of receive antennas, it can then decide which decoding algorithm to apply. Conventional STBC codes offer great complexity in varying the number of receive/transmit antennas. The TAST codes [6] are flexible with respect to number of transmit/receive antennas. In this section we introduce a different and simple technique of flexible ST codes construction which are also flexible with respect to number of transmit/receive antennas and reducing decoding complexity.

We start with basic simple Alamouti code.

$$\mathbf{A}_1 = \phi_l \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \end{bmatrix} \quad (3)$$

$$\mathbf{A}_2 = \phi_l \begin{bmatrix} s_3 & -s_4^* \\ s_4 & s_3^* \end{bmatrix} \quad (4)$$

$$\mathbf{A}_3 = \phi_l \begin{bmatrix} s_5 & -s_6^* \\ s_6 & s_5^* \end{bmatrix} \quad (5)$$

$$\mathbf{A}_4 = \phi_l \begin{bmatrix} s_7 & -s_8^* \\ s_8 & s_7^* \end{bmatrix} \quad (6)$$

For $1 \leq l \leq L$ (L being the numbers of layers)

where ϕ_l is Diophantine number and it is not difficult to verify that taking any one matrix from (3) to (6) results a simple Alamouti codes as we know from (2) that $\phi_l = 1$.

As our proposed scheme is flexible with respect to number of transmit and receive antennas, so by simple reshuffle of (3) to (6) we get different structure of TAST codes for different set up of transmit/ receive antennas. Below is a body of a simple program that might be used for this purpose.

Let N_T , N_R , L , A , denote number of transmit antennas, number of receive antennas, number of layers, and number of Alamouti matrices (given in (3) to (6)), respectively.

Initialization, N_T , N_R

Condition (No. of transmit & receive antenna)

Select (value for L and A)

Process (build TAST codeword matrix with given no. of L , and N_T)

end

Note that for all the following structure of codes we

Table 1. Thread distribution.

1	4	3	2
2	1	4	3
3	2	1	4
4	3	2	1

consider Diophantine number ϕ_l same as in (2).

For case of $N_T = 2$, (Alamouti code) we simply take any one matrix from ((3) to (6)). For $N_T = 2$ and $N_R \geq 2$, we shall add any two matrices from ((3) to (6)) with a minor manipulation. To save space we avoid going in detail. Likewise for $N_T = 3$ and $N_R \geq 2$, we add any three matrices from ((3) to (6)) with a slight modification. In same way we can develop a code for $N_T = 4$. In next section we discuss one of such type of code for $N_T = 4$ and $L = 2$.

4. DNF TAST Code

In case of Diophantine numbers flexibility, the case is interesting for $N_T = N_R = 4$ and $L = 2$. Therefore in what follows, we discuss a case for $N_T = 4$ and $L = 2$, and at the end of this section we give the numeral representations for others set up as well.

The necessary condition of layering concept in [5] that the more than one antenna cannot transmit symbols from a given layer at a given time instant has been relaxed. A group of transmit antennas may now belong to a series of layers (for simplicity we call a series of layers as block) for a given symbol period.

A block layer is indexed by a set b , $b \subset b_{N_T} \times b_T$, $b = (w, t) \in \{1, 2, \dots, N_T\}$. Like TAST [6] and DAST [10] schemes, the idea is to map each block layer to a different subspace so that they are as far away from each other as possible. With the concept of block layers, the total number of layers becomes less and consequently a less number of Diophantine numbers are required which increases the coding gain. Also, real or complex rotated symbols are used to further increase the coding gain. In each block we use Alamouti's transmit diversity scheme that ensures simple decoding at the receiver.

Combining (3) to (6)

$$\begin{bmatrix} \phi_1(A_1) & \phi_2(A_2) \\ \phi_2(A_4) & \phi_1(A_3) \end{bmatrix} \quad (7)$$

Or more precisely

$$\begin{bmatrix} \phi_1 s_1 & -\phi_1 s_2^* & \phi_2 s_3 & -\phi_2 s_4^* \\ \phi_1 s_2 & \phi_1 s_1^* & \phi_2 s_4 & \phi_2 s_3^* \\ \phi_2 s_7 & -\phi_2 s_8^* & \phi_1 s_5 & -\phi_1 s_6^* \\ \phi_2 s_8 & \phi_2 s_7^* & \phi_1 s_6 & \phi_1 s_5^* \end{bmatrix} \quad (8)$$

It is straightforward to verify that the modified representation in (8) has the same property as the original Alamouti code. However, this modified representation clearly falls within the scope of the threaded coding framework.

In (8) ϕ_1 and ϕ_2 are two Diophantine numbers

and $[s_1, s_2, \dots, s_8]$ is the rotated information vector to be transmitted.

In matrix form the DNF-TAST code for $N_T = N_R = 4$ and $L = 2$ is given in (9) which uses 2^q PSK or QAM signal constellation, and has a rate of $R = 2q$. For TAST code we use the notation $\mathcal{T}_{N_T, L, R}$ while for flexible TAST code we use $\bar{\mathcal{T}}_{N_T, L, R}$, where the subscripts in both cases show the numbers of transmit antennas, number of layers, and symbols per channel use, respectively.

$$\bar{\mathcal{T}}_{4,2,4} = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \end{bmatrix} \quad (9)$$

The transmitted symbol \mathbf{x}_l corresponding to source information symbol s_l over l^{th} block layer is

$$\mathbf{x}_l(s_l) = \phi_l \mathbf{x}_l = \phi_l \mathbf{M}_l s_l, \quad l = 1, \dots, L$$

where L represents the total number of block layers and $\mathbf{x}_l = \mathbf{M}_l s_l$ are the rotated information symbol vectors. Here \mathbf{M}_l is an $N_T \times N_T$ real or complex rotation matrix built on an algebraic number field $\mathcal{Q}(\theta)$ with θ an algebraic number of degree n , and the numbers ϕ_l , $l = 1, \dots, L$ are the Diophantine numbers. Both for real and complex rotation matrices we use the matrices same as given in [6].

In general, one can use different rotation matrices in different blocks. A general and simple MATLAB program which generate rotation matrix \mathbf{M}_d of any dimension $d = 2^q$ on a number field $\mathcal{Q}(\cos 2\pi/8d)$ is given in [10].

$$\mathbf{M} = \text{sqrt}(2/d) * \cos(\pi / (4*d)) * (4 * [1 : d] - 1) * (2 * [1 : d] - 1); \quad (10)$$

To construct a rotation matrix \mathbf{M}_d of higher dimensions in d the following recursive approach can be used [11].

$$\mathbf{M}_d = \begin{bmatrix} \mathbf{M}_{d/2}^1 & -\mathbf{M}_{d/2}^2 \\ \mathbf{M}_{d/2}^2 & \mathbf{M}_{d/2}^1 \end{bmatrix} \quad (11)$$

where $\mathbf{M}_{d/2}^1$ is the optimal real rotation in dimension $d/2$ and $\mathbf{M}_{d/2}^2$ is an orthogonal transformation in dimension $d/2$. The Diophantine approximation intends to achieve full diversity and maximize the coding gain [6]. For a DNF-TAST code with L layers the Diophantine numbers are chosen same as (2) with L denoting number of block layers.

For a neat comparison for $N_T = 4$ and $L = 2$, we reproduce the code as given in [8] in (12). It is crystal clear that the performance of the code in (8) is much better

than in (12), as the former contain no zeros in transmission matrix.

$$\mathcal{T}_{4,2,2} = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix} \quad (12)$$

For $N_T = 3$ and $L = 2$, we can get flexible TAST code by deleting last row and adding last and second last columns in (12).

$$\bar{\mathcal{T}}_{3,2,2} = \begin{bmatrix} 1 & 0 & 2 \\ 2 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix} \quad (13)$$

For $N_T = 3$ and $L = 3$, we can get flexible TAST code by adding third thread on empty layer in (13).

$$\bar{\mathcal{T}}_{3,3,3} = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{bmatrix} \quad (14)$$

For $N_T = 2$ and $L = 2$, we get a code by deleting last row and adding last and second last columns in (13).

$$\bar{\mathcal{T}}_{2,2,2} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \quad (15)$$

5. Decoding

For the set up with one and three Diophantine numbers, we can use simple decoding schemes given in [1] and [6] respectively. Here we elaborate decoding scheme for our proposed code in (9).

The received signal can be written as

$$Y = H \bar{\mathcal{T}}_{N_T, L, R} + N \quad (16)$$

where H is the $N_R \times N_T$ complex Gaussian random channel matrix with element $h_{i,j}$, $i = 1, 2, \dots, N_R$ and $j = 1, 2, \dots, N_T$, and N is a complex Gaussian random noise vector.

Let

$$y = \text{vec}(Y^T) \quad (17)$$

arranges the matrix Y^T in one column vector by stacking its columns one after other, and let

$$y = [y_1, y_2, \dots, y_{N_R N_T}] \quad (18)$$

Simplifying equation (1) and (16), we get

$$y' = \mathbf{H}' \phi' \mathbf{M}' u + n \quad (19)$$

where

$$\mathbf{M}' = \begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A} & \mathbf{M} \end{bmatrix} \quad (20)$$

where \mathbf{A} is a $0_{4 \times 4}$ matrix, \mathbf{M}' , ϕ' and \mathbf{H}' are respectively rotation matrix, Diophantine matrix and the channel matrix given in (20), (21) and (22), and n is obtained by converting $\text{vec}(N^T)$ into column vector by stacking its columns one after other, and u is a vector carrying source information symbols.

$$\hat{\phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \quad (21)$$

where

$$\phi_1 = \begin{bmatrix} \phi_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \phi_1^* & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \phi_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \phi_1^* & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \phi_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \phi_1^* & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \phi_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \phi_1^* & 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$\phi_2 = \begin{bmatrix} \phi_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \phi_2^* & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \phi_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \phi_2^* & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \phi_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \phi_2^* & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \phi_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \phi_2^* & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{H}' = [h_1 \ h_2] \quad (22)$$

where

$$h_1 = \left(\begin{bmatrix} h_{ij} & 0 & h_{ij+1} & 0 & 0 & 0 & 0 & 0 \\ 0 & h_{ij+1}^* & 0 & -h_{ij}^* & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_{ij+2} & 0 & h_{ij+3} & 0 \\ 0 & 0 & 0 & 0 & 0 & h_{ij+3}^* & 0 & -h_{ij+3}^* \end{bmatrix} \right)_{i=1,2,3,4}^{j=1}$$

and

$$h_2 = \left(\begin{bmatrix} h_{ij-1} & 0 & h_{ij} & 0 & 0 & 0 & 0 & 0 \\ 0 & h_{ij}^* & 0 & -h_{ij-1}^* & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_{ij-3} & 0 & h_{ij-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & h_{ij-2}^* & 0 & -h_{ij-3}^* \end{bmatrix} \right)_{i=1,2,3,4}^{j=4}$$

Note that h_1 and h_2 are stacked into column for different values of i .

The simulation results given in **Figure 1** confirm our mathematical analysis for obtaining better performances of our proposed code in (9) over his brethren codes with $L = 2$ and 4. When comparing with the code when $L = 2$ as given in (12), our proposed codes absolutely, however in case when $L = 4$, our code outperform at low SNR. Due to the hardware constraint we could not carried out simulation for large no of antennas but we intelligently guess that our code gains better performance over the both codes for large no of antennas.

6. Conclusions

TAST codes with different number of transmit/ receive antennas and Diophantine numbers have been proposed. A code having four transmit antennas and two layers is discussed which attains a better performance as compared to same class of code having four layers in certain scenarios. For four receive antennas our proposed code outperform TAST code at low SNR, but for higher SNR TAST code works better. Due to limitation of hardware we could not simulate for higher number of receive antennas but we guess intelligently that increasing the number of receive antennas may enhance the performance of our proposed code. In addition ML decoding is

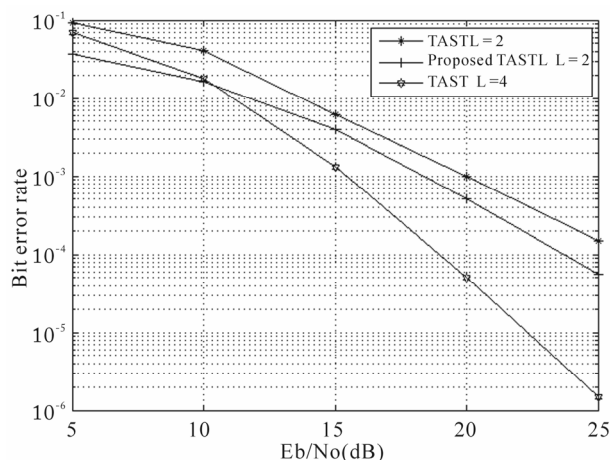


Figure 1. Comparison of different class of TAST codes.

another positive point of our scheme.

7. References

- [1] S. M. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communication," *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 8, 1998, pp. 1451-1458.
- [2] X. -B. Liang and X. -G. Xia, "On the Nonexistence of Rate-One Generalized Complex Orthogonal Designs," *IEEE Transactions on Information Theory*, Vol. 49, No. 11, 2003, pp. 2984-2988.
- [3] V. Tarokh, H. Jafarkhani and A. R. Calderbank, "Space Time Block Codes from Orthogonal Designs," *IEEE Transactions on Information Theory*, Vol. 45 No. 5, 1999, pp. 1456-1467.
- [4] G. J. Foschini, "Layered Space Time Architecture or Wireless Communication in a Fading Environment when Using Multiple Antennas," *Bell Labs Technical Journal*, Vol. 1, No. 2, 1996, pp. 41-59.
- [5] H. Gamal and A. R. Hammon, "A New Approach to Layered Space Time Coding and Signal Processing," *IEEE Transactions on Information Theory*, Vol. 47, No. 6, 2001, pp 2321-2334.
- [6] H. E. Gamal and M. O. Deman, "Universal Space Time Coding," *IEEE Transactions on Information Theory*, Vol. 49, No. 5, 2003, pp. 1097-1119.
- [7] M. O. Damen, A. Tewfik and J.-C. Belfiore, "A Construction of a Space-Time Code Based on Number Theory," *IEEE Transactions on Information Theory*, Vol. 48, No. 3, 2002, pp. 753-760.
- [8] F. Oggier, J. Belfiore and E. Viterbo, "Cyclic Division Algebras: A Tool for Space Time Coding," Boston, Delft, 2007.
- [9] B. A. Sethuraman, B. S. Rajan and V. Shashidhar, "Full-Diversity, High-Rate Space Time Block Codes from Division Algebras," *IEEE Transactions on Information Theory*, Vol. 49, No. 10, 2003, pp. 2596-2616.
- [10] M. O. Damen, K. A. Meraim and J.-C. Belfiore, "Diagonal Algebraic Space-Time Block Codes," *IEEE Transactions on Information Theory*, Vol. 48, No. 3, 2002, pp. 628-636.
- [11] J. Boutros and E. Viterbo, "Signal Space Diversity: A Power and Bandwidth Efficient Diversity Technique for the Rayleigh Fading Channel," *IEEE Transactions on Information Theory*, Vol. 44, No. 4, 1998, pp. 1453-1467.

Winning Strategies and Complexity of Nim-Type Computer Game on Plane^{*}

Boris S. Verkhovsky

Computer Science Department, College of Computing Sciences
New Jersey Institute of Technology, Newark, USA

E-mail: verb@njit.edu

Received June 10, 2010; revised July 17, 2010; accepted August 22, 2010

Abstract

A Nim-type computer game of strategy on plane is described in this paper. It is demonstrated that winning strategies of this two-person game are determined by a system of equations with two unknown integer sequences. Properties of winning points/states are discussed and an $O(\log \log n)$ algorithm for the winning states is provided. Two varieties of the Game are also introduced and their winning strategies are analyzed.

Keywords: Nim-type Game, Two-person Strategy Game, Winning Strategies, Newton Algorithm, Fibonacci Numbers

1. Introduction

A Nim game is probably one of the most ancient of all known games. There are several varieties of Nim: categorical games in which no draw is possible; futile games which permit a tie (draw); Grundy's game is a special type of Nim. The game is played by the following rules: given a heap of size n , two players alternately select a sub-heap and divide it into two unequal parts. A player loses if he or she cannot make a legal move. The *Misere* form of Nim is a version in which the player taking the last piece is the loser, [1].

In Fibonacci Nim, two players deal with a pile of n stones, where $n > 1$. The first player may remove any number of stones, provided that at least one stone is left. Players alternate moves under the condition that if one player removed x stones, then another one may remove at most $2x$ stones. Some of them are described in [1-5].

Several years ago the author of this paper introduced a Nim game with a heap of N stones, where each player is allowed to take at most m stones, provided that he/she does not repeat the last move of her/his opponent ("do not be a copycat"). The player taking the last stone is the winner. However, a player loses if he/she cannot make a feasible move. Winning strategies for an arbitrary $m > 1$ were provided by the author of this paper and implemented in [6] and [7] by his graduate students.

In the late 1980's the author also introduced a variety of the Nim-game that is discussed in this paper. In the paper we study properties of winning points, provide an algorithm for direct computation of winning points and analyze its complexity. It is demonstrated that the algorithm has $O(\log \log n)$ time complexity and does not require any storage, save a couple of numbers that are pre-computed at the beginning of the game. Preliminary results of this paper are published in [15].

2. Two-player Game on Plane

1) The Game starts after *two* distinct non-negative integers (S_0, L_0) are selected randomly; here.

$$1 \leq p \leq S_0 < qS_0 \leq L_0; S := S_0; L := L_0; \quad (1)$$

Remark 1: In the following discussion (S, L) is a point on a two-dimensional plane with integer coordinates; all further points are located in the positive quadrant of the plane; p and q determine a "level" of the Game. It is assumed that $0 \leq S < L$ holds, otherwise we swap the coordinates.

2) Three types of moves that allowed are: *horizontal, vertical and diagonal*.

The players on their move may decrease either

- The first coordinate on an integer t , $(S, L) \rightarrow (S - t, L)$, {horizontal move, *h-move*, for short} or
- The second coordinate on an integer u ,

^{*}© Boris S. Verkhovsky April, 2001

$(S, L) \rightarrow (S, L - u)$, {vertical move, *v-move*, for short} or
 c). Both coordinates on the *same* integer x ,
 $(S, L) \rightarrow (S - x, L - x)$, {diagonal move, *d-move*, for short};

The first player that reaches (0,0)-point on her/his move is the *winner* of the Game. An analogous Nim game was introduced by Wythoff [14]. Whytoff' game is played with two heaps of counters: a player is allowed to take any number from either heap or the same number from both. The player taking the last counter wins.

As in every two-person game with complete information, this Game has a winning strategy for one of the players [8-10]. In the following discussion we consider that a *Human (Hugo)* plays against a *Computer (Cora)*.

All points can be divided onto two classes: *winning* points for *Cora* and *losing* points for *Cora*. It is clear that a winning point for *Cora* is a losing point for *Hugo*, and vice versa.

Definition 1: We will say that the Game is in a *winning state* if after *Cora's* move it is in a winning point.

Let's denote *Cora's* winning states as w_n , for $n \geq 0$. Here $w_0 = (0, 0)$.

Example 1: The $w_1 = (1, 2)$ point is a *Cora's* winning point, because *Hugo* cannot reach $w_0 = (0, 0)$ point on his move. The $w_2 = (3, 5)$ is another winning point for *Cora*, because on his move *Hugo* cannot reach either $w_0 = (0, 0)$ point or $w_1 = (1, 2)$ point.

On the other hand, after any move by *Hugo*, *Cora* reaches either (0,0) or (1,2).

3. Seven Properties of Winning Points

P1. It is easy to see that if (c, d) is a winning point, then (d, c) is also a winning point.

P2. With exception of the (0,0)-point, in all other winning points $c \neq d$. Indeed, let (c, c) be a winning point for *Cora*. Then *Hugo* can reach the (0,0)-point using a *diagonal* move, {via subtracting the same $y = c$ from both coordinates}.

P3. If the Game is in a winning point w after *Cora's* move, then there is no move by which *Hugo* can reach another winning state w' . On the other hand, if the Game is in a losing point l , then there exists at least one move that transforms the Game into a winning state. For example, if after *Hugo's* move the Game is in the state (7, 9), then there are *two* winning moves for *Cora*: (4,7) and (3,5).

In general, let W be a set of all winning points and L be a set of all losing points. Then after one move the Game is transformed from W to L . However, if the Game is in L , then there exists at least one move that transforms the game into W . Formally it means that, if $(S, L) \in W$, then for any positive integer u , $(S - u, L) \notin W$ or

$(S, L - u) \notin W$ or $(S - u, L - u) \notin W$.

P4. Proposition 1: There are *no* two winning points $w_i = (c_i, d_i)$ and $w_k = (c_k, d_k)$ such that

$$d_i - c_i = d_k - c_k = r \quad (2)$$

where r is an integer.

Proof: Let's assume that for $i < k$ Equation (2) holds and $v := d_k - d_i$. Then after *Cora's* move w_k *Hugo* can reach w_i via a *diagonal* move, i.e., by subtracting from both coordinates the same integer v .

P5. Let w_i and w_k be two distinct winning points. Then all c_i, d_i, c_k, d_k are distinct integers, otherwise *Hugo* would be able to transform the Game into another winning state.

P6. Proposition 2: Let $(S, L) \in W$. Then for every $n = 1, 2, \dots$ holds

$$d_n - c_n = n. \quad (3)$$

Proof: Let's assume that there exists at least one winning point (c_m, d_m) , for which $d_m - c_m \neq m$, and m is the smallest integer; and let $s := d_m - c_m$.

Consider (c_s, d_s) : if $s < m$, then $d_s - c_s = s$ since by assumption m is the smallest.

Therefore, $d_m - c_m = d_s - c_s = s$,

or $d_m - d_s = c_m - c_s$.

Let $z := d_m - d_s = c_m - c_s$. Then there is a *diagonal* move that transforms one winning (c_m, d_m) point into another winning point (c_s, d_s) , which contradicts with the definition of a winning point. Therefore, $s \leq m - 1$ is impossible.

Let's now assume that $s \geq m + 1$.

Observe that $d_m - 1 \neq d_k$, otherwise for $k \leq m - 1$ $c_k \geq c_m$.

Consider *Hugo's* move $(c_m, d_m - 1)$, where $d_m - 1 - c_m \geq m$. But in this case *Cora* cannot make either a horizontal move or a diagonal move that transforms the Game into a winning state. The latter implies that $(c_m, d_m - 1)$ is a winning state, which in its turn contradicts the earlier assumption that (c_m, d_m) is the winning state. Q.E.D.

P7. Theorem 1: {Fundamental property of the winning points}:

a) Let $1 \leq x_0 \leq 2$; and for all $k \geq 0$

$$x_{k+1} := (x_k^2 + 1) / (2x_k - 1); \quad (4)$$

b) Let $G = \lim_{k \rightarrow \infty} x_k$;

c) (S, L) is a winning point if

$$S = \lfloor (L - S)G \rfloor \quad (5)$$

For the sake of simplicity of further discussion, we assume that in every point (c,d) $c < d$.

