Institute of Advanced Engineering and Science
International Journal of Information & Network Security (IJINS)
Vol.1, No.1, April 2012, pp. 1 - 8
ISSN: 2089-3299

Improved G-3PAKE Protocol with Formal Verification

H. K. Pathak [1], Manju Sanghi [2]

[1]S.o.S in Mathematics, Pt. Ravishanker Shukla University, Raipur (C.G.) 492010, India
[2]Deptt. of Mathematics, Rungta College of Engg.& Tech., Bhilai (C.G.) 492006, India

Abstract

Password based mechanism is widely used for authentication since it allows people to choose their own passwords without any device to generate or store them. However these protocols should resist all types of password guessing attacks due to the low entropy of passwords. Recently many three party password authenticated key exchange (3PAKE) protocols have been proposed but most of them are vulnerable to various attacks. The present paper proposes an improved G-3PAKE protocol which can resist all the known attacks along with its formal proof.

Keywords: *Password based key exchange protocols, Undetectable on line password guessing attacks, Off-line password guessing attacks, AVISPA.*

1. Introduction

Password authenticated key exchange protocols play a significant role in secure communications in which **two** clients agree a common session key in an authentic manner based on passwords. In 1992, Bellovin and Merrit [1] proposed the first two party password based authenticated key agreement protocol. Considering the problem of key management in large communication environments, Steiner et al. [2] extended the concept to three party and proposed STW-3PEKE protocol. Since then many three party password authenticated key exchange protocols have been proposed. Password based protocols are however, vulnerable to password guessing attacks due to the low entropy of passwords. Ding and Horster [3] divided password guessing attacks into 3 classes. (1) Detectable online password guessing attacks (2) Undetectable online password guessing attacks and (3) Off-line password guessing attacks.

In (1995) Ding and Horster and Sun et al. [4] showed that Steiner et al's 3PAKE protocol is vulnerable to undetectable on line password guessing attacks. In (2000) Lin et al. [5] also showed that STW3PEKE suffers not only undetectable online password guessing attacks but also off-line password guessing attacks. Moreover, they presented a new LSH-3PEKE protocol using server's public key. Lin et al. [6] (2001) presented LSSH-3PEKE protocol resistant to both off-line and undetectable on line password guessing attacks without using server public keys. In (2004) Chang and Chang [7] gave LHL-3PEKE with round efficient version. In the same year Lee et al. [8] presented two enhanced three party encrypted key exchange protocols without using public key techniques. In 2007, Lu and Cao [9] proposed a simple 3 party authenticated key exchange protocol (S-3PAKE). They claimed that their protocol is superior to similar protocols with respect to security and efficiency. In 2008, Guo et al. [10] have shown that S-3PAKE protocol is completely insecure against man-in-the-middle attack and undetectable on-line password guessing attack. They also provided an improved protocol (G-3PAKE) that addresses the identified security problems. Recently, Choi and Yoon [11] have demonstrated that G-3PAKE protocol still falls prey to undetectable on-line password guessing attack by any other client. In the present paper an improved G-3PAKE protocol has been proposed which can resist all the known attacks along with its formal proof. Formal verification of the protocols is necessary to get the user confidence. There are many tools available for verification of the protocols. The analysis and verification of the proposed protocol is done using AVISPA tool [12] (Automated validation of internet security protocols and applications).

The rest of the paper is organized as follows. In Sections 2& 3 Lu and Cao's S-3PAKE protocol and Guo et al.'s G-3PAKE protocol have been reviewed. Section 4 discusses the proposed protocol. Section 5 gives the formal verification of the proposed protocol and the paper is concluded in Section 6.

2. Review of G-3PAKE protocol

This Section briefly reviews simple three party authenticated key exchange (S-3PAKE) protocol proposed by Lu and Cao. The following notations have been used.

Notations:
G, g, p: a finite cyclic group G generated by an element g of prime order p.
M, N: two elements in G.
S : a trusted server.
A, B : two clients.

Received February 4, 2012; Accepted March 10 , 2012

pw_A : the password shared between A and S.
pw_B : the password shared between B and S.
x, y, z : random exponents.
H : one-way hash function.
|| : a bitwise concatenation.

Protocol description:
Assume that two clients A and B wish to agree on a common session key. As they do not hold any shared information in advance they cannot directly authenticate each other and have to resort to the trusted server S. The following are the detailed steps of the protocol as explained in Fig.1.

Step1. A chooses a random number $x \in Zp$ and computes $U = g^x.M^{pwA}$ and sends A || X to B.

Step2. B also chooses a random number $y \in Zp$ and computes $V = g^y.N^{pwB}$ and send A||U||B||V to S.

Step3. Upon receiving A || U|| B || V, S uses pwA and pwB to compute $g^x = U/M^{pwA}$ and $g^y = V/N^{pwB}$. Then S chooses a random number $z \in Zp$ and computes $g^{xz} = (g^x)^z$ and $g^{yz} = (g^y)^z$. Finally S computes $U' = g^{yz}.H(A, S, g^x)^{pwA}$ and $V' = g^{xz}.H(B, S, g^y)^{pwB}$ and sends U′|| V′ to B.