4. Game in Progress: An Example

Let $(S_0, L_0) = (29, 51)$ be a randomly generated initial point; *Hugo* makes the 1st move; in *italics* are shown *Hugo's* moves; in **bold** are *Cora's* winning points **Table 1**.

5. System of Equations with Infinite Sequences

Proposition 3: 1). Let $A := \{a_n\}$; $B := \{b_n\}$ be monotone increasing sequences of positive integers; here $a_1 = 1$ and $n = 1, 2, \dots$;

2). Let the sequences A and B satisfy the following system of equations:

$$B \cap A = \emptyset \quad (6)$$

$$B - A = N \quad (7)$$

$$B \cup A = N \quad (8)$$

where in (7) $B - A = \{b_n - a_n\}$, i.e. $B - A$ is a sequence of pair-wise differences of corresponding elements of B and A , and N is the set of all natural numbers $\{1, 2, 3, \dots\}$. Then the system of Equations (6)-(8) with unknown sequences A and B has a solution.

Proof {by induction}: The following algorithm is a constructive proof that a solution of (6)-(8) exists. Indeed, the sequences $A = \{a_n\}$; $B = \{b_n\}$ can be iteratively generated using an analogue of the Sieve of Eratosthenes:

StepL1: $(a_1, b_1) := (1, 2)$;

StepL2: Let $A_{k-1} := \{a_1, a_2, \dots, a_{k-1}\}$;

$B_{k-1} := \{b_1, b_2, \dots, b_{k-1}\}$ be sequences such that for every $k < n$ the following conditions hold: $a_1 < a_2 < \dots < a_{k-1}$;

$$b_1 < b_2 < \dots < b_{k-1} \quad (9)$$

$$A_{k-1} \cap B_{k-1} = \emptyset \quad (10)$$

and for every $1 \leq i \leq k-1$

$$b_i - a_i = i \quad (11)$$

StepL3: Let $J = \{j: b_j \geq a_{n-1} + 1\}$;

StepL4: Compute an integer $u := \min x$, where $x > a_{n-1}$ and for all $i \in J$ $x \neq b_i$;

StepL5: $a_n := u$;

StepL6: $b_n := a_n + n$.

Then the conditions (9)-(11) also hold for $k = n$. Q.E.D.

Applying the Steps L1-L6, we sequentially generate the winning points

$W = \{(1, 2); (3, 5); (4, 7); (6, 10); (8, 13); (9, 15); (11, 18); (12, 20); (14, 23); (16, 26); (17, 28); (19, 31); (21, 34); (22, 36); (24, 39); (25, 41); (27, 44); \dots\}$,

i.e., $(a_{17}, b_{17}) = (27, 44)$.

Therefore from the StepL3 $J = \{11, 12, 13, 14, 15, 16, 17\}$, and $u = 29$.

Then $(a_{18}, b_{18}) = (29, 47)$.

6. Alternative Formulation of L1-L6 Algorithm

W1: $a_1 := 1$; $c_1 := 2$; $j_1 := 1$;

W2: **for** $n=1, 2, \dots$ **do** $a_{n+1} := c_n + 1$;

W3: **if** $a_{n+1} < a_{j_n+1} + n$

then $c_{n+1} := c_n + 1$; $j_{n+1} := j_n$;

else $c_{n+1} := c_n + 2$; $j_{n+1} := j_n + 1$;

Here, j_n stands for the largest index k of b_k that was used in $\{a_n\}$, and c_n stands for the largest number of the set $\{1, 2, \dots, k\}$ which we cover for a_j and b_j for $j \leq n$ [11].

7. Sequences A, B and Winning Points

Theorem 2: For every integer $n \geq 1$

$$w_n := (a_n, b_n) \quad (12)$$

Proof: The following sequence of steps is a constructive proof of Theorem 1. Indeed, let

$$m := L - S \quad (13)$$

T1. If $S = a_m$, then by (3), (7) and (13) $b_m = L$; {*Hugo* is now in the winning point}.

T2. If $S > a_m$, then *Cora* selects $y := S - a_m$; $S := S - y$ and $L := L - y$; since $S > a_m$ implies that $L > b_m$. Indeed $L = S + m > a_m + m = b_m$.

T3. {If $S < a_m$, then *Cora* finds either an index $k < m$ such that $a_k = S$ or an index $i < m$ such that $b_i = S$ }.

T3a. If there exists an integer $k < m$ such that $a_k = S$, then we select $L := b_k$; {both $S < a_m$ and $a_k = S$ imply that $k < m$ and $L > b_k$. Indeed, an assumption that $k \geq m$ leads to a contradiction, because $m \leq k$ implies that $S < a_m \leq a_k$, but $a_k = S$ }.

Table1

Player	<i>Hugo</i>	Cora	<i>Hugo</i>	Cora	<i>Hugo</i>	Cora
Examples of moves	(23, 51)	(14, 23)	(6, 15)	(6, 10)	(4, 6)	(3, 5)
Type of move	<i>h-move</i>	v-move	<i>d-move</i>	v-move	<i>v-move</i>	d-move

T3b. If there exists an integer $i < t$ such that $b_i = S$, then we assign $L := S$; $S := a_i$; {since $b_i = S$ implies that $a_i < L$: $a_i < b_i = S < L$ }, [6].

8. Iterative Algorithm and its Complexity

In applications for computer games, an iterative computation of a_n and b_n for a large n is time consuming, since its time complexity $T(n)$ and space complexities $S(n)$ are both of order $O(n)$. For instance, if $n = 10^{12}$, then we need to generate and store one trillion pairs of integers. A brief analysis shows that this is well beyond of current size of memory for PC. A more efficient algorithm is described below.

9. Direct Computation of a_n and b_n

To decrease the complexity of computation of a_n and b_n and avoid excessive storage, let's find a closed-form expression for $a_n := v(n)$. Then from (11)

$$b_n := v(n) + n \quad (14)$$

Conjecture 1: (properties of winning points):

C1. $a_m / m = z + o(m)$;

$$\text{and } \lim_{m \rightarrow \infty} a_m / m = z \quad (15)$$

where z is a constant.

C2. For every integer $n \geq 1$

$$a_n = \lfloor nz \rfloor \quad (16)$$

The property (16) and the asymptotic behavior (15) are observed in numerous computer experiments.

Conjecture 2: For large n

$$b_n / a_n = z + o(n) \quad (17)$$

Remark 2: The Conjecture 2 is also based on extensive computer experiments.

Theorem 3: Conjecture 1 implies that z is an irrational number.

Proof: An assumption that z is a rational number leads to contradiction. Assume that $z = q/s$, where both q and s are relatively prime integers. Then there exists an infinite number of pairs a_n and b_r such that $a_n = b_r$. Indeed, select

$$n := (q+s)st \text{ and } r := qst \quad (18)$$

Then for the integer $t = 1, 2, 3, \dots$ it follows from (16) and (14) that

$$a_n = (q+s)qt \quad (19)$$

$$\text{and } b_r = q^2t + qst, \quad (20)$$

which is a violation of the conditions (6) and (10).

Conjecture 3: $z = g+1$, where g is a golden ratio,

$$\text{i.e., } g = (\sqrt{5} - 1) / 2 \quad (21)$$

The property (21) is observed in numerous computer experiments. It plausibility follows from the following:

Since for a large n $a_n = nz + o(n)$

$$\text{and } b_n = (z+1)n + o(n) \quad (22)$$

then it follows from (17) and (22) that

$$b_n / a_n \approx (z+1)/z \approx z \quad (23)$$

Then for large n 's, z is a positive solution of the equation.

$$x^2 - x - 1 = 0 \quad (24)$$

i.e., $z \approx (\sqrt{5} + 1)/2$, {the value of the golden ratio + 1}.

Let the Game be in the state (S, L) after *Hugo's* move and let

$$m := L - S \quad (13).$$

Then the Game is implicitly in one of five states {where by convention holds that $S < L$ }:

A. $(S, L) = (a_m, b_m)$, {the game is in a winning state for *Hugo*};

B. $(S, L) = (a_i, a_j)$, where $i < j$;

C. $(S, L) = (a_i, b_j)$; where either $i < j$ or $i > j$;

D. $(S, L) = (b_i, a_j)$, $i < j$;

E. $(S, L) = (b_i, b_j)$, $i < j$.

However, from the condition (11) alone we do not know yet in which of the states A, B, C, D or E the Game is.

10. Algorithm for Winning Points (AWP)

A1: Let $m := L - S$;

A2: Using (16) and (21), compute a_m ;

if $a_m = S$ **then** by (11) $b_m = L$; {*Hugo* is now in the winning state w_m } };

A3: **if** $S > a_m$ **then do** $y := S - a_m$; $S := S - y$ and $L := L - y$;

A4: **if** there exists an integer $k < m$ such that $a_k = S$ **then** $L := b_k$;

else find an integer $i < m$ such that $b_i = S$; $L := S$; $S := a_i$.

11. Validation of AWP

V1. In A3, $S > a_m$ implies that $L > b_m$.

Then $L = S + m > a_m + m = b_m$;

V2. If $S < a_m$, then there exists either an integer $k < m$ such that $a_k = S$ or an integer $i < m$ such that $b_i = S$;

V3. Both $S < a_m$ and $a_k = S$ imply that $k < m$ and $L > b_k$. Indeed, an assumption that $k \geq m$ leads to a contradiction, because $m \leq k$ implies that $S < a_m \leq a_k$, but $a_k = S$;

V4. In A4, $b_i = S$ implies that $a_i < L$. Hence $a_i < b_i = S < L$.

Example 2 {case $S > a_m$ }: Let the Game be in the state $(S, L) = (19, 26)$ after the *Hugo's* move.

Because $m = 26 - 19 = 7$, compute $a_7 = \lfloor 7(g+1) \rfloor = 11$ and $b_7 = 18$.

Since $11 < 19$ and $18 < 26$, then *Cora* moves

$S := S - m = S - 7$ and $L := L - m = L - 7$;

Example 3 {case $S < a_m$ }: Let now after *Hugo's* move $(S, L) = (15, 32)$.

Since $m = 32 - 15 = 17$, compute $a_{17} = \lfloor 17(g+1) \rfloor = 27$.

Since $15 < a_{17}$, but $15 = b_6$, then *Cora's* move is $L := 15$ and $S := L - 6 = 9$.

Example 4: {case $S < a_m$ }: Let $(S, L) = (14, 29)$ after *Hugo's* move.

Since $m = 15$, compute $a_{15} = \lfloor 15(g+1) \rfloor = 24$.

Since $a_{15} > 14$, but $14 = a_9$, then *Cora* moves

$L := 29 - 6 = b_9 = 23$.

Example 5: {case $a_m = S$ }: Then $b_m = L$, and the game is in the winning state for *Hugo*.

12. Fibonacci Properties of Winning Points

1. If n is an *odd* Fibonacci number, i.e., if $n = F_{2k-1}$, then

$$a_n = F_{2k} \quad (25)$$

2. If n is an *even* Fibonacci number, i.e., if $n = F_{2k}$, then

$$a_n = F_{2k+1} - 1 \quad (26)$$

Indeed, $a_{F_3} = 3$; $a_{F_5} = 8$; $a_{F_7} = 21$; $a_{F_9} = 55$;

But $a_{F_2} = 1$; $a_{F_4} = 4$; $a_{F_6} = 12$; $a_{F_8} = 33$.

13. Solution of Equation with Unknown Index

On the step A4 of the algorithm we must solve either equation $a_k = S$ or $b_i = S$ in order to respectively determine the indices k or i . In order to determine the indices we must solve either the equation

$$a_k = S \quad (27)$$

or the equation

$$b_i = S \quad (28)$$

This can be done by using (16)

$$a_k = \lfloor k(g+1) \rfloor = S \quad (29)$$

I 1) Find the smallest integer k^* satisfying the inequality $kg > S$; if $\lfloor k^*g \rfloor = S$ then

$$k = k^*; a_{k^*} = S; \quad (30)$$

I 2) If i^* is the smallest integer satisfying the ine-

quality $i^*(g+1) > S$
then

$$i = i^* \text{ and } b_{i^*} = S \quad (31)$$

If (16) has an integer solution, then from (29) we find the smallest integer $k = k^*$ satisfying the inequality

$$k^*(g+1) > S \quad (32)$$

Otherwise, {if $\lfloor k(g+1) \rfloor \neq S$ } we solve the equation $b_i = S$.

Then from (14) and (29)

$$b_i = \lfloor i(g+1) \rfloor + i = S \quad (33)$$

Example 6: Find an integer index k such that $a_k = 102$. Then $k^* = 64$ is the smallest integer for which holds

$$k^* \geq 102/(g+1) \text{ \{see (29)\}.$$

14. Required Accuracy for g

It is assumed that in the *Examples* 3-6 and 8 we know the exact value of an irrational number g . However, to find an integer solution of (17) for an arbitrary large index k or i we must compute g with a high precision. Let

$$g = d_1/10 + d_2/10^2 + \dots + d_n/10^n + \dots \quad (34)$$

where d_i is the i -th decimal digit of g
and

$$g(t) := \lfloor 10^t g \rfloor / 10^t = d_1/10 + d_2/10^2 + \dots + d_t/10^t \quad (35)$$

i.e., $g(t)$ contains only the first t decimal digits of g .

Theorem 4: Let $n \leq 10^k$. $a_n = S$ (36)

Then for all $t \geq k$ also holds that

$$a_n^{(t)} := \lfloor ng(t) \rfloor = S. \quad (37)$$

15. $O(\log \log n)$ Time Complexity for Winning Strategies

It is easy to verify that a positive root of (24) can be computed using a Newton iterative process

$$x_{r+1} := (x_r^2 + 1) / (2x_r - 1),$$

Where

$$x_0 := 1.618 \quad (38)$$

The process (38) has the following properties:

a). It converges to $(1 + \sqrt{5})/2$, i.e., for large r

$$x_r = (1 + \sqrt{5})/2 + \varepsilon_r, \quad (39)$$

where ε_r is a degree of accuracy (error) after r iterations.

b). The error ε_r satisfies the inequality

$$\varepsilon_r \leq \varepsilon_0^r = |x_0 - g|^{2^r},$$

i.e., it has a quadratic rate of convergence, and

$$\varepsilon_0 < 0.001 = 10^{-3}, [12]. \quad (40)$$

Then from the inequality $10^{-3 \times 2^r} \leq 10^{-k}$ we derive that

$$3 \times 2^r \geq k \quad (41)$$

Thus

$$r \geq \lceil \log_2(k/3) \rceil \geq \lceil \log_2(\lceil \log_{10} n \rceil / 3) \rceil \approx \lceil \log_{10} \log_{10} n \rceil \quad (42)$$

The inequalities (42) are derived from (36), (40) and (41). Then from analysis of (37) it follows that the time complexity $T(n)$ for solution of (16) is equal

$$T(n) = O(\log \log n).$$

The **Table 2** shows how many Newton iterations $r(n)$ are required to compute a_n as a function of n .

In addition, we do not need to store any winning points. Instead, as it is demonstrated below, only a *single* real value of g_* must be stored.

However, $\lfloor 64(g+1) \rfloor = 103 \neq 102$. Hence the equation $a_k = 102$ does not have a solution. On the other hand, $b_i = 102$ does have a solution. Indeed, from (33) it follows that $i \geq 102/(g+2) = 38.961$, i.e., $i^* = 39$. And finally $\lfloor 39(g+2) \rfloor = 102$.

16. Solution of Index Equations Revisited

R0.1. Let $S := s; L := l$; where both integers (s, l) are generated randomly at the beginning of the Game; let $t := L - S$;

R0.2. $r := \lceil \log t \rceil$; using the iterative process (36), compute

$$x_r; \quad (43)$$

R0.3. Let

$$g_* := x_r; \quad (44)$$

{during the entire Game use g_* as an approximation of g in the Equations (29) or (33)};

R1. Find the smallest integer k^* satisfying the inequality

$$k g_* > S \quad (45)$$

if $\lfloor k^* g_* \rfloor = S$

$$\text{then } k = k^*; a_{k^*} = S; \quad (46)$$

R2. If i^* is the smallest integer satisfying the inequality

$$i^* (g_* + 1) > S \quad (47)$$

Table 2. Logarithmic growth of $r(n)$.

$n = 10^t$	[0,3]	[4,6]	[7,12]	[13,24]
$r(n)$	0	1	2	3

$$\text{then } i = i^* \text{ and } b_{i^*} = S \quad (48)$$

Example 7: Let at the beginning of the Game $s := 2,718,282$ and $l := 3,141,593$.

Then $m := L - S = l - s = 323,311 < 10^6$. From the inequality $6 \geq 3 \times 2^r$, {see (40) and (41)}, it follows that $r = 1$. Hence, only one iteration of (38) is necessary to find g_* with required accuracy.

17. The Algorithm

It is assumed that *Hugo* makes the first move by randomly generating positive integers S_0 and L_0 such that

$$L_0 \geq (e-1)S_0 \geq Q \quad (49)$$

where e is Euler number, {see Remark3 below};

V: $m := L - S$;

if $m = 0$ **then** $z := S; S := S - z; L := L - z$; {end of the Game: *Cora* is the winner};

else

$$t := \lceil \log_{10} m \rceil;$$

$$r := \lceil \log_2(t/3) \rceil; x_0 := 1.618;$$

for k **from** 0 **to** $r-1$

do

$$x_{k+1} := (x_k^2 + 1) / (2x_k - 1);$$

$$G := x_r; a_m := \lfloor Gm \rfloor;$$

if $S = a_m$ **then** the Game is already in the winning state for *Hugo*;

{Nevertheless *Cora* might decide to continue the Game hoping that *Hugo* will make a mistake, i.e., he will “miss the point”};

if $L > 3$ **then** with $prob = 1/2$ $c := 1$ **or** 2 ; $L := L - c$;

goto V;

else if $S > a_m$

then $z := S - a_m; L := L - z; S := S - z$; **goto** V; **else**

$$k := \lceil S / G \rceil; \quad (50)$$

$\lfloor kG \rfloor = S$ **then**

$$y := m - k; L := L - y; \quad (51)$$

else $i := \lceil S / (G+1) \rceil; a_i := \lfloor Gi \rfloor$;

$u := L - a_i; temp := S; S := L - u; L := temp$; **goto** V.

Remark 3: In order to assure that the first randomly generated point is not a winning point, it is sufficient to select such S_0 and L_0 that

$$S_0 \neq \lfloor (L_0 - S_0)(g+1) \rfloor \quad (52)$$

That is guaranteed by (47) and (15), since

$$L_0 \geq 1.718S_0 \neq 1.618S_0.$$

18. Randomization

Let n_a and n_b be the number of integers on interval $[1, M]$ such that $1 \leq a_k \leq M$ and $1 \leq b_k \leq M$ respectively, i.e., $n_a + n_b = M$.

Then

$$n_a(M) \approx M / (g+1)$$

$$\text{and} \quad n_b(M) \approx M / (g+1)^2 \quad (53)$$

Hence, if a pair of integers (S, L) is generated randomly, then it is more likely that they will be elements of the sequence A , than the sequence B .

Remark 4: The sequence of the operations (50) and (51) in the Algorithm is based on the observation that for every M , $n_a(M) > n_b(M)$.

That is why on the A4 we first check whether there is a solution of $a_k = S$ and only then whether there is a solution of $b_i = S$. This sequence of verifications decreases the average complexity of the algorithm. Another approach is to randomize the sequence of these operators: Namely, with the probability $g = 0.618$ to execute (50) and then, if necessary, to execute (51). And with the probability $g = 0.382$ to execute (51) and only then, if necessary, to execute (50).

Example 8: If $M = 50$, then $n_a(50) = 31$ and $n_b(50) = 19$. Thus, if u is an arbitrary selected integer on the interval $[1, 50]$, then with probability g there exists an index k such that $a_k = u$, and with probability $g^2 = 0.382$ there exists an index i such that $b_i = u$.

19. The First Move

Without a third *independent* party, it seems impossible to introduce a random and trustworthy mechanism for deciding whose move is the first. As a palliative solution, the following procedure is suggested: immediately, after the first point (S_0, L_0) is generated, Hugo has a short period of time (say, a couple of seconds) to decide *who* must make the first move. One way to preclude *Hugo* from cheating and to introduce more variety to the Game, select $Q := 2Q$ on every consecutive run of the Game

with the same player. More detailed analysis of possible alternatives is beyond the scope of this paper.

20. Varieties of Nim-Game on Plane

Of many possible varieties I consider only two: the *Attrition* game and the *Flip-Flop* game.

In both games the moves are the same as in the Game described above in this paper. Only the goals are different.

Attrition game: The first player that reaches point $(0, 0)$ is a loser.

Flip-flop game: Only once during the Game players on their move can change the goal of the Game if

$$L \geq S + 2 \geq 7 \quad (54)$$

21. Winning Strategies

Let the winning points ${}_k w$ in the Attrition game. It is clear that both ${}_1 w = (0, 1)$ and ${}_2 w = (2, 2)$ are the winning points for *Cora*. Indeed, after *Cora*'s move $(0, 1)$ *Hugo* is losing the Game. The same is with $(2, 2)$: after that move *Hugo* is forced to reach $(0, 0)$, because *Hugo* can make either $(0, 2)$ or $(1, 1)$ or $(1, 2)$ move. Then *Cora* moves $(0, 1)$ and *Hugo* has no other choice but move $(0, 0)$.

Winning points f_k in Flip-Flop game: Although it seems confusing, actually the winning points for the Flip-Flop Game are very simple. It follows from an observation that for all $k \geq 2$

$${}_k w = w_k, \quad (55)$$

i.e., ${}_2 w = (3, 5)$; ${}_3 w = (4, 7)$; and only ${}_1 w = (2, 2)$ and ${}_0 w = (1, 0)$.

Hence, if the Game is in the *attrition phase*, then

$$f_k = {}_k w, \text{ otherwise } f_k = w_k. \quad (56)$$

From the (56) winning strategy it follows that “*Only once during the Game*” — requirement is inessential and it is introduced for a psychological reason only. The Game can be further modified if the flipper must pay for every flip, and the winner gets the “bank”.

After this paper was completed, the author discovered that the Game has been described in [13, 14].

22. Acknowledgements

I appreciate H. Wozniakowski for his suggestion, an anonymous reviewer for several corrections and B. Blake for comments that improved the style of this paper.

23. References

- [1] C. L. Bouton, "Nim, a Game with a Complete Mathematical Theory," *Annals of Mathematics*, Princeton 3, 35-39, 1901-1902.
- [2] M. Gardner, "Mathematical Games: Concerning the Game of Nim and its Mathematical Analysis," *Scientific American*, 1958, pp. 104-111.
- [3] M. Gardner, "Nim and Hackenbush," Chapter 14 in *Wheels, Life, and Other Mathematical Amusements*, W. H. Freeman, 1983.
- [4] E. Berry and S. Chung, "The Game of Nim," Odyssey Project, Brandeis University, 1996.
- [5] S. Pfeiffer, "Creating Nim Games," Addison Wesley, 1997.
- [6] R. D. D. Arruda, "Nim-Type Computer Game of Strategy and Chance," Master Project, CIS Department, NJIT, 1999.
- [7] R. Statica, "Dynamic Randomization and Audio-Visual Development of Computer Games of Chance and Strategy," Master Thesis, CIS Department, NJIT, 1999.
- [8] J. von Neumann and O. Morgenstern, "Theory of Games and Economic Behavior," 3rd edition, 1953.
- [9] R. D. Luce and H. Raiffa, "Games and Decisions-Introduction and Critical Survey," 2nd edition, Dover Publications, 1989.
- [10] G. M. Adelson-Velsky, V. Arlazarov and M. V. Donskoy, "Algorithms for Games," 1987.
- [11] H. Wozniakowski, "Private communication," Columbia University, Columbia, March 2002.
- [12] D. Kahaner, C. Moler and S. Nash, "Numerical Methods and Software," Prentice Hall, 1989.
- [13] C. Berge, "The Theory of Graphs and its Applications," *Bulletin of Mathematical Biology*, Vol. 24, No. 4, 1962, pp. 441-443.
- [14] W. A. Wythoff, "A Modification of the Game of Nim," *Nieuw Archief voor Wiskunde*, 199-202, 1907-1908.
- [15] B. Verkhovsky, "Winning Strategies and Complexity of Whytoff's Nim Computer Game," *Advances in Computer Cybernetics*, Vol. 11, 2002, pp. 37-41.

A Secure Transfer of Identification Information in Medical Images by Steganocryptography

Shuhong Jiao¹, Robert Goutte²

¹*Information and Telecom Department, Harbin Engineering University, Harbin, China*

²*Lab. CREATIS, UMR CNRS 5520, INSERM U 630, INSA, Université of Lyon, Bâtiment Leonard de Vinci, 21 avenue Jean Capelle 69621 VILLEURBANNE Cedex, France*

E-mail: jiaoshuhong@hotmail.com, goutte@creatis.insa-lyon.fr

Received July 26, 2010; revised August 23, 2010; accepted September 25, 2010

Abstract

The fast growth of the exchange traffic in medical imagery on the Internet justifies the creation of adapted tools guaranteeing the quality and the confidentiality of the information while respecting the legal and ethical constraints, specific to this field. The joint usage of steganography and cryptography brings an efficient solution, whose implementation in medical routine is realistic, thanks to the current progress in data processing (broad band Internet access and grid computing).

Keywords: Medical Image, Identification, Steganography, Cryptography

1. Introduction

The current needs in medical imaging security comes mainly from the development of the traffic on Internet (tele-expertise, telemedicine) and to establishment of medical personal file [1]. Among all possibilities it is interesting to work on the messages concealment in the image itself, then regarded as a medium coverage.

2. Objective

Insert in a medical image 2D black and white of any modality, a hidden message with all information identification (the radiologist and patient), historical of record, parameters of examination (nature, location, diagnosis and comments of the radiologist). Ideal characteristics sought:

- 1) Access, in reception, at the original image, without alteration or loss of information
- 2) The method used for the encryption must resist to attacks.
- 3) The hidden message must be, in reception, readable by the holder of the key.

This group of ideal conditions, with contradictory's imperatives, will be practically never satisfied. However; the proposed methods must be realized with objectives neighboring of these limits.

3. Features Specific Constraints in Medical Imagery

- 1) Need to use standards [2]
- 2) Respect of legislation
- 3) Rapidity and simplicity of implantation
- 4) Compatibility with JPEG compression

It is important to note that, in medical imagery domain, the compression, to be truly operational, shall keep useful information for the diagnosis and should relate essentially optimization of acquisition parameters, noise reduction and elimination of temporal redundancies.

4. Steganography

The steganography (Greek steganos: Covered and Graphein: Write) is the art hide a message in a medium coverage (medical image for example), so no one can distinguish the medium (original image), after the inclusion in the hidden message [3].

The hidden message can be a plain text or his encrypted version. In this latter case (which is interesting here), we use the term steganocryptography. For this, we use a prior encryption of the hidden message, before the introduction in the original image, considered here as a medium coverage. We propose to use a symmetric en-

encryption algorithm, known as international standard. En agreement with the work of W. Puech and M. Rodrigues [4], we choose the algorithm AES. This symmetric cipher uses blocs of data swapped of 128 bits and key sizes of 128, 192 or 256 bits.

4.1. Example of AES Encryption and Decryption [2]

Password: Creatisuniversitedelyon

Plaintext: Secure transfer in medical imagery

Encrypt it:

z4USOP2/O1sZdydqd4hSddOQUfRQabfKUCAoJP2drF6entGV

Decrypt it: Secure transfer in medical imagery

4.2. Conversion of this Encrypted Message in New Digital Message, ASCII 8 Bits by Character [5]

z4U50P2/O1sZdydqd4hSdd0QU.....

01110100011010001010101001101010100111101010000

00110010001011110100111100.....

5. Insertion by Steganography of Digital Data in Original Digital Image

These methods require five successive steps:

The first step is to divide the image into 8×8 square blocks (one byte for one grey level). In the second step we compute the different DCT coefficients (Discrete Cosine Transform [6] of these different blocks.

We select, in the third step, two spectral coefficients: (a_{mn}) and (a_{kl}) in the block i . Their location, in this block requires $2 \times (2 \times 3)$ bits. These 12 bits, expressed with two ASCII characters (6 bits per character) are the beginning of the shared hidden key.

In the 4th step we use the following rule: If $b_i = 1$ and $(a_{kl}) > (a_{mn})$ or if $b_i = 0$ and $(a_{kl}) < (a_{mn})$ nothing is changed. If these conditions are not carried out we exchange the values of (a_{kl}) and (a_{mn}) .

The amplitudes of the change of the spectral coefficients can be adjusted depending on the level of noise and the rounding's error. The frequency's position of the two coefficients is important. If it is located in BF, the method is robust, but risk of be visually detectable.

Instead, if it is located in HF, the original image will be virtually unchanged, but the method will be more sensitive to photometric fluctuations.

5.1. Variant

If necessary, for obtain another form of resistance to attacks, we can take different coordinates for the coefficients (a_{mn}) and (a_{kl}) . In this case, it is possible, for example, with one bit, of displace the sequence for the block i , for obtain the sequence of the block $i + 1$.

If coordinates $(a_{kl})i = 010$ and 011 , and if coordinates $(b_{mn})i = 001$ and 100 we obtain the key chain for the block i : 010011001100 and 001001100110 for the block $i + 1$.

The detection of these coordinates is more difficult, but it is not possible to choice a single optimal domain for these coefficients, in the spectral plan.

We have the ability to hide 1 bit per square block or, for an original image of size 1024×1024 , to hide 16384 bits. With ASCII code, extension UNICODE (with 8bits per characters), we obtain the possibility to hide 2048 characters in this image.

The 5th step is the extraction of the hidden message. The method is similar to that the insertion: at the reception we compare the values of the two selected coefficients. The previous rule allows us to know if the bit concerned is 1 or 0.

Remark: The marking brought a very slight loss of information, since the image in the reception is not exactly identical to original, but the quantity of bits transferred is the same.

It is important to note this favorable factor: Any location in the spectral domain involve an displaying in the image plane, which facilitate the invisibility of the insertion

6. Radiographic Application

The original image (**Figure 1**) is a pulmonary radiography, obtained in tomodensitometry (X ray scanner).

We isolate on this image one block 8×8 .

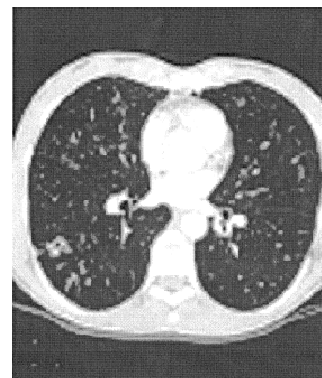


Figure 1. Pulmonary scannography

142	120	100	87	82	78	79	81
131	113	98	87	79	83	82	82
119	107	97	90	84	83	82	80
119	112	106	100	95	85	83	80
134	127	118	107	100	91	87	82
150	140	126	111	100	96	91	85
156	144	129	114	103	96	92	86
145	132	119	108	100	91	88	83

(Original block image, before transform)

After Discrete Cosine Transform DCT, we obtain this block, in the spectral domain.

3.2240	0.5578	0.1552	0.0572	0.0132	0.0177	0.0020	0.0074
0.2305	-0.0818	0.0513	0.0124	0.0143	0.0023	0.0074	-0.0042
0.0148	<u>0.0510</u>	<u>0.0676</u>	0.0211	0.0094	0.0037	0.0036	-0.0020
0.0986	0.0704	0.0268	0.0042	-0.0060	0.0037	0.0014	0.0002
-0.0270	-0.0070	0.0226	-0.0046	0.0152	-0.0070	-0.0071	0.0109
0.0156	0.0113	-0.0051	-0.0028	0.0062	-0.0026	-0.0046	0.0044
-0.0194	-0.0052	0.0026	-0.0027	0.0009	-0.0005	-0.0010	-0.0001
0.0048	0.0046	0.0007	0.0010	0.0002	-0.0011	-0.0013	0.0017

With $k = 3$, $l = 2$ and $m = 3$, $n = 3$, $a_{kl} = 0.0510$, and $a_{mn} = 0.0676$

Here: $a_{kl} = 0.0510 < a_{mn} = 0.0676$

If I want to introduce a hidden bit $b = 0$, nothing is changed.

If I want to introduce a hidden bit $b = 1$, we invert the values of a_{kl} and a_{mn}

$a_{kl} = 0.0676 > a_{mn} = 0.0510$

In this case ($b = 1$) we obtain after inverse transformation DCT^{-1}

142	120	101	88	83	78	78	79
131	113	98	87	79	83	82	81
119	107	97	90	84	83	82	81
119	112	105	99	94	85	84	82
134	127	117	106	99	91	88	84
150	140	126	111	100	96	91	86
156	144	129	114	103	96	92	85
145	132	120	109	101	91	87	81

(bloc image after crypt)

On each pixel modified, the positive or negative error is approximately one grey level. Only 24 pixels dispersed on 64 are modified and the mean of grey levels is unchanged. The image is good and the message is not visible. Practically, in a medical image with 256 gray levels, a deviation of one pixel has no physical significance and presents no interest for visual observation or subsequent numerical processing. It results mainly from the presence of noise and artifacts of rounding obtained when quantifying.

To avoid the consequences of too large distance between these 2 factors (a_{kl} and a_{mn}) which may introduce a important disturbance of the spectrum) and also the opposed consequence of a gap too low (rendering the method unstable in presence of noise) we use, if necessary the following rule:

If the gap $a_{mn} - a_{kl}$ is $2e$, and if $0.0010 < e < 0.0120$, nothing is modify;

If $e < 0.0010$ we take $a_{kl}' = 0.5(a_{kl} + a_{mn}) - 0.0010$

and $a_{mn}' = 0.5(a_{kl} + a_{mn}) + 0.0010$

If $e > 0.0020$ we take $a_{kl}' = 0.5(a_{kl} + a_{mn}) - 0.0120$

and $a_{mn}' = 0.5(a_{kl} + a_{mn}) + 0.0120$

0.0010 and 0.0120 are too adjustable parameters, in function of the level of noise in the image and the confidential degree wished.

6.1. Example of Hidden Message

Place of examination: Cardiologic Hospital, HEH Lyon, Service of Radiology.

Date: 25/15/2009

Instrumentation: Tomodensitometer, Pulmonary Scanner

Radiologist: Dr. Jean Martin

Identification: XXXXXXXX

Patient: Michel Dupont

Identification: YYYYYYYY Age:45 years

Conditions of observation: Axial

Incidence transverse

Commentaries: This patient presents a small parenchymatous nodule of the higher segment of the lobe lower right. Presence also of a discrete bronchial dilation in the average lobe.

This hidden message (in italic) possesses 347 characters, with spaces. In ASCII we obtain 1.421.312 bits, it is necessary to have a dimension for the original image equal or upper of 1.5 megabits. This dimension is usual in medical imagery.

Thus, in our example, the proposed coding is invisible and involves no loss of useful information.

This approach has been submitted to a panel of radiologists from hospital, specialists from different imaging modalities and their comments and proposed additions have been included in this final implementation.

7. Generalization

The method proposed is well suited to the JPEG compressed images [7] because, in this case, the compression algorithm use also the Discrete Cosine Transform (DCT). It is possible to extend this method to color images, which can be considered as a set of three images black and white (RGB). In three dimensional imaging, we can consider $8 \times 8 \times 8$ cube and use the 3D block DCT algorithm.

8. Conclusion

The steganocryptography can transmit, with invisibility and robustness, the information accompanying a medical digital radiography. The constraints of legal requirements, safety and confidentiality are fully satisfied.

9. References

- [1] Liliane DUSSEY, Rapport du Conseil National de l'Ordre des médecins, France, 2002.
- [2] Federal Information Processing Standards, Public. 197, announcing the Advanced Encryption Standards (AES), USA, 2001.
- [3] Christian REY, Jean Luc DUGELAY, Panorama des méthodes de tatouage, *Traitement du Signal*, Vol. 18, spécial No., 2001.
- [4] W. Puech and J. M. Rodrigues, "Crypto-Compression of Medical Images by Selective Encryption of DCT," *13th European Signal Processing Conference, EUSIPCO'05* Antalya, Turkey, 2005.
- [5] Wikipedia, Encyclopédie libre, Norme ASCII.
- [6] Y. Wang and P. Moulin, "Steganalysis of Block-DCT, Image Steganography," *Proceedings of IEEE Workshop on statistical Signal Processing*, St Louis, 2003, pp. 339-342.
- [7] H.-W. Tseng and C.-C. Chang, "Steganography Using JPEG Compressed Image," *Fourth International Conference on Computer and Information Technology (CIT 04)*, 2004, pp. 12-17.

Performance Improvement of Wireless Communications Using Frequency Hopping Spread Spectrum

Yang Liu

Department of Information Technology, Vaasa University of Applied Sciences, Vaasa, Finland

E-mail: yang.liu@puv.fi

Received June 18, 2010; revised August 1, 2010; accepted September 6, 2010

Abstract

To improve the performance of short-range wireless communications, channel quality must be improved by avoiding interference and multi-path fading. Frequency hopping spread spectrum(FHSS) is a transmission technique where the carrier hops from frequency to frequency. For frequency hopping a mechanism must be designed so that the data can be transmitted in a clear channel and avoid congested channels. Adaptive frequency hopping is a system which is used to improve immunity toward frequency interference by avoiding using congested frequency channels in hopping sequence. In this paper mathematical modelling is used to simulate and analyze the performance improvement by using FHSS with popular modulation schemes, and also the hopping channel situations are investigated.

Keywords: Frequency Hopping Spread Spectrum, Adaptive Frequency Hopping, Performance Evaluation

1. Introduction

In this paper the focus is to improve wireless communication performance by adaptive frequency hopping which is implemented by selecting sets of communication channels and adaptively hopping sender's and receiver's frequency channels and determining the channel numbers with less interference. Also the work investigates whether the selected channels are congested or clear then a list of good channels can be generated and in practice to use detected frequency channels as hopping sequence to improve the performance of communication and finally the quality of service.

The Fourier transform mathematical modules are used to convert signals from time domain to frequency domain and vice versa. The mathematical modules are applied to represent the frequency and simulate them in MATLAB and as result the simulated spectrums are analysed. Then a simple two-state Gilbert-Elliot Channel Model [1,2] in which a two-state Markov chain with states named "Good" and "Bad" is used to check if the channels are congested or clear in case of interference. Finally, a solution to improve the performance of wireless communications by choosing and using "Good" channels as the next frequency hopping sequence channel is proposed.

2. Review of Related Theories

2.1. Spread Spectrum

Spread spectrum is a digital modulation technology and a technique based on principle of spreading a signal among many frequencies to prevent interference and signal interception [3]. As the name shows it is a technique to spread the transmitted spectrum over a wide range of frequencies. It started to be employed by military applications because of its Low Probability of Intercept or demodulation, interference and anti-jamming from enemy side. The idea of spreading spectrum is to spread a signal over a large frequency band to use greater bandwidth than the data bandwidth while the power remains the same. And as far as the spread signal looks like the noise signal in the same frequency band it is difficult to recognize the signal which this feature of spreading provides security to the transmission. Compared to a narrowband signal, spread spectrum spreads the signal power over a wideband and the overall SNR is improved because only a small part of spread spectrum signal is affected by interference. In sender and receiver sides of a communication system one spreading generator is located based on the spreading technique they synchronize the received modulated spectrum. Shannon capacity Eq-

uation is the basis for spread spectrum systems, which typically operate at a very low SNR, but use a very large bandwidth in order to provide an acceptable data rate per user. Applying spread spectrum principles to the multiple access environments is a development occurring over the last decade [4].

2.2. Frequency Hopping Spread Spectrum (FHSS)

Frequency hopping spread spectrum is a transmission technology used in wireless networks and a technique to generate spread spectrum by hopping the carrier frequency. FHSS uses narrow band signal which is less than 1 MHz. In this method data signal is modulated with a narrowband carrier signal that hops in random and hopping happens in pseudo-random predictable sequence in a regular time from frequency to frequency which is synchronized at both ends. FHSS improves privacy. It is a powerful solution to avoid interference and multi-path fading (distortion). It decreases narrowband interference, increases signal capacity, and improves the signal to noise ratio. The efficiency of bandwidth is high and it is difficult to intercept. Also this transmission can share a frequency band with many types of conventional transmissions with minimal interference. For frequency hopping a mechanism must be defined to transmit data in a clear channel and to avoid the congested channels. Frequency hopping is the periodic change of transmission frequency and hopping happens over a frequency bandwidth which consists of numbers of channels. Channel which is used as a hopped channel is instantaneous bandwidth while the hopping spectrum is total hopping bandwidth. Frequency hopping categorized into slow hopping and fast hopping which by slow hopping more than one data symbol is transmitted in same channel and by fast hopping frequency changes several times during one symbol. Hopping sequence means which next channel to hop. There are two types of hopping sequence: random hopping sequence and deterministic hopping sequence. The focus of this work is on slow and deterministic frequency hopping sequence. In a frequency hopping network, there can be different number of receivers which one sender is designed as base which is responsible to transmit the synchronization data to the receivers.

2.3. Adaptive Frequency Hopping

Adaptive frequency hopping (AFH) is a system in which devices constantly change their operating frequency to avoid interference from other devices to improve communication performance [5]. AFH classifies channels as "Good" or "Bad" and adaptively selects from the pool of

Good channels. Bad channels are the channels with interference. The idea of using AFH is to hop only over Good channels, which means to choose the frequency channels that have less interference. For using AFH there must be a mechanism to choose "Good" and "Bad" channels. Received signal strength indication (RSSI) tells each channel quality to generate a list for "bad channels". The system and principle of a proposed AFH scheme are illustrated in [6], assuming that there is a duplex transceiver system. The system is an ordinary frequency hopping system which uses a number of narrowband channels.

2.4. Channel and Interference

Compared to the other kinds of wireless communications, high frequency communication is selectively fading because of the multi-path propagation and abundance of interference from the others. Interference always exists in any wireless system. Bit error rate is highly important for the performance improvement of the communication systems. Every frequency channel due to interferences and fading shows different signal to noise ratio. In some of the frequency channels there are stronger SNR and these channels are more suitable for the transmission. Adaptive frequency hopping is a powerful solution and a technique that deals with different kind of interference, noise and fading. For the simplicity of the work the focus is only on the interference as the main disturbance in achieving a desired and suitable transmission quality and neglects all the other disturbance resources such as other noises and fading.

3. Gilbert-Elliot Channel Model

3.1. Markov Chain

Bit error models generate a sequence of noise bits (where 0's represent good bits and 1's represent bit errors) to produce output bits, and modulo 2 to the input bits must be added. Models are grouped into two classes: memoryless models and models with memories [7]. In memoryless models the noise bits are produced by a sequence of independent trials that each trial has the same probability $P(0)$ of producing a correct bit and probability $P(1) = 1 - P(0)$ of producing a bit error.

The actual measurement from the communication channels indicates that these channels are with memories, for example the probability of 100th bit error is dependent on the 99th bit. For modelling of such kind of probabilistic situation a commonly technique is used which is Markov chain. This technique helps to make the bit error probability depend on the states. The use of Markov

chain in bit error models has been introduced by Gilbert-Elliot for the first time. Gilbert-Elliot channel model based on Markov chain has two states G (Good) and B (Bad). In state G, transmission is error-free and in state B the link has probability h of transmitting a bit correctly. **Figure 1** shows a transition diagram and bit error probabilities for Markov chain. The model has three independent parameters (p , P and h) to describe the error performance of wireless links. The situation of small p is where transition jumps from B to G and the capital P is where transition jumps from G to B. Also the states B and G tend to persist and the model simulates bursts of errors.

The parameters p , P and h are not directly observable and therefore must be determined from statistic measurements of the error process. It's also important to note that runs of G alternates with runs of B. The run length has geometric distributions, with mean $1/P$ for the G-runs and $1/p$ for the B-runs.

3.2. Geometric Distribution

The run lengths of Good and Bad states can be expressed by geometric distribution in which for the Good runs, mean value of $1/P$ and for the Bad runs, the mean value of $1/p$ is used. The time fraction in both of Good and Bad states based on persistence in each state can be calculated, for example the fraction of time spent in B state is:

$$P(B) = \frac{P}{P + p} \quad (1)$$

The sequence of states cannot be reconstructed from the sequence of bits in the error process, because both of 0's and 1's (the Good bits and Bad bits) are produced in the B state and since bit errors happen only in state B with probability of $1-h$ then the probability of error is:

$$P(1) = P(1, B) = P(B)P(1 | B) = (1-h) \frac{P}{P + p} \quad (2)$$

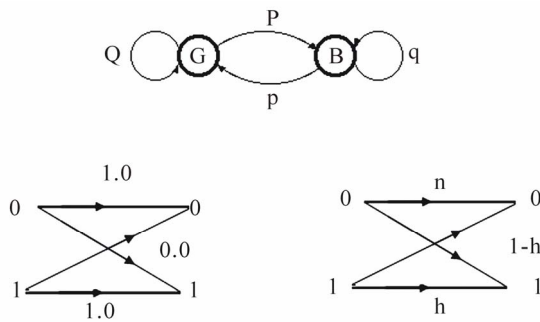


Figure 1. Transition diagram and bit error probabilities model.

The bits of the error process (runs of 0's and 1's) and the distribution of run lengths of 0's (error gaps) and 1's (error bursts) are observable to determine model parameters.

3.3. Parameter Estimation

The determination of the three parameters p , P , and h from measurements of the error process requires that parameters be expressed as functions of three other parameters that are directly observable. For Markov chain parameter estimation the functions have been proved formerly [7]:

$$\mu_{EB} = \frac{1}{1 - q(1 - h)} \quad (3)$$

$$\mu_{EG} = \frac{h(1 - Q) + (1 - q)}{(1 - h)(1 - Q)[1 - q(1 - h)]} \quad (4)$$

$$\sigma_{EB}^2 = \frac{\sqrt{q(1 - h)}}{1 - q(1 - h)} \quad (5)$$

$$\sigma_{EG}^2 = \frac{(1 - h)(qJ + p - Q)J(J + 1)}{[1 - q(1 - h)](J - L)(1 - J)^3} + [J \leftrightarrow L] - \mu_{EG}^2 \quad (6)$$

In the Equations μ_{EB} is the mean error burst length and μ_{EG} is the mean error gap length, σ_{EB}^2 is the variance of the error burst distribution and σ_{EG}^2 is the variance of error gap distribution. J and L are defined as:

$$2J = Q + hq + \sqrt{(Q + hq)^2 + 4h(p - Q)} \quad (7)$$

$$2L = Q + hq - \sqrt{(Q + hq)^2 + 4h(p - Q)} \quad (8)$$

4. Matlab Modelling

4.1. Gilbert-Elliot Modelling

Gilbert-Elliot channel model is used for modelling a telecommunication channel. For obtaining the parameters of this model, first a sequence of data bit is given to the transmitter and then from the receiver side the transmitted data is received as output data. With the input sequence and output sequence, bit error sequence can be calculated easily. By having this bit error sequence and the method of parameter estimation in [7] the model parameters can be calculated.

For this reason channel simulation is done with Simulink. To obtain the bit sequence of input and output, two variables with names "in" and "out" are used. With XORing the input and output bit sequences the bit error sequence is calculated. By setting bit error sequence at argument of function `marcov`, Markov parameters can be

achieved from the output of function `marcov`. In function `marcov` by using the function `coef`, the sequence of error burst and error gap can be calculated. After calculation of statistical parameters of these two sequences, Markov parameters can be then calculated by function `fsolve` which solves nonlinear Equations.

4.2. Defining Markov Chain Parameters

To obtain Markov parameters in Matlab, a function of `marcov` is created as follow.

```
error_seq = xor(in,out);
z = marcov(error_seq);
z = fsolve(@solv,[.1 .1],[],meh,meg,veg);
```

In this function the error sequence is first inputted to the function of `coef` then the output of sequence is obtained as 0's and 1's.

For example assume there is a sequence of:

```
error_seq = [0 1 0 0 0 1 1 1 0 1 1 0 1 0 1 1 1 1 0 0 0]
```

Then at the output of function `coef` will obtain:

```
error_burst_seq = [1 3 2 1 4]
```

```
error_gap_seq = [1 3 1 1 3]
```

Now from the output `error_burst_seq` and `error_gap_seq` which is the sequence of error runs it can be seen that the length of the run of the errors has come in order of their happenings. Next step is to calculate the mean value and the variance of the sequence.

4.3. Channel Performance Evaluation

100 communication channels are evaluated and channel performances are categorized based on Gilbert-Elliot channel model. Gilbert-Elliot model is used for modeling a real communication channel and evaluating the performance of the channels, in which first a bit sequence is sent through a channel and then its bit error sequence is computed. Using bit error sequence helps to find out the parameters of the model. Markov parameters can be used to find following two functions: Fraction of time spent in state B (Bad) from Equation (1) and probability of the error from Equation (2).

To evaluate the channel performance based on Gilbert-Elliot Markov chain model the information about bit error sequence is collected to simulate the channel model with Matlab. Additive white Gaussian noise (AWGN) channels with 100 random input powers are used in simulation.

First the percent of time is computed which each channel spends in state B or in the other word the probability of being in state B that multiplied by 100. **Figure 2** shows the result of each channel being in Bad state.

The achieved result from **Figure 2** helps to categorize the Channels based on three different groups as "Bad

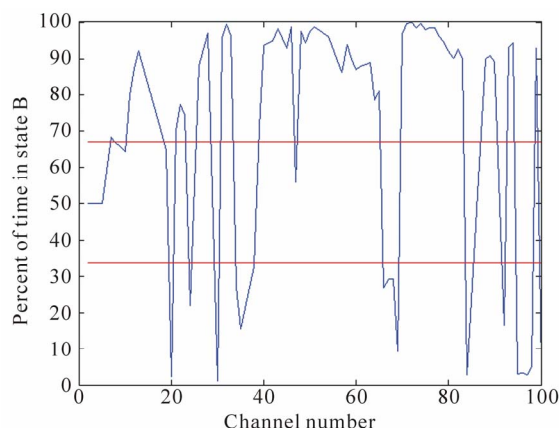


Figure 2. Percent of time that each channel spends in state B.

Channels", "Good Channels" and "Very Good Channels" by identifying two threshold values and categorizing those decides to transmit data over "Very Good" and "Good" channels then by such transmission the performance of the communication system can be improved. Then the error probability in Bad state for each channel is computed. **Figure 3** shows these probabilities for 100 different channels.

4.4. Testing

Gilbert-Elliot channel model is used to simulate the error process and correctly reproduce all of its statistical properties. To validate the model, the error process generated by the model must be compared to the measured error process. For testing, the program bit error sequence is generated using Markov chain model. Two programs are made as follow: `marcov_gen` is a bit error sequence generator for Markov parameters and `marcov_test` tests the bit error sequence and the output is displayed in workspace.

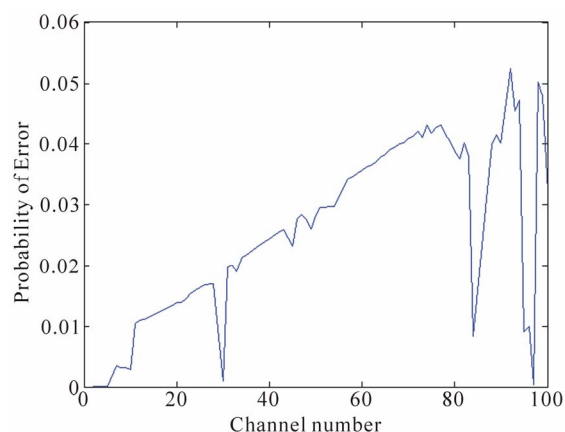


Figure 3. Error probability being in Bad state for each channel.

The objective of the parameter estimation is to choose values of the model parameters that generate error burst and error gap distributions that reassembles the corresponding measured distributions as close as possible. Therefore for testing the mean and variance of error burst and error gap of regenerated error sequence are calculated and compared by statistical parameters of channel bit error sequence, where the result is shown in **Table 1**.

A: Statistical parameters channel error sequence.

B: Statistical parameters of regenerated error sequence.

For testing, first Markov model parameters of a channel error sequence are computed, then a sequence of the model is generated and statistical parameters are computed. The statistical parameters must be as equal as channel error sequence. It is important to mention that first state of the Markov model in function `marcov_gen` chooses the probability 0.5, so sometimes two different answers can be seen and that the nearest one to the error sequence statistic is the correct one.

5. Evaluation of Frequency Hopping

To design the frequency hopping (FH) model, MATLAB Simulink has been used. The spreader at transmitter section is an M-FSK modulation but the input of the modulator is hopping index. This section consists of a PN Sequence Generator, an Assemble Packets block and a Goto block as shown in **Figure 4**.

The design of frequency hopping spreader is shown in **Figure 5**. The spreader part consists of M-FSK modulator base (with M equal to 64), a From block (Hop index that is created in previous step), a To Frame block and a Multiplication block. The block parameter of FSK modulator is 64 in M-FSK number and it means that there are 64 hopping sections. These sub-bands are selected by the hop indexes.

The design of frequency hopping despreader, is the same as spreader section but the output of M-FSK modulator block is complex conjugated as shown **Figure 6**.

This frequency hopping model is used for evaluation of three different modulations: QAM, QPSK, GFSK, and compares the performance with the situation without frequency hopping. Performance evaluation is based on BER values under two situations (with and without FH) versus normalized signal-to-noise ratio (SNR) measured by E_b/N_0 values of the channel, as shown in **Figures 7-9**.

From **Figure 7** it can be seen that applying FH with QAM modulation does not lead to a sensible improvement in performance or significant reduction of BER. From **Figure 8** it can be seen that applying FH with QPSK modulation gives a good result and reduces

Table 1. Statistical parameters of channel error and regenerated error sequence.

	error burst mean	error gap mean	error gap variance	
SNR = 3dB	1.0568	18.1134	319.4516	A
Input power = 1	1.0492	18.4713	319.3184	B
SNR = 3dB	1.1456	7.7868	48.9981	A
Input power = 2	1.1500	8.0421	55.0026	B
SNR = 3dB	1.2201	5.6570	25.5754	A
Input power = 3	1.2271	5.5166	24.5044	B

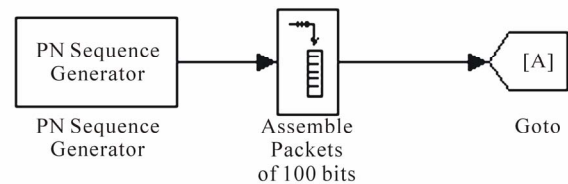


Figure 4. Model of frequency hopping in Simulink.

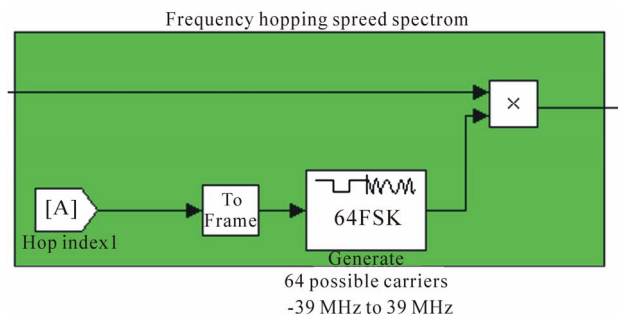


Figure 5. Design of frequency hopping spreader.

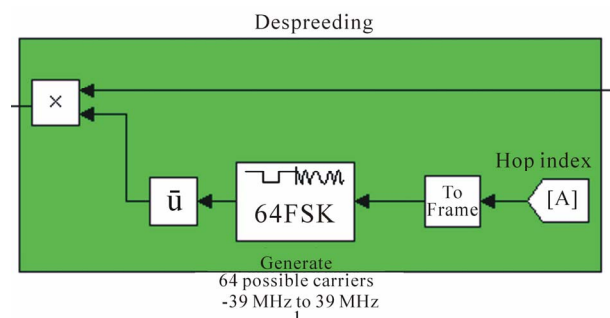


Figure 6. Design of frequency hopping despreader.

significantly BER compared to without FH at same level of SNR. From **Figure 9** it can be seen that applying FH with GFSK modulation reduces dramatically BER compared to without FH at same level of SNR and lead to a much higher performance.

Performance comparison of QAM modulation with and without FH

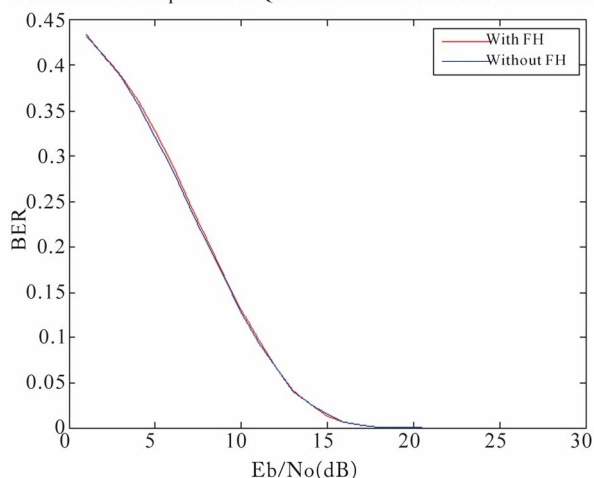


Figure 7. QAM modulation.

Performance comparison of QPSK modulation with and without FH

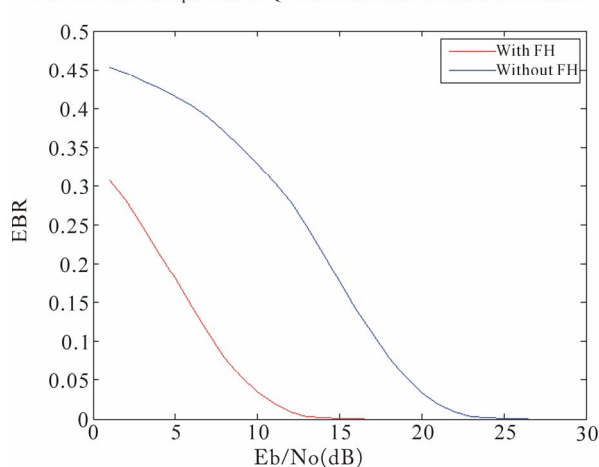


Figure 8. QPSK modulation.

Performance comparison of GFSK modulation with and without FH

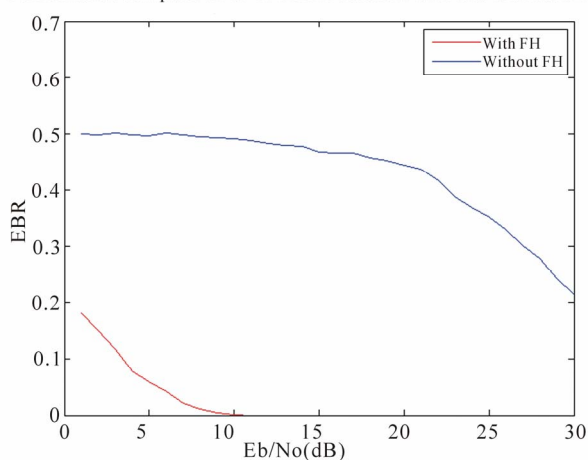


Figure 9. GFSK modulation.

In overall, based on the evaluation results it can be concluded that applying the designed FH schemes with certain modulations can improve their communication performances, especially at weak SNR levels as most cases of short range wireless communications have.

6. Conclusions

As a result of the work it can be concluded that adaptive frequency hopping is a powerful technique to deal with interference and Gilbert-Elliott channel model is a good technique to analyze the situations of channels by categorizing the channel conditions based on their performance as Good or Bad, and then apply adaptive frequency hopping which hops frequencies adaptively by analyzing the state of the channel in case of environmental problems such as interferences and noises to improve the communication performance.

Frequency hopping spread spectrum is modelled with MATLAB and three different modulations *i.e.* QAM, QPSK and GFSK are studied to investigate which of these modulations are good to apply with FHSS model. The simulation results show that applying FHSS with QAM modulation dose not lead to a remarkable reduction of BER, but with QPSK modulation gives a good result and reduces BER at lower SNR, while in GFSK modulation shows a significant reduction of BER and lead to a high performance.

7. References

- [1] E. N. Gilbert, "Capacity of Burst-Noise Channels," *Bell System Technical Journal*, Vol. 39, 1960, pp. 1253-1265.
- [2] E. O. Elliott, "Estimates of Error Rates for Codes on Burst-Noise Channels," *Bell System Technical Journal*, Vol. 42, 1963, pp. 1977-1997.
- [3] R. E. Ziemer, R. L. Peterson and D. E. Borth, "Introduction to Spread Spectrum Communications," Prentice Hall, Englewood Cliffs, New Jersey, 1995.
- [4] R. J. Bates and D. W. Gregory, "Voice & Data Communications Handbook," McGraw-Hill Osborne Media, Berkeley, CA, 2001.
- [5] Y. Liu, "Enhancement of Short Range Wireless Communication Performance Using Adaptive Frequency Hopping," *Proceeding of 4th IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, Dalian, China, October 2008, pp. 1-5.
- [6] J. Zander and G. Malmgren, "Adaptive Frequency Hopping in HF Communications," *IEE Proceedings Communications*, Vol. 142, No. 2, 1995, pp. 99-105.
- [7] J. J. Lemmon, "Wireless Link Statistical Bit Error Model," Institute for Telecommunication Sciences, 2002.

Solutions for 3 Security Problems and its Application in SOA-FCA Service Components Based SDO

Nannan Wang, Zhiyi Fang, Kaige Yan, Yu Tang, Xingchao An
College of Computer Science and Technology, Jilin University, Changchun, China
E-mail: wangnannan_1985@126.com, zyfang@public.cc.jl.cn, yankaige@163.com,
mailtangyu@gmail.com, 81435756@qq.com

Received June 19, 2010; revised August 2, 2010; accepted September 10, 2010

Abstract

Service-Oriented Architecture (SOA), which is an open architecture, provides developers with more freedom. However, its security problem goes from bad to worse. By taking an insurance business in Formal Concept Analysis (SOA-FCA) Service Components based Service Data Object (SDO) data model transfer with proxy as an example, the security issue of SDO data model was analyzed in this paper and this paper proposed a mechanism to make sure that the confidentiality, integrity, and non-repudiation of SDO data model are preserved by applying encryption/decryption, digest, digital signature and so on. Finally, this mechanism was developed and its performance was evaluated in SOA-FCA Service Components.

Keywords: Service-Oriented Architecture, Service Data Object, Data Confidentiality, Data Integrity, Data Non-Repudiation

1. Introduction

As a new way and environment for distributed software system, Service-Oriented Architecture (SOA) [1], contains running environment, programming model, architecture and methodologies. Service plays a core role in SOA. The whole IT system is treated as a collection of services, not a collection of application programs. Each service provides a unique function and the granularity of each function can be either big or small. Other applications or services can “consume” this service. SOA aims at providing an exchangeable, highly adaptable and flexible standard. SDO can facilitate this and provide some help. SDO [2], which can simplify and unify the access to heterogeneous data by using a unique API, can also be used in other data process applications. Due to using of a new and open standard in substitution for traditional security parameters, SDO data model has a lot of data security issues. The new standard doesn’t take the security into its consideration at the time of its origination. Thus, its security issue, especially data security issue, becomes even worse than before. The confidentiality, integrity, non-repudiation issues should be taking into account, when it comes to data security. Through an in-depth study of insurance business, this paper selects six representative insurance products and abstracts the

information on the insurance application to be the entities of formal context. This paper solves the problem of data security, the confidentiality, integrity, non-repudiation on this insurance product based SDO data model.

2. Introduction and Analysis of SOA-FCA Service Components of Insurance

2.1. FCA-Based Business Entity Object

Six representative insurance products based SDO data model were selected by this paper. We abstract the form of insurance application to be objects of formal context and the insurance underwriter, insurance applicant, etc to be attributes. These six representatives involve compulsory insurance for traffic accident of motor vehicles, commercial insurance for motor vehicles, insurance for farming reproducible sow, hail insurance for planting onion, basic property insurance and construction insurance.

FCA provides a formal process for extracting and classifying all the business concepts involved in a particular business system. By excluding the influence of human factor on the analysis result, this rigorous mathematical tool makes the analysis result of the business entity of insurance underwriting module, a core business of in-

surance, much more objective. Driven by real business operation, eight business entities of insurance underwriting are abstracted from the six representatives. The corresponding E-R diagram is shown in **Figure 1**. There are three business entities that are reused more frequently than others. They are basic information about insurance application, computation sheet of insurance amount and in insurance payment schedule. These three entities are more likely to be reused when new insurance products are introduced. With the business expansion and request update, these entity objects of insurance can be reused directly or reorganized into new entity objects to be used by the new insurance products.

2.2. Analysis of Underwriting Service Component Based on FCA

The well-designed underwriting business entities in 2.1 Section suggest that a specific business function can be achieved by applying some operations such as adding, deleting, editing and querying on certain entities. A service component can be viewed as a combination of certain operations and entities. Real business components in an insurance information system spans across two dimensions of function and insurance category, which will reuse the components to a larger extent. For service components based SDO are reused continuously, the problem of complexity in system will increasingly become serious so it is necessary for us to pay attention to its security issues. Thus more regard must be paid to security requirement of the information transmitted on the internet and we

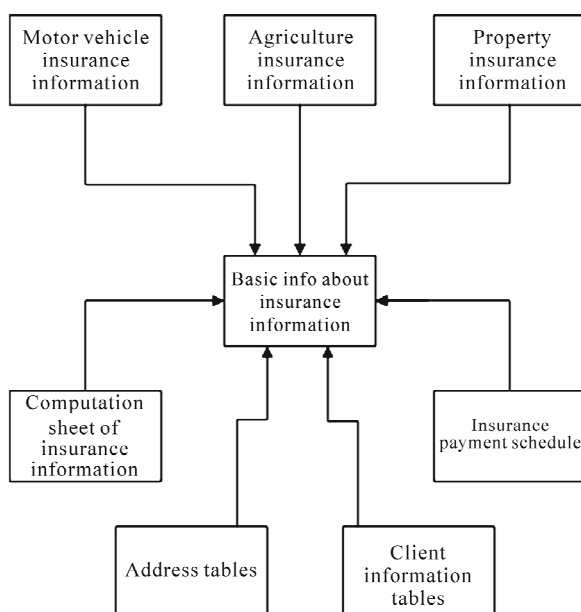


Figure 1. Business entity relationship.

must find a comprehensive solution which can solve the data security issues of SDO data model.

3. Analysis to Data Security Issues of SDO

3.1. Data Confidentiality of SDO

The confidentiality [3] indicated that when the data were transmitted, it cannot be eavesdropped. It means that the information should not be wiretapped, and cannot see the original message even if they got the data. The data encrypted can only be decrypted by authenticated users. In some SOA environments free from protection, the message transmitted on the internet can be easily overheard and intercepted by unauthenticated users.

3.2. Data Integrity and Non-Repudiation of SDO

The data integrity [4] makes sure that the data should not be tampered in the process of transmission. If the data are tempered, the receiver is supposed to know this. The non-repudiation [5], which is also called data signature, means that both the sender and receiver should not deny its transmission and reception respectively. Both integrity and non-repudiation are hard to achieve, as SDO takes little attention to its openness and security.

4. Requirements Analysis and Solutions of Data Security

4.1. Analysis of SDO Data Security Solution

The application message level's encryption of SOA is a reasonable solution. Due to the judgment that public key encryption, private key decryption, public key signature and public key authentication run slow and cannot be used to big sum of numbers, this paper only commits encryption and signature on private data contained in SDO. This brings two major advantages. First, the data processed by this mechanism are also based on the open standard and the receiver can treat them as SDO data. Second, the running speed will not slow down while the numbers of processed data increased correspondingly.

4.2. Overview of Instance and Data Security Requirements

In the real scene of the SOA-SCA (Service Component Architecture), if a service request is submitted by a service consumer, the SDO data submitted by users may be through a number of service providers so as to achieve

this service request. Take the example of **Figure 2**; the customer called Tony would like to complete the transfers request between insurance accounts, according to the agents WTAM of insurance services. First of all, a brief security analysis of the service request is carried out in the following aspects: data confidentiality, data integrity and non-repudiation.

It is indispensable to ensure the information confidentiality in the transmission not only from John to the agent WTAM but also from the agent WTAM to the insurance agent so that we can achieve confidentiality of information. In order to ensure the integrity of the information filled by Tony, it is necessary to prevent network hackers tampering with the data, but also to prevent Agents (WTAM in the scene) modifying customer data. For the agent WTAM, it must has non-repudiation of John's insurance accounts operation, and at the same time, for the insurance system, it must not only ensure that users cannot deny their operations of the accounts, but also ensure that the intermediate agents cannot deny account deputy operation that they would like to do instead of customers.

4.3. Security Solutions of Data Confidentiality, Integrity, Non-Repudiation

The design of Data security solution (data privacy, data integrity, data non-repudiation) is shown as follows.

Step 1. Tony, WTAM, Insurance agent First, to generate their own public key, private key, and then their public key will be posted to the CA (certificate authority) respectively, and CA generate their certificate, the certificate contains the public key and their own identity information.

Step 2. If Tony wants to send their service requests to WTAM Service Agent, first he would go to the CA certificate to get WTAM certificate, and then CA replies WTAM certificate encrypted with Tony's public key.

Step 3. Tony analyses WTAM certificate from CA and obtains WTAM public key, and sends confidential data

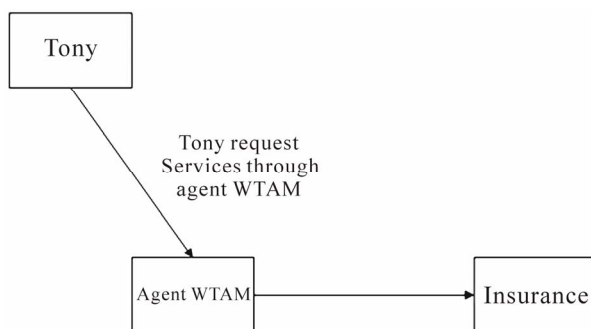


Figure 2. Tony wants to do insurance transfer through agent WTAM.

of SDO data to WTAM encrypted with WTAM agent's public.

Step 4. Tony gets the information summary of SDO data to send, and signs the information summary with their private key.

Step 5. Tony calls agent WTAM service and sends SDO data encrypted and signs with Tony signature to the WTAM agent.

Step 6. WTAM agent requires Tony certificate from CA, and resolves certificate to get John public key.

Step 7. WTAM decrypts confidential data of the SDO data with its own private key from Step 5; generates the specific data of information.

Step 8. WTAM verifies the Tony's information signature and makes use of information summary from Step 7 and Tony's public key from step 6, if proved to be successful, then the message from Tony to the WTAM has not been tampered with by third parties illegal in network transmission and after Tony signed, then continues to Step 9, otherwise, there are two errors, one is information may be modified in transmission, that is information integrity has been destroyed, the second is likely Tony's signature failure, in which case, WTAM will be prompted to require Tony to re-send service requests, and tells Tony the specific reasons of errors, the service process is terminated.

Step 9. If step 8 is successful, WTAM requests bank's certificate from the CA and resolves the insurance agent's public key.

Step 10. WTAM encrypts SDO confidential data to be sent to insurance agent with the bank's public key.

Step 11. WTAM adds its signature with its own private key in the SDO data which has been signed by Tony.

Step 12. The SDO encryption data from step 10, WTAM signed data from step 11, and Tony signed data from step 4 will be sent to insurance agent by WTAM.

Step 13. Insurance agent receives encrypted and signature data from WTAM, and uses their own private key to decrypt the encrypted data to obtain the information of confidential data.

Step 14. Insurance agent requests WTAM agent's and Tony's certificate from the CA, and resolves WTAM agent's and Tony's public key.

Step 15. Insurance agent verifies the WTAM agent's signature by WTAM agent's public key from step 14, data from step 4 and step 11, if successful, go to step 16, Otherwise, authentication fails, the WTAM agent's signature may indicate failure, or information being modified in transmission from WTAM to insurance agent, so insurance agent needs to throw an exception to tell WTAM that signature verification failure. WTAM may return to step 6 to re-run step 6 to step 12.

Step 16. Insurance agent verifies the Tony's signature by Tony's public key from step 14, data generated from step 4 and information summary from step 13, if successful, then go to step 17, Otherwise, throws an exception to tell WTAM that signature verification failure, WTAM may also return to step 6, re-run step 6 to step 12. Signature verification failure may be the following reasons: data are modified in WTAM agent's internal, or are modified in data transmission from WTAM to insurance agent, or Tony's failure contained in the customer's data is invalid, or the signature is incorrect, but it has already been used, so that Tony could have been avoided operating Tony's account data without the authorization.

Step 17. Insurance agent runs the requested service from customer Tony, and ends of this process.

It is necessary to note that insurance agent must verify the SDO data contain Tony's signature so that it can operate Tony's accounts, otherwise, the insurance agent will not carry out any operation of Tony's accounts. In other words, if the insurance agent wants to modify any critical data of customers, it must firstly get permission from customers; otherwise, it will not modify any critical data. If the data pass through a number of intermediate agents from producers to consumers, these agents need to sign the key and confidential SDO data, and then verify the signature in the opposite order. If there is any failure in the process of verification, the information from pro-

ducers to consumers is illegally modified by a third party during transmission.

5. Application of Data Security Solutions in Insurance Business

5.1. Application Environment of Insurance Business

Through analyzing and designing the underwriting business entity object and SOA service components, SOA service function model of insurance transaction system is shown in **Figure 3**.

Application situation and relation of general components, individual components and variant components, which are introduced in Section 3, are clearly expressed in **Figure 3**. Information system is made to be more flexible and effective by SOA service architecture.

5.2. Application Result of Solutions in Insurance Service Components

In Section 4 we use public-key encryption, private-key signature and public-key certificate to design a solution of data security issues. Data security issues are solved by this solution in SOA-FCA service components based

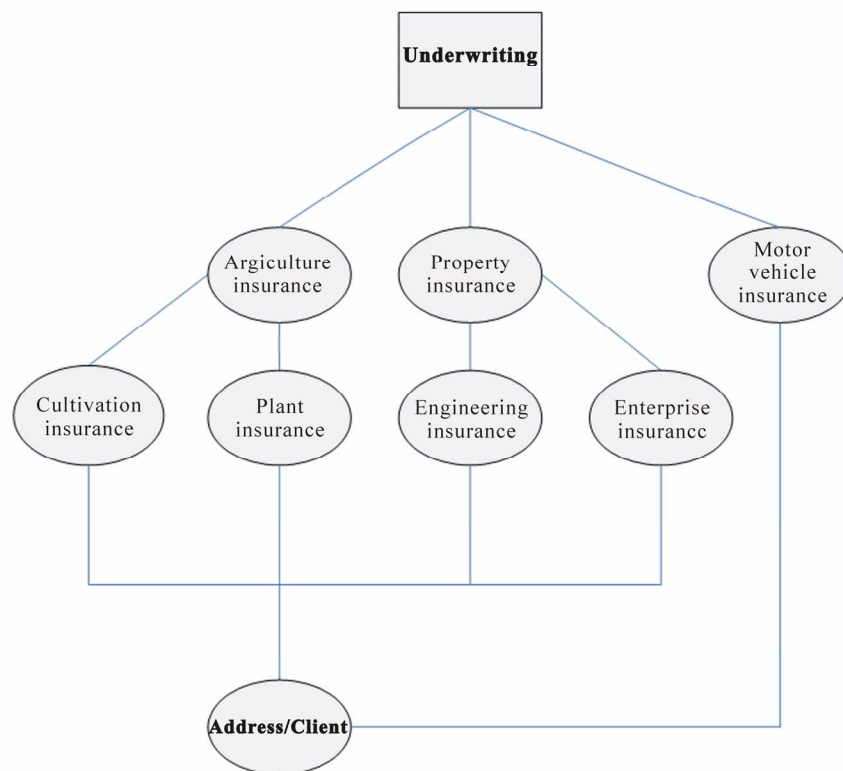


Figure 3. Insurance service function model.

SDO data model. We applied this solution into insurance information system so as to solve the confidentiality, integrity, and non-repudiation of SDO data model when SDO data is transmitted on the internet. In this way, it makes SOA-FCA service components more open, flexible, extensive, meanwhile it also improves and perfects its data security of SDO data model. Through application of data security solution, we solve security issues of SDO data transmitted in insurance information system and better data security issues in SOA-FCA service components based on SDO.

5.3. Performance Analysis

Only the encryption data is processed when the insurance system realizes, and the hardware configuration of the current serve has been greatly improved. The network transmission speed gets a large scale enhancement, although some extra processing and data causing the transmission traffics to increase are added in order to guarantee the security of SDO data, the system performance does not reduce greatly on the condition of without the extra processing and data.

6. Conclusions

SOA, compared with traditional application programs, brings a bigger openness, more flexibility and extendibility. But the openness also leads the security problem at the same time. The human and the machine can access to

the data from the service supplier according to the standard protocol at any time or place. SOA service components of Insurance have data security issues in the process of transmission. We solve these data security issues in SOA-FCA Service Components of insurance based SDO data model in our paper, design confidentiality, integrity, and non-repudiation realization scheme of SDO data, and then we carry on realization and analysis; validate the effectiveness of the scheme.

7. References

- [1] T. Erl, "Service-Oriented Architecture (SOA): Concepts, Technology, and Design," Prentice Hall PTR, Upper Saddle River, New Jersey, 2005.
- [2] B. Portier and F. Budinsky, "Introduction to SDO," IBM developerWorks, 2004.
- [3] M. John, "Security Models: Encyclopedia of Software Engineering," Wiley Press, 1994.
- [4] M. Benedikt, C. Cheeyonyong, W. F. Fan, *et al.*, "Capturing Both Types and Constraints in Data Integration," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, San Diego, California, USA, 2003, pp. 277-288.
- [5] D. Chaum and H. Van Antwerpen, "Undeniable Signatures," *Advances in Cryptology-CRYPTO'89*, LNCS 435. Springer-Verlag, Berlin, 1989.
- [6] S. W. Galbraith, "Invisibility and Anonymity of Undeniable and Confirmer Signatures," *Topics in Cryptology-CT-RSA'03*, LNCS 2612, Springer-Verlag, Berlin, 2003.

Reliable Multicast with Network Coding in Lossy Wireless Networks

Wei Yan, Shengyu Yu, Yueming Cai

Institute of Communications Engineering, PLA University of Science and Technology, Nanjing, China

E-mail: caiym@vip.sina.com

Received July 7, 2010; revised August 12, 2010; accepted September 15, 2010

Abstract

To reduce the feedbacks between access point and all nodes in lossy wireless networks, a clustered system model consisting of a cluster head and multiple common nodes is investigated. Network coding has been proposed for more efficient retransmissions in reliable multicast. However, in existing schemes the access point retransmits coded packets, which causes severe delay and considerable feedbacks. In this paper, an XOR scheme based on clustered model is presented. For this scheme, the cluster head broadcasts combined packets by XORing lost packets appropriately to recover lost packets locally. We also analyze the performance in terms of expected number of transmissions. Simulation results verify theoretic analysis. And our results show that proposed XOR offers a compromise between ARQ and random linear network coding.

Keywords: Multicast, Network Coding, ARQ, Wireless

1. Introduction

In wireless networks, multicast is an effective way to distribute information from a source to multiple destinations due to the wireless broadcast nature [1]. As fading is intrinsic in wireless links and different destinations may endure independent signal fading, it is hard to guarantee reliable transmissions for all destinations. Automatic repeat request (ARQ) is an existing approach to transmit information reliably and effectively over an error-prone network [2]. However, we can note that if a packet is not received successfully, it will be retransmitted. Using ARQ in reliable multicast, we can easily find that it may cause severe delay and considerable feedbacks, especially with a large number of destinations or high loss probability of broadcast channel. In this paper, we mainly focus on designing a practical and simple multicast protocol in lossy wireless networks.

Data packets are transmitted by store-and-forward mechanism in traditional networks. Network coding is the generalized approach to packet routing that allows an intermediate router to encode an outgoing packet by mixing multiple incoming packets appropriately [3]. Recently, network coding has been applied to wireless networks and received significant attention to improve multicast efficiency while guaranteeing reliability [4,5], such

as XOR, random linear network coding (RLNC) schemes. In [4], Katti *et al.* implemented a simple XOR-based testbed deployment in multi-hop wireless networks and showed a substantial network throughput over the current approach. In [6], transmission strategies were designed for a source and multiple destinations network by XORing a maximum set of lost packets from different receivers. In [7], the authors presented a multicast protocol with network coding exploiting a relay to further improve throughput. In [8], XOR Rescue (XORR) was proposed to solve the feedback overhead. The access point (AP) in XORR probabilistically estimated reception status based on the Bayesian-learning estimation technique. This scheme was hard to make a trade-off, as it depended on many dynamic parameters such as the number of users and wireless channel conditions [5] quantified the reliability benefit of RLNC in lossy wireless networks by computing the expected number of transmissions. But it did not consider the complexity and overhead of the feedback mechanism. And random linear network coding scheme was difficult to implement yet. For energy-limited wireless networks such as wireless sensor networks (WSNs) or mobile ad-hoc networks (MANETs), a practical and simple reliable multicast protocol was more important for uninterrupted data transmission without replenishing batteries frequently.

In traditional reliable multicast of the lossy wireless

networks, all nodes send the feedback messages to the access point. In this paper, the nodes are divided into several clusters. Only the cluster head sends the feedback to the access point, which can greatly reduce the amount of feedbacks between the access point and all nodes. Moreover, to take full advantage of the feedback messages from common nodes, the cluster head combines lost packets appropriately to help common nodes recover lost packets locally. We present an XOR scheme based on clustered model and analyze the performance in terms of expected number of transmissions. And we select ARQ and RLNC in simulation results for comparison.

This paper is organized as follows. Section 2 provides the system model and protocol description in detail. In Section 3, we present some theoretical analysis on the performance of ARQ and XOR. In Section 4, simulation results and discussions are presented. Finally, we conclude the conclusions in Section 5.

2. System Model and Protocol Description

In a typical data multicast transmission from AP to a lot of nodes, the nodes are divided into several clusters. As depicted in **Figure 1**, our system model is the scenario where the AP broadcasts the packets to a single cluster, which consists of a cluster head (CH) and K common nodes (CNs). The cluster head takes responsibility to deliver the packet to common nodes in the cluster. Namely, common node can not communicate with the other common nodes and communication links only exist between the CH and CNs. The AP can be considered as an unmanned aerial vehicle (UAV) or stratospheric telecommunication platform, which conveys the information to the nodes on the ground. Due to high signal attenuation, communications from the AP to the nodes suffer from high loss rates. However, the communications among the nodes on the ground always experience good channel quality. So the nodes on the ground can cooperate together to recover lost packet locally.

A three-phase transmission mechanism for reliable packet delivery from the AP to all nodes is considered in

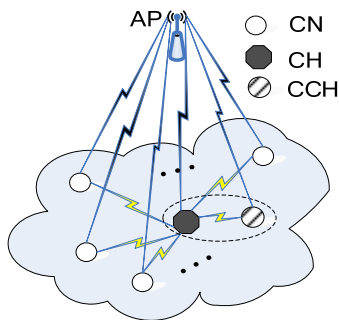


Figure 1. System model.

this paper. In the first phase, the AP broadcasts a sufficient number of packets to all nodes in the cluster such that every packet is received by at least one of the nodes. For each packet, every CN sends an ACK (ACKnowledgment) message or NACK (Negative ACKnowledgment) message to the CH after the AP's transmission. If there is at least one of successful receivers in the cluster, the CH sends an ACK message to the AP. If none of the nodes receives the packet successfully, the CH sends a NACK message to the AP and then the AP retransmits the lost packet. The second phase depends on whether the CH receives the all packets successfully. If the CH receives the all packets successfully, the second phase has already achieved. If not, the CH chooses cooperative cluster head (CCH) from the nodes which receive correctly. Then the CCHs send corresponding packets to the CH until the CH has the all packets. During the third phase, the CH helps all CNs recover the lost packets.

For ARQ, the CH multicasts the corresponding lost packets such that all CNs receive the all packets in the third phase. That is, one lost packet transmission is completed if every CN has this packet.

For XOR, the CH multicasts the combined packets by XORing different lost packets appropriately in the third phase. Let M denote the packet number of a data block. After the first phase, the CH sets up a feedback matrix $\mathbf{F}_{K \times M}$ according to the ACKs/ NACKs from the CNs, where $\mathbf{F}(i, j); i = 1, 2, \dots, K, j = 1, 2, \dots, M$ denotes whether CN i receives packet j successfully. If node i receives packet j successfully, $\mathbf{F}(i, j) = 0$ and if not, $\mathbf{F}(i, j) = 1$. According to the feedback matrix, the CH forms a combined packet by XORing a maximum set of the lost packets from different common nodes. In this way, the number of the packets for transmission from the CH to all common nodes is reduced. For instance, as shown in **Figure 2**, a feedback matrix for three common nodes and data block size $M = 9$ is given. The combined packets are $1 \oplus 3$, $4 \oplus 5 \oplus 6$, $7 \oplus 8$ and 9. Hence, Only 4 packets need to be sent, compared to 8 transmissions without network coding.

We define C as the set of packet sequences for a new combined packet. The set R denotes the searched rows. The set E denotes the packet sequences avoiding the decoding failure which can't be chosen as an element of the set C . For K CNs and data block size M , the

$$\mathbf{F}_{3 \times 9} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 2. An example of feedback matrix for three common nodes and $M = 9$.

detailed algorithm for general combination of lost packets is given in **Algorithm 1**.

Algorithm 1:

Input: $\mathbf{F}_{K \times M}$

$C \leftarrow \emptyset$ $E \leftarrow \emptyset$ $R \leftarrow \emptyset$

Find a row i of $\mathbf{F}_{K \times M}$ where $\max\{\text{sum}(\mathbf{F}(i,:))\}$

Find a column j of the row i with $\mathbf{F}(i,j)=1$, and
 $C \leftarrow C \cup \{j\}$

foreach $i \in \{1, 2, \dots, K\}$ do

if $\mathbf{F}(i,j)=1$ then $R \leftarrow R \cup \{i\}$

foreach $i \in R$ do

foreach $j \in \{1, 2, \dots, M\}$ do

if $\mathbf{F}(i,j)=1$ then $E \leftarrow E \cup \{j\}$

foreach $n \in \{1, 2, \dots, M\}$ do

if $n \in \{1, 2, \dots, M\}/E$ then

foreach $i \in \{1, 2, \dots, K\}/R$ do

if $\mathbf{F}(i,n)=1$ then $C \leftarrow C \cup \{n\}$ $R \leftarrow R \cup \{i\}$

foreach $j \in \{1, 2, \dots, M\}$ do

if $\mathbf{F}(i,j)=1$ then $E \leftarrow E \cup \{j\}$

Output: C

Through clustering the nodes, the amount of feedbacks can greatly reduce between all nodes and the AP. For ease of performance analysis, in this paper, we assume all ACKs/NACKs are instantaneous and reliable. Transmissions from the AP, over a wireless link to all nodes on the ground, are subject to random losses. We assume that losses for all nodes are described by independent Bernoulli processes with parameter p_0 . Similarly, local transmissions between CH and CNs are subject to random losses, where the loss process is Bernoulli with parameter p_1 . In practice, the channels between AP and all nodes have relatively lower channel gains than the corresponding channels among the nodes, that is $p_0 \ll p_1$.

3. Performance Analysis

In this section, we provide theoretical analysis on the number of transmissions per packet for ARQ and XOR. With three phase model, N , the number of transmissions per packet, includes three components of X , Y and Z . X , Y and Z separately denote the number of transmissions in three different phase. For ARQ, $N = X + Y + Z$. For XOR, N is given by $N = (X + Y + Z)/M$.

3.1. Distribution of the Number of Transmissions

1) ARQ Performance

In the first phase, the probability distribution function of X is given by [5]:

$$P\{X \leq x\} = 1 - p_0^{(K+1)x} \quad (1)$$

In the second phase, we have

$$\begin{aligned} P\{Y \leq y\} &= p_0(1 - p_1^y) + (1 - p_0) \times 0 \\ &= p_0(1 - p_1^y) \end{aligned} \quad (2)$$

Let W denote the number of common nodes that have received a copy of the packet. The probability distribution of W can be expressed as

$$P\{W = w\} = \binom{K}{w} (1 - p_0)^w p_0^{K-w} \quad (3)$$

Hence, we can obtain that the probability distribution function of Z is

$$\begin{aligned} P\{Z \leq z\} &= \sum_{w=0}^K P\{W = w\} P\{Z \leq z | W = w\} \\ &= \sum_{w=0}^K P\{W = w\} (1 - p_1^z)^{K-w} \\ &= (1 - p_0 p_1^z)^K \end{aligned} \quad (4)$$

2) XOR Performance

The probability that a packet transmitted by the AP is received by at least one node in a cluster is given by $1 - p_0^{K+1}$. Similar to the first phase of RLNC [5], X has a negative binomial distribution with success probability $1 - p_0^{K+1}$. Therefore, the probability distribution of X is

$$P\{X = x\} = \binom{x-1}{M-1} (1 - p_0^{K+1})^M p_0^{(K+1)(x-M)} \quad (5)$$

In the second phase, the CH assigns the CCHs to transmit lost packets according to feedback matrix. Let a random variable U denote the number of packets successfully received by the CH. The probability distribution of Y can be computed as

$$P\{Y = y\} = \sum_{u=0}^M P\{Y = y | U = u\} P\{U = u\} \quad (6)$$

where

$$P\{Y = y | U = u\} = \binom{y-1}{M-u-1} (1 - p_1)^{M-u} p_1^{y-M+u} \quad \text{and}$$

$$P\{U = u\} = \binom{M}{u} (1 - p_0)^u p_0^{K-u}.$$

In the third phase, CH broadcasts the combined pack-

ets to CNs. When data block size M goes to ∞ , the number of transmissions is dominated by the CN which has the maximum lost packets [6]. For ease of analysis, we assume that at least one CN has lost M packets. Therefore, the expectation of Z is

$$\lim_{M \rightarrow \infty} \frac{1}{M} E[Z] = \frac{1}{1-p_1} \quad (7)$$

3.2. Asymptotic Analysis

1) ARQ Performance

The average number of transmissions in the first and second phase is separately given by:

$$E[X] = \sum_{x=0}^{\infty} P\{X > x\} = \frac{1}{1-p_0^{K+1}} \quad (8)$$

and

$$\begin{aligned} E[Y] &= \sum_{y=0}^{\infty} yP\{Y = y\} \\ &= \sum_{y=0}^{\infty} y[P\{Y \leq y\} - P\{Y \leq y-1\}] = \frac{p_0}{1-p_1} \end{aligned} \quad (9)$$

The average number of CNs receiving the packet successfully is

$$E[W] = \sum_{w=0}^W wP\{W = w\} = (1-p_0)K \quad (10)$$

To simplify the analysis, the number of CNs receiving the packet unsuccessfully is replaced by its mean value, i.e., each packet from CH is required by p_0K CNs. Based on the analysis [5], the expected number of transmissions is

$$\begin{aligned} \lim_{K \rightarrow \infty} E[Z] &= \frac{\ln p_0 K}{-\ln p_1} + \frac{\gamma}{-\ln p_1} + \frac{1}{1-p_1} + \frac{1}{\ln p_1} \\ &= \Theta(\log(p_0 K)) = \Theta(\log(K)) \end{aligned} \quad (11)$$

where γ is Euler's constant. Therefore, for ARQ, the expected number of transmissions per packet scales as $\Theta(\log(K))$.

2) XOR Performance

In the first phase, we can obtain

$$\lim_{M \rightarrow \infty} \frac{1}{M} E[X] = \frac{1}{1-p_0^{K+1}} \quad (12)$$

The expectation of U can be computed as

$$E[U] = \sum_{u=0}^M uP\{U = u\} = (1-p_0)M \quad (13)$$

For simplicity, we assume the number of packets suc-

cessfully received by the CH is $E[U]$. It is also obtained that

$$\lim_{M \rightarrow \infty} \frac{1}{M} E[Y] = \frac{p_0}{1-p_1} \quad (14)$$

Hence, for XOR, the expected number of transmissions per packet scales as $\Theta(1)$ according to (7), (12) and (14).

4. Simulations

In this section, simulation results on the expected number of transmissions for different schemes are discussed. For ARQ, **Figure 3** shows the expected number of transmissions for a wide range of values of K and different loss probabilities. The horizontal axis shows $\log_2(K)$, and it can be seen that the four curves are very close to straight line. Hence, the simulation results validate the logarithmic scale. For XOR, the expected number of transmissions for data block size $M = 128$ versus different loss probabilities is shown in **Figure 4**. The expected number of transmissions approaches a constant value with increasing the number of CNs.

Figure 5 and **Figure 6** show that the expected number of transmissions for different schemes with $p_0 = 0.5, p_1 = 0.05$ And $p_0 = 0.5, p_1 = 0.2$, respectively. As seen, XOR offers a compromise between ARQ and RLNC. An interesting observation is that the expected number of transmissions for XOR is almost close to ARQ with 32 CNs. When $K = 8$, XOR can obtain the best performance gain. For different probabilities and data block size, an open problem arises as to how many CNs to make XOR achieve maximal performance gain over ARQ. For XOR and RLNC with $M = 32, 64, 128$, it can be seen that the expected number of transmissions reduces with increasing data block size. Similar to RLNC, XOR has the same result that a moderate block size suffices to obtain the advantage applying network coding.

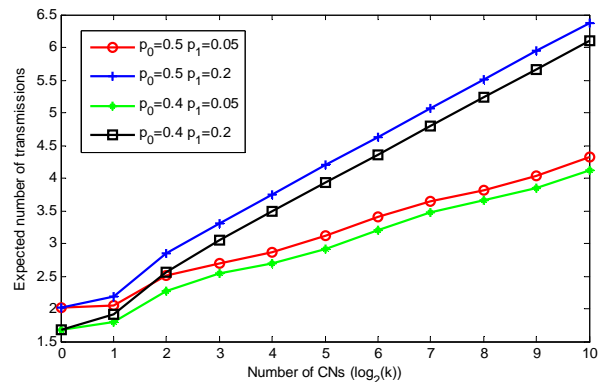


Figure 3. The performance of ARQ for different probabilities.

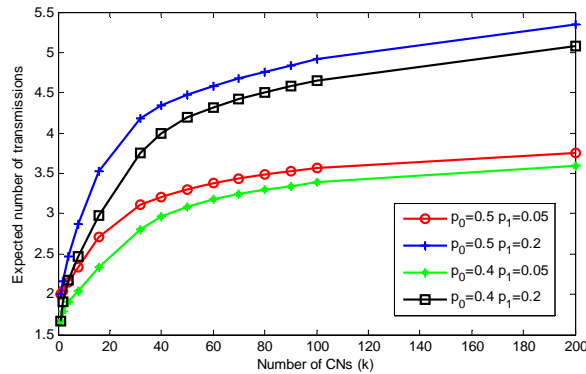


Figure 4. The performance of XOR for different probabilities.

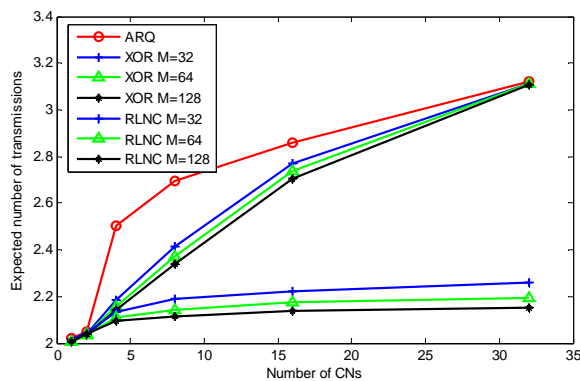


Figure 5. Expected number of transmissions for different schemes with $p_0 = 0.5$, $p_1 = 0.05$.

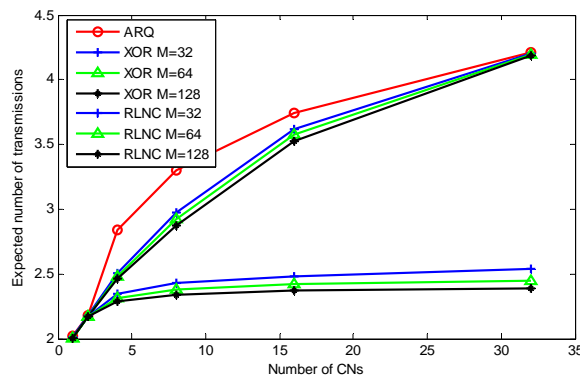


Figure 6. Expected number of transmissions for different schemes with $p_0 = 0.5$, $p_1 = 0.2$.

5. Conclusions

In this paper, we investigated network coding for reliable multicast and presented an XOR scheme based on clustered model in lossy wireless networks. Instead of access point, cluster head retransmits the combined packets to

recover the lost packets for multiple common nodes. We provide theoretical analysis in terms of expected number of transmissions and extensive simulations. Simulation results show that proposed XOR can achieve maximal performance gain with some common nodes for different probabilities and data block size. And our XOR can offer a compromise between ARQ and RLNC. It must be emphasized, however, that clustered network was considered in this paper. In the future, the extension to a decentralized network is an interesting topic in lossy wireless networks, which requires further research.

6. Acknowledgements

This work is supported by the Important National Science & Technology Specific Project under Grant 2010 ZX03006-002-04, the National Natural Science Foundation of China under Grant No. 60972051, the National 863 Project of China under Grant No. 2009AA01Z235 and the Project of Natural Science Foundation of Jiangsu province under Grant No. BK2010101.

7. References

- [1] P. Chaporkar and S. Sarkar, "Wireless Multicast: Theory and Approaches," *IEEE Transactions on Information Theory*, Vol. 51, No. 6, 2005, pp. 1954-1972.
- [2] S. Lin and D. Costello, "Error Control Coding," Prentice Hall, Upper Saddle River, New Jersey, 2004.
- [3] R. Ahlswede, N. Cai, R. Li and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, Vol. 46, No. 4, 2000, pp. 1204-1216.
- [4] S. Katti, H. Rahul, W. Hu, D. Katabi, et al., "XORs in the Air: Practical Wireless Network Coding," *Proceedings ACM SIGCOMM*, Pisa, Italy, September 2006, pp. 497-510.
- [5] M. Ghaderi, D. Towsley and J. Kurose, "Reliability Gain of Network Coding in Lossy Wireless Networks," Department of Computer Science, University of Calgary, Technical Report: TR-07-08, January 2008.
- [6] D. Nguyen, T. Nguyen and B. Bose, "Wireless Broadcast with Network Coding," Technical Report: OSU-TR-2006-06, Oregon State University, June 2006.
- [7] P. Fan, Z. Chen, W. Chen and K. B. Letaief, "Reliable Relay Assisted Wireless Multicast Using Network Coding," *IEEE Journal of Selected Areas in Communications*, Vol. 27, No. 5, 2009, pp. 749-762.
- [8] F. C. Kuo, K. Tan, X. Y. Li, et al., "XOR Rescue: Exploiting Network Coding in Lossy Wireless Networks," *IEEE Sensor, Mesh and Ad Hoc Communications and Networks Conference (SECON)*, Rome, Italy, June 2009, pp. 1-9.

Using Least Squares Support Vector Machines for Frequency Estimation

Xiaoyun Teng, Xiaoyi Zhang, Hongyi Yu

Information Science and Technology Institute, Zhengzhou, China

E-mail: ieukey@163.com

Received July 6, 2010; revised August 9, 2010; accepted September 11, 2010

Abstract

Frequency estimation is transformed to a pattern recognition problem, and a least squares support vector machine (LS-SVM) estimator is derived. The estimator can work efficiently without the need of statistics knowledge of the observations, and the estimation performance is insensitive to the carrier phase. Simulation results are presented showing that proposed estimators offer better performance than traditional Maximum Likelihood (ML) estimator at low SNR, since classification-based method does not have the threshold effect of nonlinear estimation.

Keywords: Carrier Recovery, LS-SVM, Pattern Recognition

1. Introduction

In digital communication systems with burst transmission, carrier recovery within each information burst is a critical issue. The estimation of carrier frequency in additive noise is one of the very important problems in the theory and applications of digital signal processing. Various techniques have been proposed for carrier frequency recovery [1-5]. The estimators in [1] are based on a maximum likelihood criterion, which is known to be an excellent estimator, but it suffers from the threshold effect of nonlinear estimators. Frequency estimation in colored noise is addressed in [6] and [7], which model the colored noise as an AR or MA process. However, most of above estimators require the statistics knowledge of the observations, such as, probability density function (pdf), autocorrelation, etc.

A Support Vector Machine (SVM) [8] uses training data as an integral element of the function estimation model as opposed to simply using training data to estimate parameters of an a priori model using maximum likelihood [9]. The SVM has the advantage over traditional learning approaches in terms of performance, complexity and convergence. SVMs have been widely used in solving classification and function estimation problems. Recently, SVM has been introduced to communication systems as a new method for channel equalization [9,10] and multiuser detector in CDMA communications [11]. The least squares support vector machines

(LS-SVM) involves solving linear equations instead of solving the quadratic programming problem required in the original SVM. In this paper, we view frequency estimation as a pattern recognition problem, and propose a different frequency estimator based on LS-SVM.

2. Signal Model

Using complex-envelope notation, the observed signal samples are expressed by

$$\begin{aligned} r_n &= a(n)e^{j(\omega n + \theta)} + v(n), \\ n &= 0, 1, \dots, N_p - 1, \dots, N - 1 \end{aligned} \quad (1)$$

where ω is the unknown carrier frequency normalized to sampling frequency f_s , for the sake of simplicity, $f_s = 1$, θ is an initial random phase, $v(n)$ are additive noise samples. a_n is the normalized transmitted BPSK symbol, i.e., $a_n = \pm 1$. We consider scenarios where the signal a_n is known, i.e., a training sequence is transmitted for carrier recovery.

Define the following vector

$$\mathbf{r} = [r_0, \dots, r_{N_p-1}, \dots, r_{N-1}]^T$$

$$\mathbf{A} = [a_0, \dots, a_{N_p-1}, \dots, a_{N-1}]^T$$

$$\Psi_N(\omega) = \text{diag} \{1, e^{j\omega}, \dots, e^{j(N_p-1)\omega}, \dots, e^{j(N-1)\omega}\}$$

$$\mathbf{v} = [v(0), v(1), \dots, v(N-1)]^T \quad (2)$$

The signal model can be arranged in matrix form as

$$\mathbf{r} = e^{j\theta} \mathbf{\Psi}_N(\omega) \mathbf{A} + \mathbf{v} \quad (3)$$

3. SVM Based Frequency Estimation

3.1. SVM Introduction

SVM developed by Vapnik is a new type of learning machines which has a high generalization performance and sparse solution. It replaces empirical risk minimization by structural risk minimization (SRM). The goal of SVM is to find the hyperplane that maximizes the minimum distance between any point and the hyperplane. The idea of SVM can be expressed as follows.

Consider (x_i, y_i) , $i = 1, 2, \dots, N$ be a linearly separable training set, where $x \in R^d$ and $y \in \{-1, +1\}$, which can be separated by the hyperplane satisfying $w^T x + b = 0$, where w is the weight vector and b is the bias. If this hyperplane maximizes the margin, then we need to solve the following optimization problem.

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2} \|w\|^2 \\ &\text{subject to} \quad y_i (w \cdot x_i + b) \geq 1 \end{aligned} \quad (4)$$

For the inputs data that is not separable, SVM uses soft margins that can be expressed as follows, by introducing the non-negative slack variables $\xi_i, i = 1, \dots, N$:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i^k \\ &\text{subject to} \quad y_i (w^T x_i + b) \geq 1 - \xi_i \end{aligned} \quad (5)$$

Data points are penalized if they are misclassified. The parameter C controls tradeoff between the complexity of the model and the classification errors.

To construct nonlinear decision functions, SVM maps the training data nonlinearly into a higher-dimensional feature space via a kernel function, and constructs a separating hyperplane with maximum margin there. The kernel function is

$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) \quad (6)$$

The typical kernel functions include RBF $K(x, y) = \exp(-\|x - y\|^2 / 2\sigma^2)$ and polynomial $K(x, y) = (1 + x \cdot y)^d$.

We prefer LS-SVM over other models of SVM, for it offers a fast method for obtaining classifiers with good generalization performance in many applications. In LS-SVM, an equality constraint-based formulation is involved instead of the convex quadratic programming

(QP) problem in (5).

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i^2 \\ &\text{subject to} \quad y_i (w^T x_i + b) = 1 - \xi_i \end{aligned} \quad (7)$$

To solve this problem Lagrange multipliers $(\alpha_i, i = 1, \dots, l; \alpha_i \geq 0)$ are used:

$$L_p = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i^2 - \sum_{i=1}^N \alpha_i [y_i (w^T x_i + b) - 1 + \xi_i] \quad (8)$$

The KKT optimality conditions are given by

$$\begin{cases} \frac{\partial L_p}{\partial w} = 0 \rightarrow \left(w - \sum_{i=1}^l \alpha_i y_i \varphi(x_i) \right) = 0 \\ \frac{\partial L_p}{\partial b} = 0 \rightarrow \sum_{i=1}^l \alpha_i y_i = 0 \\ \frac{\partial L_p}{\partial \xi_i} = 0 \rightarrow \alpha_i - C \xi_i = 0 \\ \frac{\partial L_p}{\partial \alpha_i} = 0 \rightarrow y_i (w^T \cdot \varphi(x_i)_i + b) - 1 + \xi_i = 0 \end{cases} \quad (9)$$

Optimal decision function (ODF) is then given by:

$$f(x) = \text{sign} \left[\sum_{i=1}^n \alpha_i y_i K(x_i \cdot x) + b \right] \quad (10)$$

3.2. SVM-FEA

To the best of our knowledge, SVM has not been implemented yet for parameter estimation in digital communication systems. We introduce SVM as a new method for frequency estimation, by building the following frequency estimator

$$\hat{\omega} = \arg \min_{\omega \in \Omega} \left\{ \sum_{n=1}^N \Lambda(n; \omega) \right\} \quad (11)$$

where the cost function

$$\Lambda(n; \omega) = |r(n) \cdot e^{-j(\omega n + \theta)} - s(n)| \quad (12)$$

In such a scenario, the frequency estimation problem can be transferred to a pattern recognition problem. The optimal estimate of ω can be attained by minimizing the classification error.

In a general way, the carrier phase θ existing in the received signal samples is unknown. Thus, the ideal ML detector is hard to handle the classification problem in (12). The powerful LS-SVM technique is applied in this paper.

To fit the support vector machine model, the output of the channel can be grouped into vectors

$$x(n) = [\text{Re}\{r(n) \cdot e^{-j\omega n}\}, \text{Im}\{r(n) \cdot e^{-j\omega n}\}] \quad (14)$$

Where $\text{Re}\{\cdot\}$ and $\text{Im}\{\cdot\}$ mean the real and image part of $\{\cdot\}$, separately. For training purposes, taking $x(n)$ as the input sequence of SVM, and the transmitted symbol $a(n)$ to be the desired output sequence.

This model of SVM-based frequency estimator, that we call SVM-FEA, is illustrated in **Figure 1**.

The optimal estimator cannot be found in a single step because the input data has the unknown term $e^{-j\omega n}$. So an effective searching process is needed to achieve the frequency estimation. The procedure, which is similar to the coarse search and fine search routine in ML algorithm, is particularized as follows.

1) Choose a set of ω values according to a appropriate interval λ_1 , i.e.,

$$\omega_1 = 0, \omega_2 = \omega_1 + \lambda_1, \omega_3 = \omega_2 + \lambda_1, \dots$$

2) Construct the input sequence \mathbf{x}_w using each ω value; for example,

$$\mathbf{x}_{w_1}(n) = \begin{bmatrix} \text{Re}\{r(n) \cdot e^{-j\omega_1 n}\}, \text{Im}\{r(n) \cdot e^{-j\omega_1 n}\} \end{bmatrix}$$

solve the QP problem and obtain the decision function.

3) Classify \mathbf{x}_w and identify the ω that minimizes the classification error, get a approximate estimate $\hat{\omega}$.

4) Set a refined interval λ_2 , get a new set of ω values between $\hat{\omega} \pm \lambda_1$, do 2) and 3) and get a fine estimate of ω .

4. Simulation Results

Computer simulations have been run to check the analytical results of the previous sections. We will observe the average estimate

$$E[\hat{f}] = \frac{1}{N_m} \sum_{i=1}^{N_m} \hat{f}_i$$

and the mean square error (MSE) of the estimate

$$MSE[\hat{f}] = \frac{1}{N_m} \sum_{i=1}^{N_m} (\hat{f}_i - f)^2$$

Linear LS-SVMs are used with 15 to 50 data samples. In such a scenario, C is the only parameter to be chosen by the user during LS-SVM training, C is the upper

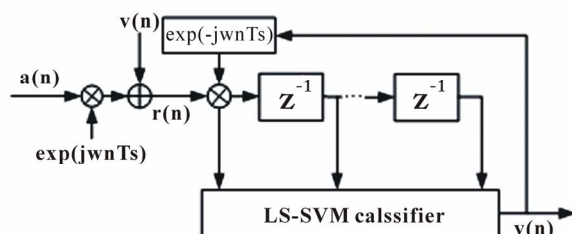


Figure 1. SVM based frequency estimator structure.

responds to assigning a higher penalty to classification errors. Simulation results showed that it has been more robust to set C between 0.1 and 10. Specifically, we fix $C = 5$.

Figure 2 illustrates the average estimates versus f when $\text{SNR} = 7$ dB, $N = 20$. The ideal line $E[\hat{f}] = f$ is also shown for comparison. The curves show that the range over which the estimates are unbiased is about $(-0.4, 0.4)$.

We compare the performance of the proposed SVM-FEA and the typical ML algorithm in AWGN channel. MSE of the estimates are compared with the CRLB as follows

$$CRLB_{\hat{\omega}} = \begin{cases} \frac{3}{\text{SNR} \cdot N(N-1)(2N-1)}, & \text{phase is known} \\ \frac{6}{\text{SNR} \cdot N(N^2-1)}, & \text{phase is unknown} \end{cases}$$

We first consider the case that the carrier phase is known, so ideal ML detector can be directly used to handle the classification problem in (13). Thus a ML detector based frequency estimator can be derived by (12). The curves of MSE versus SNR are shown in **Figure 3**. The simulation results are attained when $f = 0.34$, $N = 18$. The performance curves of the ML single tone, i.e. DA, frequency estimation algorithms with different N proposed in [1] are also shown for comparison. The likelihood functions are computed through FFT. For the sake of simplicity, only the coarse search is made and the FFT length is 32768. It can be seen in **Figure 2** that the MSE performance of the SVM-FEA is close to ML detector based frequency estimator in this case. And both classification based frequency estimation algorithm outperform the traditional ML estimator at low SNR (< -1 dB).

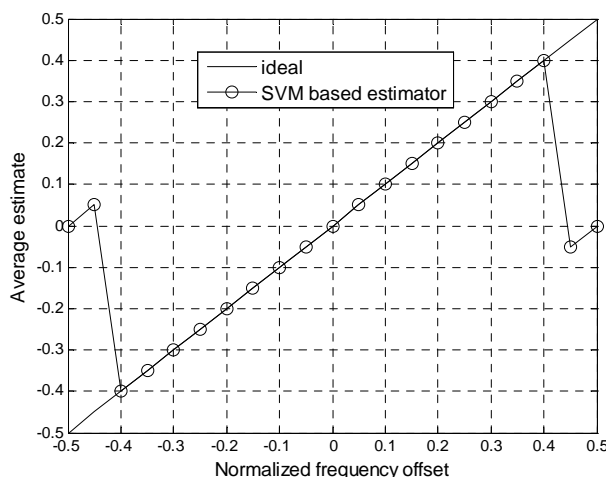


Figure 2. Average frequency estimate versus f .

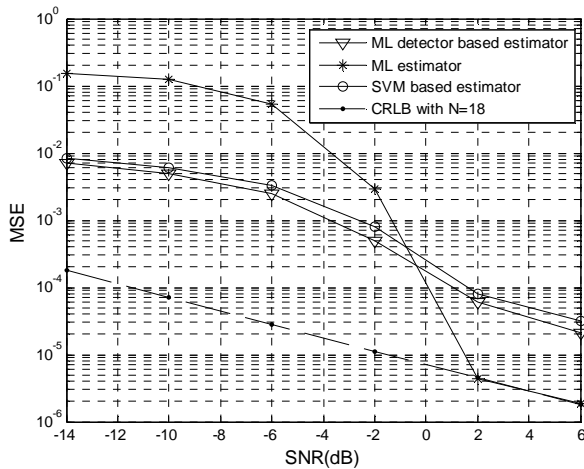


Figure 3. MSE versus SNR when phase is known.

Figure 4 illustrates the comparison of the MSE performance of the SVM-FEA and ML estimator when the phase is unknown. The simulation results are attained when $f = 0.34$, $N = 20$ and $N = 30$. Since all the data points in the input vectors have the same constant caused by the carrier phase, SVM-FEA shows almost the same performance in both cases. The performance of SVM-FEA improves with the increasing of the data length, which is not as remarkable as that of ML estimator (the cross of two curves change from 0 dB to -4 dB).

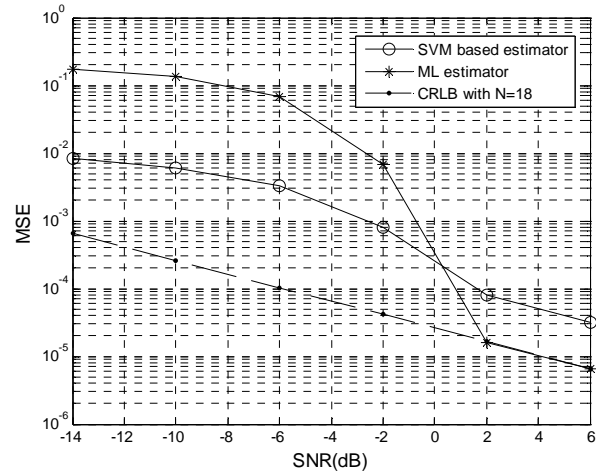
It is noticed that although SVM-FEA present a significant improvement over the ML estimator at low SNR, it can not reach the CRLB even at high SNR. The reason is that classification based frequency estimation algorithm identifies the estimate of ω corresponding to the classification error, which is insensitive to a very small frequency change to a certain extent.

5. Conclusions

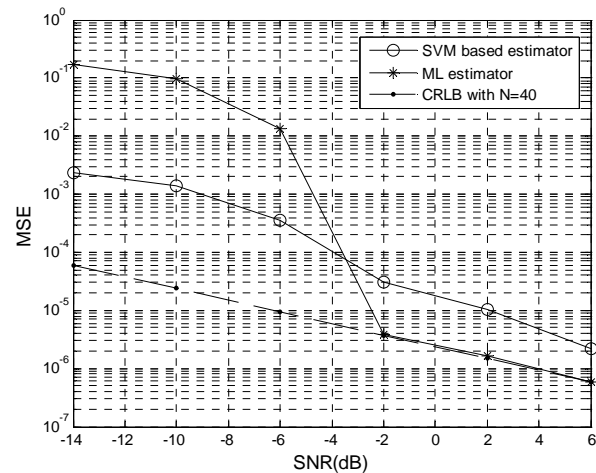
In this paper, we formulated frequency estimation of digital communication signals as a classification problem, and applied SVM technique to solve it. A primary searching routine has been proposed to find the optimal frequency estimate.

Simulations have shown that the SVM provides a robust method for frequency estimation with following attractive features: The estimator can work efficiently without the need of statistics knowledge of the observations, and the estimation performance is insensitive to the carrier phase; it shows a better performance than traditional ML estimator at low SNR, for SVM-FEA has not the threshold effect of nonlinear estimation.

A main drawback of the proposed algorithm is the



(a)



(b)

Figure 4. MSE versus SNR when phase is unknown.

high computational cost, which can be reduced by introducing faster optimization techniques and improving our searching routine. Future work will also be carried out on the generalization of the proposed procedure to multi-level modulations and other channel conditions, such as fading channel and colored noise conditions.

6. References

- [1] D. C. Rife and R. R. Boorstyn, "Single-Tone Parameter Estimation from Discrete-Time Observations," *IEEE Transactions on Information Theory*, Vol. 20, No. 5, 1974, pp. 591-598.
- [2] M. Morelli and U. Mengali, "Feedforward Frequency Estimation for PSK: A Tutorial Review," *European Tran-*

- sactions on Telecommunications, Vol. 9, No. 2, 1998, pp. 103-116.
- [3] W. Y. Kuo and M. P. Fitz, "Frequency Offset Compensation of Pilot Symbol Assisted Modulation in Frequency Flat Fading," *IEEE Transactions on Communications*, Vol. 45, No. 11, 1997, pp. 1412-1416.
 - [4] U. Mengali and M. Morelli, "Data-Aided Frequency Estimation for Burst Digital Transmission," *IEEE Transactions on Communications*, Vol. 45, No. 1, 1997, pp. 23-25.
 - [5] Y. Wang, E. Serpedin and P. Ciblat, "Optimal Blind Carrier Recovery for MPSK Burst Transmissions," *IEEE Transactions on Communications*, Vol. 51, No. 9, 2003, pp. 1571-1581.
 - [6] D. N. Swingler, "Approximate Bounds on Frequency Estimates for Short Cissoids in Colored Noise," *IEEE Transactions on Signal Processing*, Vol. 46, No. 5, 1998, pp. 1456-1458.
 - [7] J. Viterbi and A. M. Viterbi, "Nonlinear Estimation of Psk-Modulated Carrier Phase with Application to Burst Digital Transmissions," *IEEE Transactions on Information Theory*, Vol. 29, No. 4, 1983, pp. 543-551.
 - [8] V. N. Vapnik, "Statistical learning theory," John Wiley & Sons, Inc., New York, 1998.
 - [9] D. J. Sebald, "Support Vector Machine Techniques for Nonlinear Equalization," *IEEE Transactions on Signal Processing*, Vol. 48, No. 11, 2000, pp. 3217-3226.
 - [10] S. Chen, S. Gunn and C. Harris, "Decision Feedback Equalizer Design Using Support Vector Machines," *Vision, Image and Signal Processing*, Vol. 147, No. 3, 2000, pp. 213-219.
 - [11] F. Perez-Cruz, A. Navia-Vazquez, P. Alarcon-Dianna and A. Artes-Rodriguez, "SVC-Based Equalizer for Burst TDMA Transmission," *Signal Processing*, Vol. 81, No. 6, 2001, pp. 1681-1693.
 - [12] J. G. Proakis, "Digital Communications," 4th Edition, McGraw-Hill, New York, 2001.

A Turbo Decoder Included in a Multi-User Detector: A Solution to be Retained

Sylvie Kerouédan^{1,2}, Makram Touzri¹, Patrick Adde^{1,2}, Samir Saoudi^{1,2}

¹*Institut Télécom, Télécom Bretagne, UMR CNRS 3192 Lab-STICC, Brest, France*

²*Université Européenne de Bretagne (UEB), France*

*E-mail: sylvie.kerouedan@telecom-bretagne.eu, mak_fr@yahoo.fr, patrick.adde@telecom-bretagne.eu,
samir.saoudi@telecom-bretagne.eu*

Received July 13, 2010; revised August 18, 2010; accepted September 20, 2010

Abstract

This paper deals with the presentation of different multi-user detectors in the Universal Mobile Telecommunications System (UMTS) context. The challenge is always to optimize the compromise between performance and complexity. Compared with the solution commonly used today, the rake detector, successive interference cancellation (SIC) detector has better performance despite its higher complexity. Our innovative solution proposes joining detector and channel turbo decoder to get a significant gain in terms of performance. Furthermore, when detection and decoding are implemented in a single function, complexity does not increase much.

Keywords: CDMA, Turbocode, Multi-User Detection, Turbo-CDMA

1. Introduction

There is today a high demand for increasing the number of users in wireless communication systems, and sharing techniques have been implemented. When many users have to share the same spectrum resource, multi-user detection (MUD) algorithms have to be implemented in the receiver. Well-known MUD techniques use time, frequency or code division to share resources between users. In our study we focus on one technique: code division multiple access (CDMA). **Figure 1** summarizes the principle of spread spectrum and code division in a CDMA system in order to give some key notations useful for reading the paper.

On the other hand, outstanding channel coding algorithms, such as turbo techniques, can reach very high data rates, or can offer the possibility of low power emission. Joining a MUD receiver and a turbo decoder in an iterative process has been seen as a good way to merge the advantages of the two techniques. This association can be done in different ways:

- ◆ separately, which means doing first a few stages of SIC-receiver (Successive Interference Cancellation) and then several iterations of a turbo decoder (SIC-turbo configuration),
- ◆ jointly, which means that the turbo decoder is

an inner part of the SIC unit (turbo-SIC configuration).

The second proposal is very interesting in terms of performance but seems to be very complex due to the presence of a turbo decoder in the core of the SIC cell. At present time, in the universal mobile telecommunications system uplink context, the solution generally retained in the base station is the classical rake detection followed by a bank of channel decoders. The goal of our study is to show that the detector and turbo decoder association can be competitive against the simplicity of the classical solution. We compare different architectures: the classical rake receiver (CONFIG 1) [1], the SIC-turbo receiver (CONFIG 2) and the turbo-SIC receiver (CONFIG 3) [2]. The three architectures have been described in C language to get performance curves, and then in VHDL to be synthesized with Synopsys Design Analyzer on ST 90 nm target technology to get complexity data. This study and the results have been widely described and justified in [3].

The paper is organized as follows: in Section 2 we describe the implementation of a successive interference cancellation detector; in Section 3, the different associations of channel decoding and multi-user detector are explained; and the last section is dedicated to the comparisons of the different architectures.

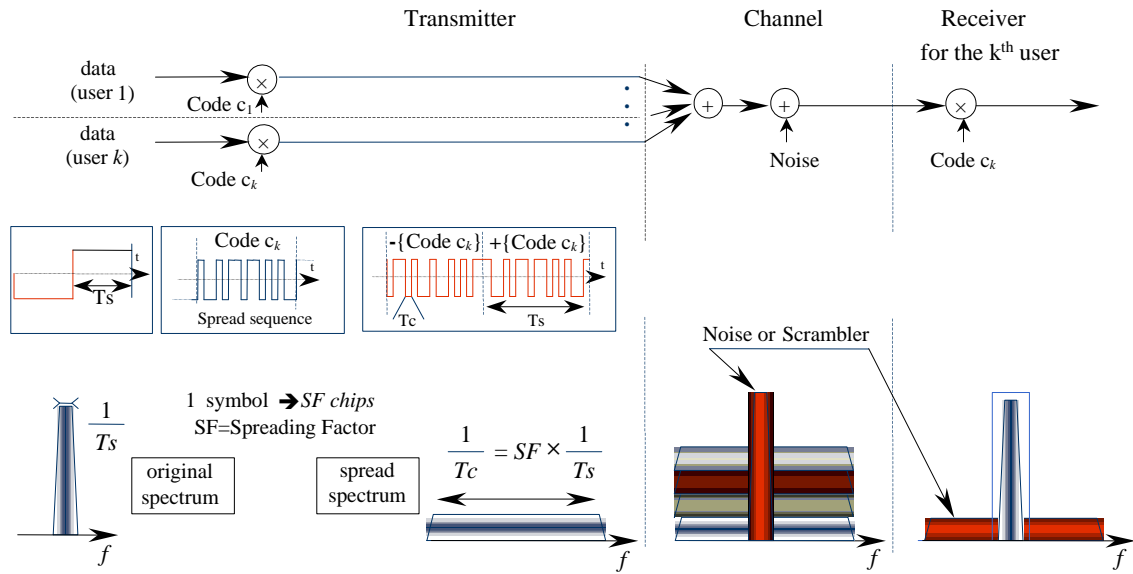


Figure 1. Principle of spread spectrum and code division in CDMA system.

2. Successive Interference Cancellation Detection

2.1. Generalities

In a multi-user context, the goal of interference cancellation is to eliminate interference due to the current user by estimating the transmitted signal and then subtracting it from the received signal. The successive interference cancellation (SIC) detector is based on serial processing of the estimation and the interference cancellation. The SIC detector is a good compromise between performance and complexity compared with parallel or hybrid interference cancellation detectors [4].

SIC structure, shown in **Figure 2**, is composed of M steps of K interference cancellation units (ICU), where K is the total number of users. Inside each ICU, we can find as demonstrated in **Figure 3**:

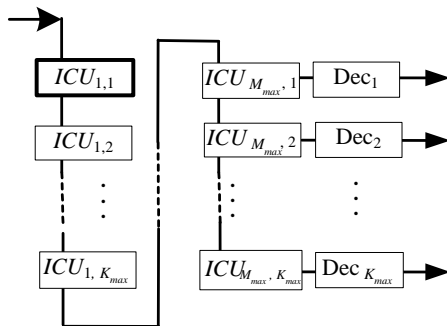


Figure 2. Classical structure of a SIC detector.

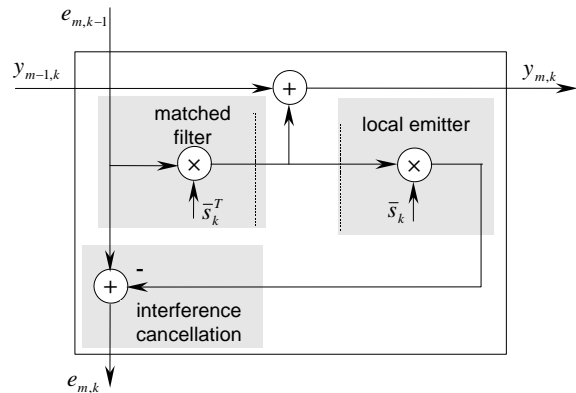


Figure 3. Synoptic of the internal structure of an ICU in the case of a transmission over a Gaussian channel.

- ◆ a matched filter linked to the current user k ,
- ◆ a local emitter to regenerate the interference due to the current user,
- ◆ an operator which computes the residual signal $e_{m,k}$ after current interference cancellation.

This residual signal is then sent to the following ICU _{$m,k+1$} . The internal structure of an ICU can be more or less complex depending on whether channel decoding is implemented inside or not.

2.2. Implementation of an ICU

In this section, we describe the implementation of the different blocks of the ICU _{m,k} as shown in **Figure 2**. First it is important to notice that the system is clocked either by the chip rhythm (period T_c) or the symbol rhythm

(period T_s) (cf. **Figure 1**).

2.2.1. Matched Filter Block

Figure 4 shows the architecture of the matched filter. The inputs of this block for the current user k ($k = 1, 2, \dots, K$) at the step m ($m = 1, 2, \dots, M$) are

- ♦ the residual signal $e_{m,k-1}$ from previous $k-1$ user,
- ♦ the k user code (scrambling $s_k^{(s)}$ and spreading $s_{[I]k}^{(w)}$ codes for data link),
- ♦ the estimated channel coefficients $c_{k,l}$,
- ♦ the delays $t_{k,l}$ coming from the L channel paths.

The output of this block is the residual estimation of the received symbol. For each branch the input sequence of SF chips is multiplied by the conjugate of the scrambling code to select the current user k . The resulting sequence is then despread (step 2). A multiplication by the channel coefficients corrects the effect of the multiple paths (step 3). The result is then normalized.

The implementation of the block can focus either on delay (combinational architecture) or on surface (sequential architecture). The match filter complexity depends on the number L of multiple paths performed during the computation:

- ♦ the larger the L , the higher the number of gates if L branches are implemented;
- ♦ the larger the L , the slower the circuit if only one branch is implemented.

For our implementation, we choose to use only one branch.

2.2.2. Local Emitter Block

This block delivers a sequence image of the interference of the current user k which is part of the residual signal at the input of ICU k . Among several architectures, we

choose to implement a combinational function. In this solution, described in **Figure 5**, a parallel process sends SF_k (spreading factor of user k) chips to the interference cancellation block in order to take into account the multiple paths.

2.2.3. Interference Cancellation Block

This block receives the residual signal coming from the previous user $k-1$ and the image of the interference generated by the local emitter in order to compute the residual signal of the current user k . As described in **Figure 6**, SFmax operators of subtraction are implemented to compute interference cancellation during L clock cycles. Thanks to the combinational structure, this block is not very complex.

2.2.4. Time Analysis of the ICU

Depending on architecture choices, the time required to compute a data symbol can be different. In **Figure 7** we give the processing delays if we choose to implement sequential or combinational operators. In the analysis we consider $L = 6$ paths and a sliding window [5] of size 5. Thanks to the sliding window, we can recover the estimation of previously processed symbols to reduce the latency of the process. Thus the required number of clock cycles is 306 to process the interference cancellation.

2.2.5. Complexity of the Logic Glue in the ICU

To evaluate the complexity, we describe the ICU in VHDL and then synthesize the design with Synopsys Design Analyzer. The target technology is ST microelectronics 90 nm. In **Table 1**, we give common parameters.

Figure 8 shows the area of the different part of the

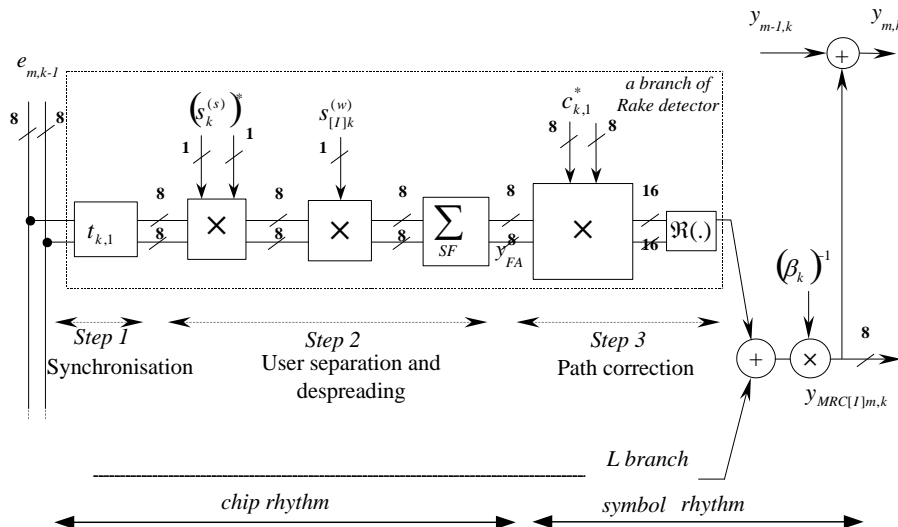


Figure 4. Operators and timing control of the matched filter cell.

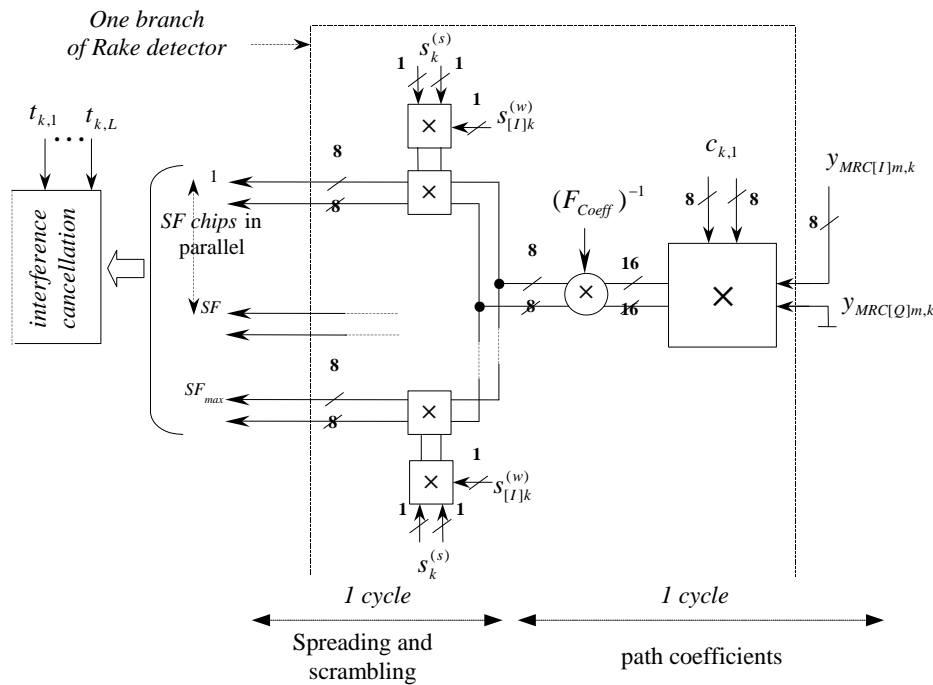


Figure 5. Architecture of the local emitter block.

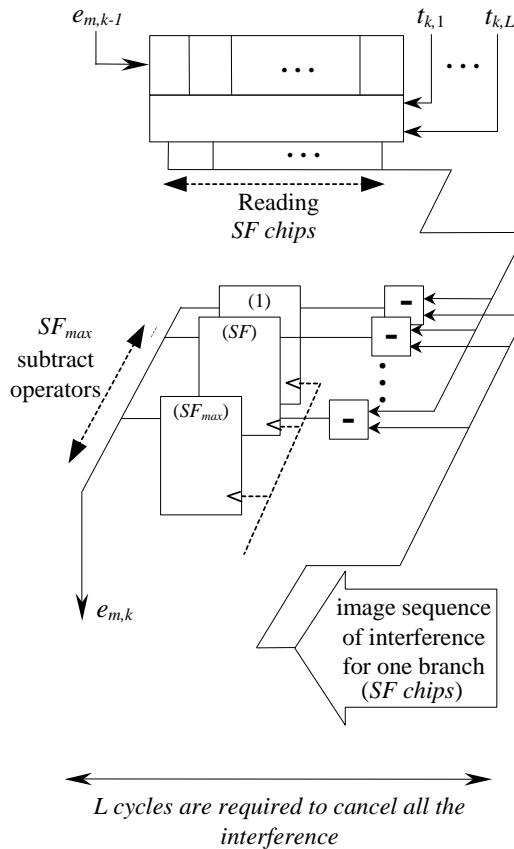


Figure 6. Architecture of the interference cancellation block in Gaussian channel.

Table 1. Value of parameters for the implementation.

Maximum multiple paths	$L_{max} = 6$
Maximum spreading factor	$SF_{max} = 16$
load factor	100%

ICU, taking into account the choices made for implementation. The total area of an ICU is then around 15700 gates.

2.2.6. Memory Requirement for the ICU

Each ICU has to exchange data with the previous and following ICUs. As detailed in chapter 3 of [3], there are eight quantization bits for the input signal. RAM cells are required:

- ◆ MY_{FA} stores the imaginary part and the real part of the symbol after despreading. Its size is then 2×8 bits.
- ◆ MY_{MRC} stores the received symbol correction by channel coefficients (Y_{MRC}). When the SIC detector is followed by a decoder, this memory stores only one symbol, thus its size is 1×8 bits.
- ◆ $Me_{m,k}$ contains the chips of received signals flowing along the ICU. It is updated during the interference cancellation process. Its size is a function of sliding window length (here 5) and spread factor: $5 \times SF \times 2 \times 8$ bits.

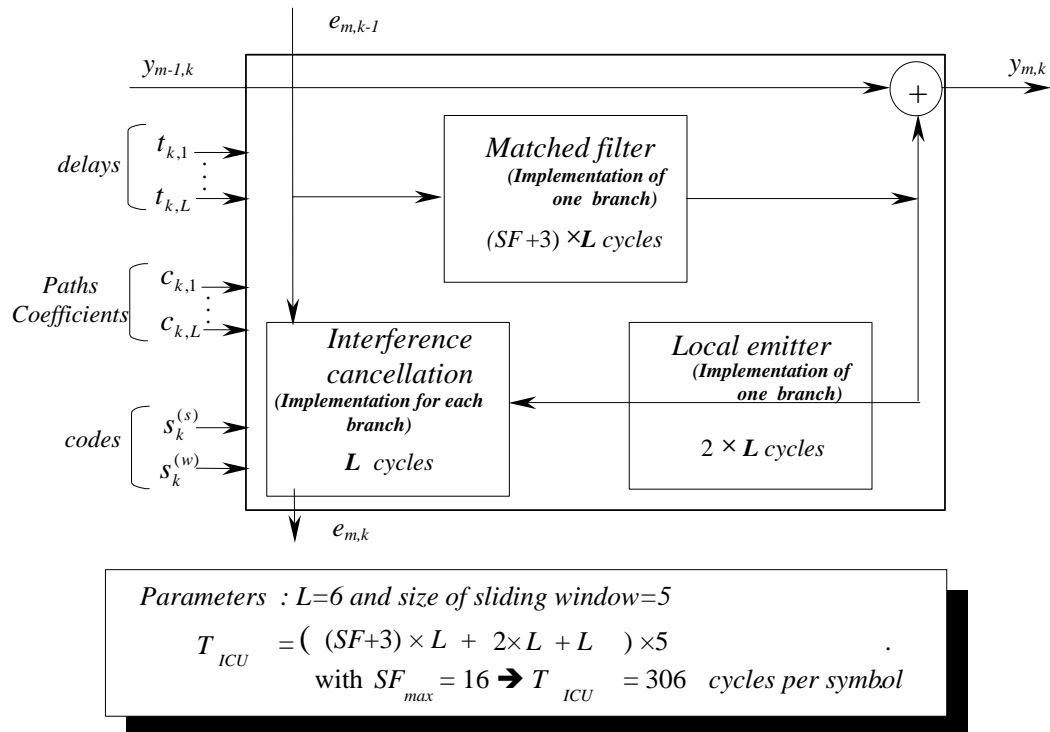


Figure 7. Timing analysis of an ICU.

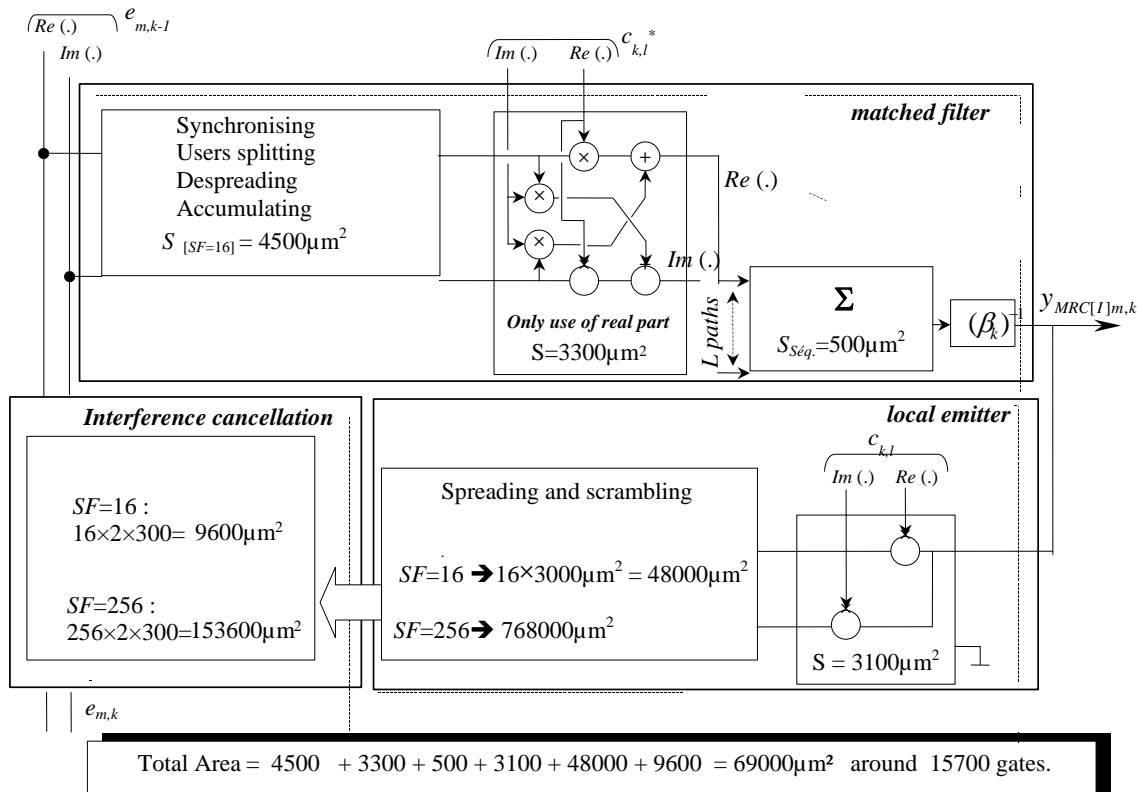


Figure 8. Complexity estimation of an ICU.

- ◆ $MY_{m,k}$ contains the results of the symbol detection for $ICU_{m,k}$. Its size depends on implementation choices. In some configurations, parameter $y_{m,k}$ can be stored through a bus common to the K_{max} stages and an adder. Its size is then $K_{max} \times 8$ bits.

Sizes of the different RAMs are summed up in **Table 2**.

3. Detection and Channel Decoding

The goal here is to analyze the three different associations between detection and channel decoding:

- ◆ In the first one, named CONFIG 1, a bank of channel decoders follows a rake detector;
- ◆ In the second one, named CONFIG 2, the channel decoders follow a 3-stage SIC detector;
- ◆ In the third one, named CONFIG 3, an M -stage joined SIC detector ($M = 2, 3$ or 4) and decoder is implemented. That means that the decoder is inside each interference cancellation unit.

As we can see in **Figure 9**, in terms of BER, CONFIG 3 is really more outstanding than CONFIG 1 or CONFIG 2. Now the question is to see whether the complexity increases dramatically or reasonably. That is the goal of this third part.

3.1. Some Words about the Channel Decoding

Nowadays the benefits of channel encoding are well-known: reducing the power emitted and the error rate. Among different channel coding techniques, we find the turbo codes family invented by Berrou *et al.* [6] in 1992.

On the encoder side, the principle is to code the data with two recursive systematic convolutional (RSC) codes separated by an interleaver.

On the decoder side (**Figure 10(a)**), two soft-in soft-out (SISO) elementary decoders work alternately. Each of them benefits from the other through extrinsic information. The iterative process gives performance close to the Shannon limit. Turbo codes are detailed more in [7]. **Figure 10(b)** shows the architecture of our turbo decoder implementation:

Table 2. Description of the different RAM required in each ICU.

RAM name	size
MY_{FA}	2×8 bits
MY_{MRC}	1×8 bits
$M_Y_{m,k}$	$K_{max} \times 8$ bits
$Me_{m,k}$	$5 \times SF \times 2 \times 8$ bits

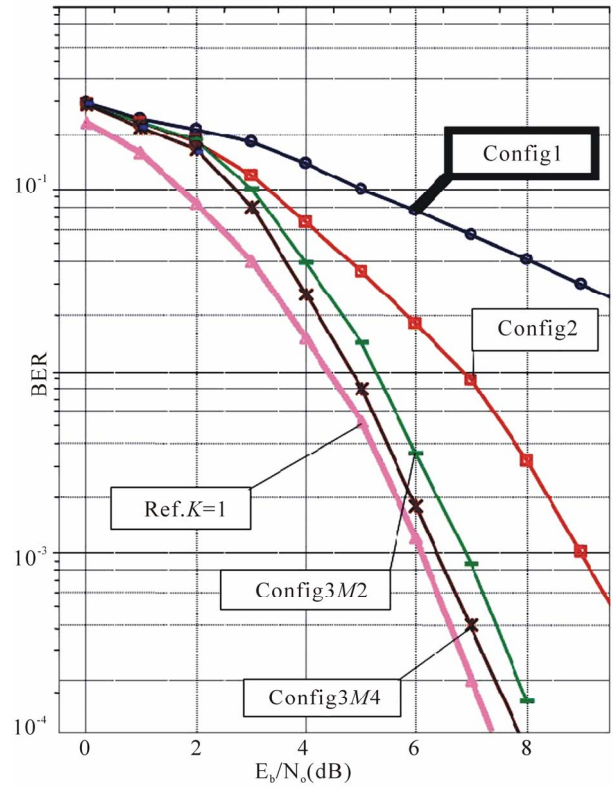


Figure 9. BER vs SNR for different configurations with comparison with single-user performance (spreading Factor = 16, $K = 16$ users, load rate = 100%).

- ◆ input memory to store the word to decode,
- ◆ a single decoder to perform the iterative process,
- ◆ internal memory to exchange the extrinsic information,
- ◆ output memory to store the decoded word.

3.2. Conventional Detection (CONFIG1)

At present, detectors implemented in base stations involve a bank of matched filters (rake detector) followed by a bank of channel decoders. Then a hard decision function determines the received sequence for each user as described in **Figure 11(a)**. To ensure reduced complexity, we choose to implement one branch and then accumulate L times. The architecture is shown in **Figure 11(b)** where it should be noted that the area is around 1900 gates and the processing time is around 110 clock cycles.

3.3. SIC Detector Followed by Turbo Decoder (CONFIG2)

We propose three architectures to implement M -stage of SIC detector for K users followed by a bank of turbo

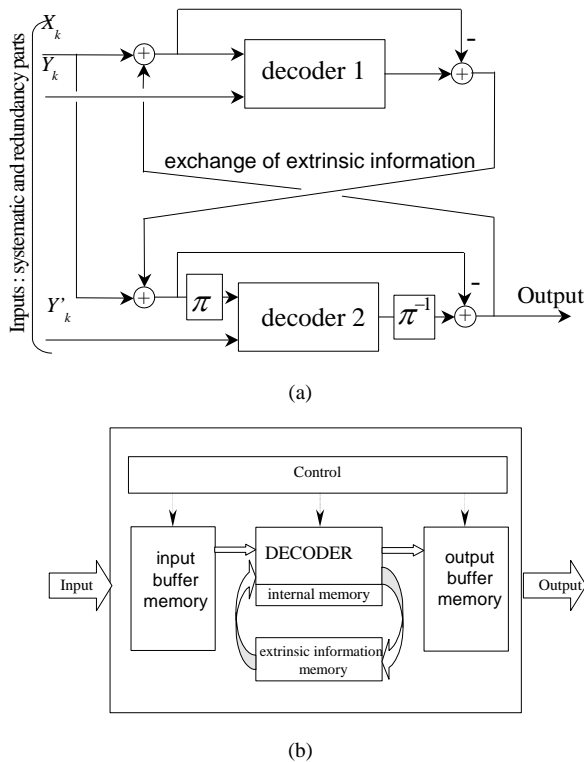


Figure 10. Turbo decoder (a) principle; (b) architecture for implementation.

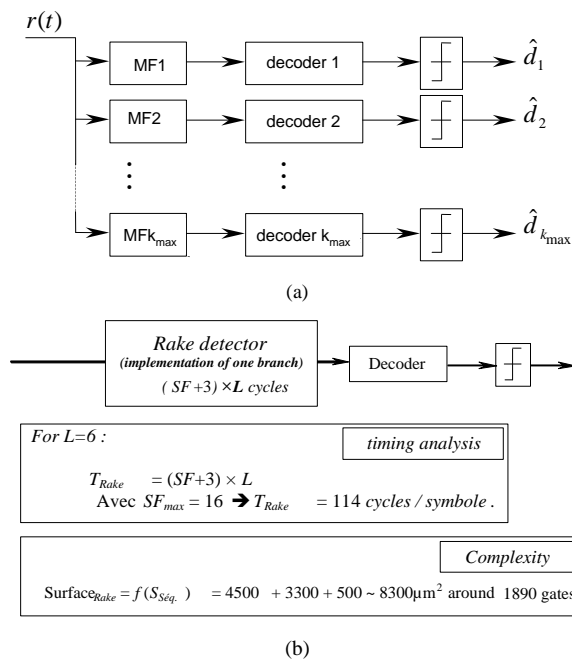


Figure 11. Rake conventional detection: (a) bank of matched filters followed by bank of decoders; (b) lowest complexity, implementation of one branch followed by an L-loop accumulator.

decoders. To process the complete detection function we can choose between:

- Architecture A:** Implementation of one ICU and processing $K \times M$ loops;
- Architecture B:** Implementation of one stage of K ICUs and processing M loops;
- Architecture C:** Implementation of M stages of K ICUs.

Table 3 gives the timing and complexity analysis of the three architectures proposed. We can notice that the latency is the same for all the architectures. The process time can be greatly increased, depending on M or K values, except for Architecture C. But the total area is inversely proportional to the required time to process data. In CONFIG 2, it is essential to implement a memory unit in each decoder as described in Figure 10(b) to allow correct transfer of the received sequence.

3.4. Turbo Decoder inside the Interference Cancellation Unit (CONFIG 3)

As shown in Figure 9, the turbo decoder is placed inside the ICU. This configuration ensures a better estimation before the interference cancellation function. In terms of bit error rate, this structure allows for better results. For the complete implementation, we can also choose between the three architectures A, B and C described in Subsection 3.3. This configuration does not require an external channel decoder, which results in a simpler global architecture.

To process the decoding function, knowledge of the whole frame is required. That is why ICU_{*k*} processes the frame before sending information to ICU_{*k+1*}. This is an important difference from the previous configurations.

The internal structure of the ICU described in Figure 3 has to be modified as shown in Figure 12. What about the impact on area?

- ♦ The area of a turbo decoder is around 450,000 μm^2 .
- ♦ The area of the local emitter and the interference cancellation do not change in comparison with CONFIG 2.
- ♦ For the matched filter it is essential to implement a combinational structure because we have to process a whole frame, so the area is increased by a factor of 10.
- ♦ We have to insert 2 adders.

Thus the computational area of an ICU in CONFIG 3 is around 165,000 μm^2 or 37,600 gates including the turbo decoder. The data given in Table 3 are correct except for the area of turbo decoding, which are now included in S_{ICU} .

Table 3. Comparisons of area and timing for the three architectures considered.

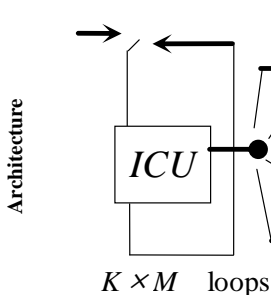
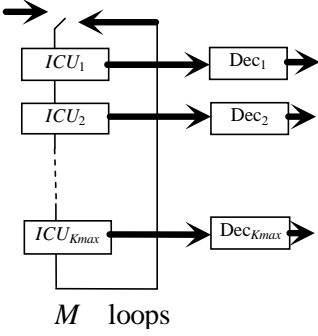
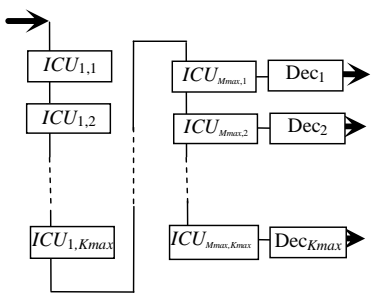
	A	B	C
Architecture	 <p>$K \times M$ loops</p>	 <p>M loops</p>	
Area for SIC detector	S_{ICU}	$S_{ICU} \times K_{\max}$	$S_{ICU} \times K_{\max} \times M_{\max}$
Area for turbo decoding		$K_{\max} \times S_{dec}$	
Process time	$f(K, M, T_{ICU})$	$f(M, T_{ICU})$	$f(T_{ICU})$
Latency of the detector		$f(K \times M \times T_{ICU})$	

Table 4. Comparisons of the different configuration.

	CONFIG 1	CONFIG 2	CONFIG 3
Architecture chosen	a rake, a decoder , K_{\max} loops	a step of K_{\max} ICU, then K_{\max} decoders	a ICU with a decoder inside $K \times M$ loops
Surface	around 0.5 mm ²	around 10 mm ²	around 1.2 mm ²
Processing time (500 MHz)	2.5 ms	2.4 ms	8 ms
SNR required to reach BER = 10 ⁻³ , with load rate 100%	Not reached	8.8 dB	7 dB

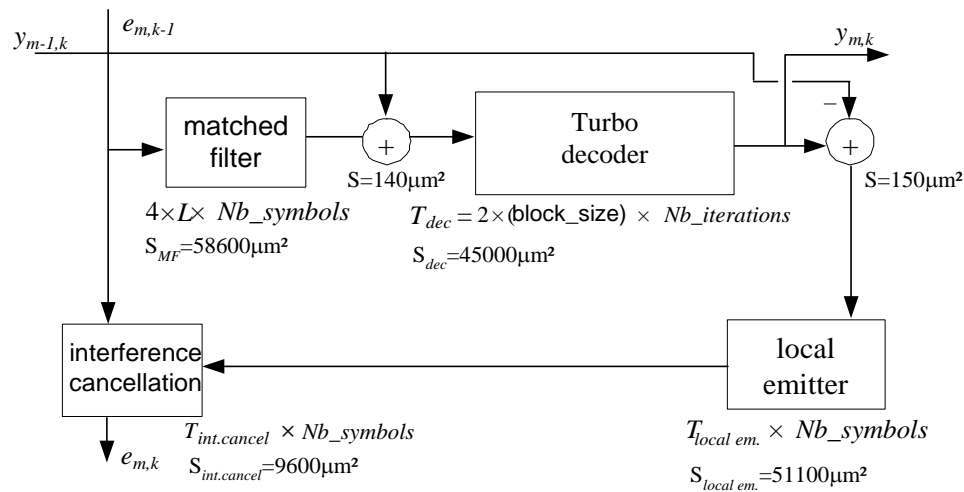


Figure 12. Timing analysis and complexity of the ICU implemented in CONFIG 3.

4. Comparison and Conclusions

In the previous section we describe three different configurations. In this section, we update the data for a real case study in order to see what can be the best choice. The parameters in the UMTS-FDD context are:

- ◆ received rate: 3.84 Mchip/s;
- ◆ frame length: 10 ms;
- ◆ delay from point to point: from 150 ms to 300 ms.

To evaluate the required time to compute one frame for a load rate of 100% ($K_{\max} = 16$ users and $SF = 16$), we apply a frequency of 100 MHz or 500 MHz to the circuit.

In the case of CONFIG 1, which is the solution presently implemented in the base station, we compare the solution of computing the K users successively or simultaneously:

- ◆ The first solution is less complex and it is possible to process one frame in less than 10ms. Indeed, if the clock frequency is 500 MHz, taking into account the results given in **Figure 11(b)** for the rake and in **Figure 12** for the decoder, the delay required to compute one frame is around 5 ms.
- ◆ If we choose to implement a parallel process to compute the K users, the delay is less than 1ms but the complexity increases by almost K .

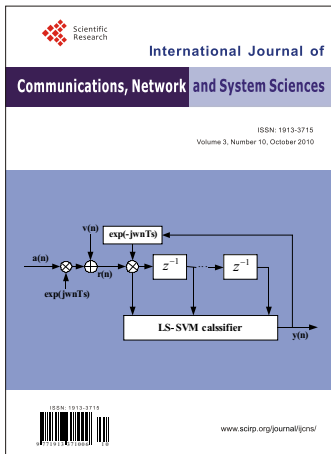
In the case of CONFIG 2, only architectures B and C described in Subsection 3.3 can compute the frame in less than 10 ms. To optimize the surface in CONFIG 2, we choose to implement one step of ICU and K decoders (Architecture B).

In **Table 4**, we sum up the performance, area and computing time for the different architectures retained. **Figure 9** gives the performance in terms of BER. We compare the different solutions by indicating the required SNR to reach a BER of 10^{-3} when $K_{\max} = 16$ users and $SF = 16$ (load rate = 100%).

The solution implemented today cannot reach the performance required in UMTS context unlike CONFIG 2 and CONFIG 3. What is more, the complexity and timing analysis studies show that architecture A can be retained for CONFIG 3, whereas we have to choose architecture B for CONFIG 2. Thus, the final result is that it is possible to implement a turbo decoder inside the interference cancellation unit required for detection. Indeed, the area is only three times higher for a beneficial gain in term of BER.

5. References

- [1] M. Ammar, S. Saoudi and T. Chonavel, "Iterative Successive Interference Cancellation for Multiuser DS-CDMA Detectors in Multipath Channels," *Annales des Télécommunications*, Vol. 57, No. 12, 2002, pp. 105-124.
- [2] S. Saoudi, M. Ammar and T. Chonavel, "Dispositif et Procédé de Décodage de Données AMRC, Système Correspondant," *Institut National de la Propriété Industrielle*, Patent FR 03 14938, 2003.
- [3] M. Touzri, "Étude d'implantation de Détecteurs Multi-utilisateurs CDMA: Application à l'UMTS," PhD: Electronique: Institut TELECOM; TELECOM Bretagne, Université de Bretagne Occidentale: 2007, 2007telb0031. p. 130.
- [4] P. Patel and J. Holtzman, "Analysis of DS-CDMA Successive Interference Cancellation Scheme Using Correlations," *IEEE Globecom'93*, Houston, Vol. 1, 1993, pp. 76-80.
- [5] L. C. A. Hui and K. B. Letaief, "Successive Interference Cancellation for Multiuser Asynchronous DS-CDMA Detectors in Multipath Fading Links," *IEEE Transactions on Communications*, Vol. 46, No. 3, 1998, pp. 384-391.
- [6] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon Limit Error-Correction Coding and Decoding Turbo Codes," *IEEE International Conference on Communications (ICC'93)*, Vol. 2, 1993, pp. 1064-1070.
- [7] C. Berrou, Ed., "Codes and Turbo Codes," Springer, Germany, 2010.



International Journal of Communications, Network and System Sciences (IJCNS)

ISSN 1913-3715 (Print) ISSN 1913-3723 (Online)

<http://www.scirp.org/journal/ijcns/>

IJCNS is an international refereed journal dedicated to the latest advancement of communications and network technologies. The goal of this journal is to keep a record of the state-of-the-art research and promote the research work in these fast moving areas.

Editors-in-Chief

Prof. Huaibei Zhou

Wuhan University, China

Subject Coverage

This journal invites original research and review papers that address the following issues in wireless communications and networks. Topics of interest include, but are not limited to:

Ad Hoc and mesh networks

Coding, detection and modulation

Cognitive Radio

Embedded distributed systems

MIMO and OFDM technologies

Network protocol, QoS and congestion control

Network reliability, security and privacy

Network security

Next generation network architectures

Resource management and quality of service

Signal processing and channel modeling

Simulation and optimization tools

3G and 4G technologies

UWB technologies

Wave propagation and antenna design

We are also interested in:

- Short reports—Discussion corner of the journal :
2-5 page papers where an author can either present an idea with theoretical background but has not yet completed the research needed for a complete paper or preliminary data.
- Book reviews—Comments and critiques.

Notes for Intending Authors

Submitted papers should not have been previously published nor be currently under consideration for publication elsewhere. Paper submission will be handled electronically through the website. All papers are refereed through a peer review process. For more details about the submissions, please access the website.

Website and E-Mail

<http://www.scirp.org/journal/ijcns>

ijcns@scirp.org

TABLE OF CONTENTS

Volume 3 Number 10

October 2010

A Comparative Analysis of Tools for Verification of Security Protocols

N. Dalal, J. Shah, K. Hisaria, D. Jinwala..... 779

Block Layering Approach in TAST Codes

Z. Ahmed, J. P. Cances, V. Meghdadi..... 788

Winning Strategies and Complexity of Nim-Type Computer Game on Plane

B. Verkhovsky..... 793

**A Secure Transfer of Identification Information in Medical Images by
Steganocryptography**

S. H. Jiao, R. Goutte..... 801

**Performance Improvement of Wireless Communications Using Frequency Hopping
Spread Spectrum**

Y. Liu..... 805

**Solutions for 3 Security Problems and its Application in SOA-FCA Service
Components Based SDO**

N. N. Wang, Z. Y. Fang, K. Yan, Y. Tang, X. C. An..... 811

Reliable Multicast with Network Coding in Lossy Wireless Networks

W. Yan, S. Y. Yu, Y. M. Cai..... 816

Using Least Squares Support Vector Machines for Frequency Estimation

X. Y. Teng, X. Y. Zhang, H. Y. Yu..... 821

A Turbo Decoder Included in a Multi-User Detector: A Solution to be Retained

S. Kerouédan, M. Touzri, P. Adde, S. Saoudi..... 826