Step4. B on receiving U′|| V′, uses pwB to compute $g^{xz} = V'/ H(B, S, g^y)^{pwB}$ and uses the random number y to compute $g^{xyz} = (g^{xz})^y$ and $\alpha = H(A, B, g^{xyz})$ and forwards α to A.

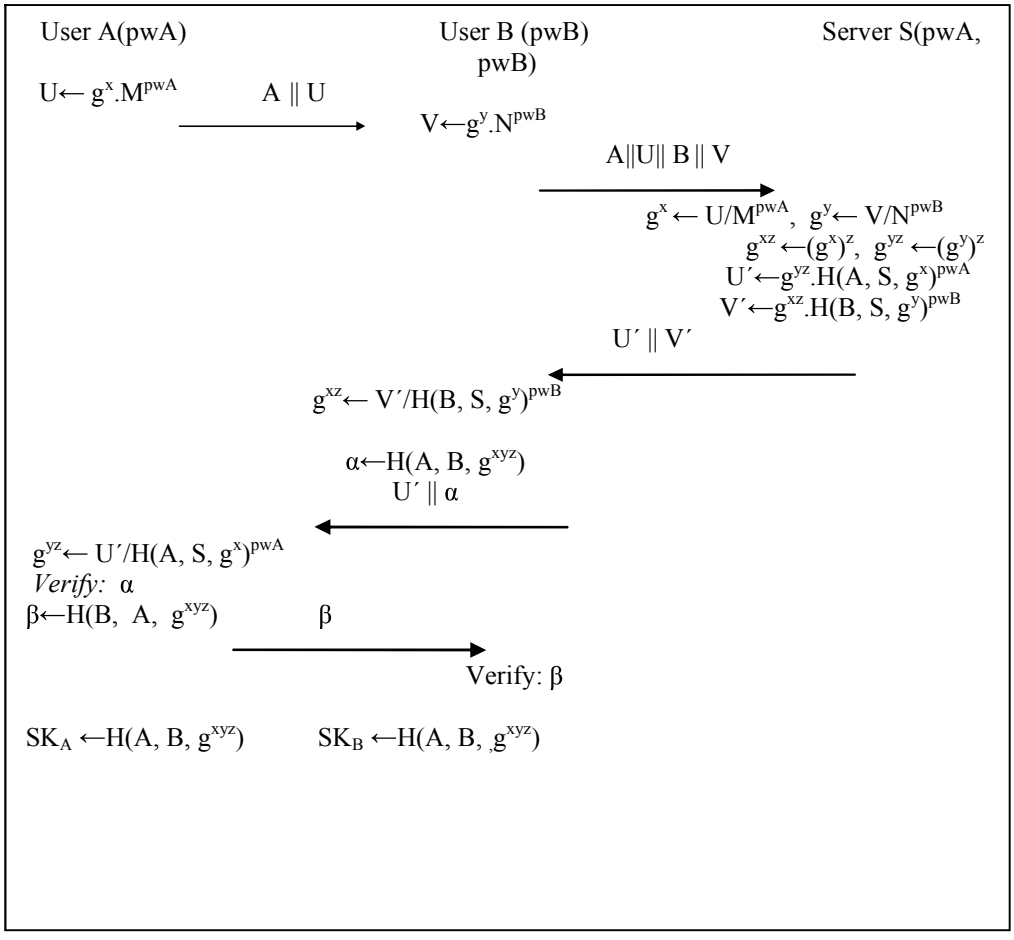

Figure 1 S-3PAKE protocol

Step5. Upon receiving U′|| α, A computes $g^{yz} = U'/H(A, S, g^x)^{pwA}$ and uses x to compute $g^{xyz} = (g^{yz})^x$ and verifies α. If the verification fails, A terminates the protocol, otherwise A computes the session key $SK_A = H (A, B, g^{xyz})$ and sends $\beta = H(B, A, g^{xyz})$ to B.

Step6. B verifies β. If it holds, B computes the session key $SK_B = H(A, B, g^{xyz})$.

3. Review of G-3PAKE protocol

In this section Guo et al's G-3PAKE protocol has been reviewed. Guo et al. found that Lu and Cao's S-3PAKE protocol has some loop holes and attacked by man-in-the-middle attack and on-line password guessing attack. As a counter measure they proposed (G-3PAKE) an improved protocol by which two side users individually implement 2-PAKE protocol to obtain message authentication codes (MAC) prior to creating a shared session key. Clients A and B create $\delta_A = MAC_{kAS}(U)$ and $\delta_B = MAC_{kBS}(V)$ respectively, where k_{AS} and k_{BS} are the MAC keys shared between A and S and B and S respectively. Improved protocol (G-3PAKE) is illustrated in Figure 2.

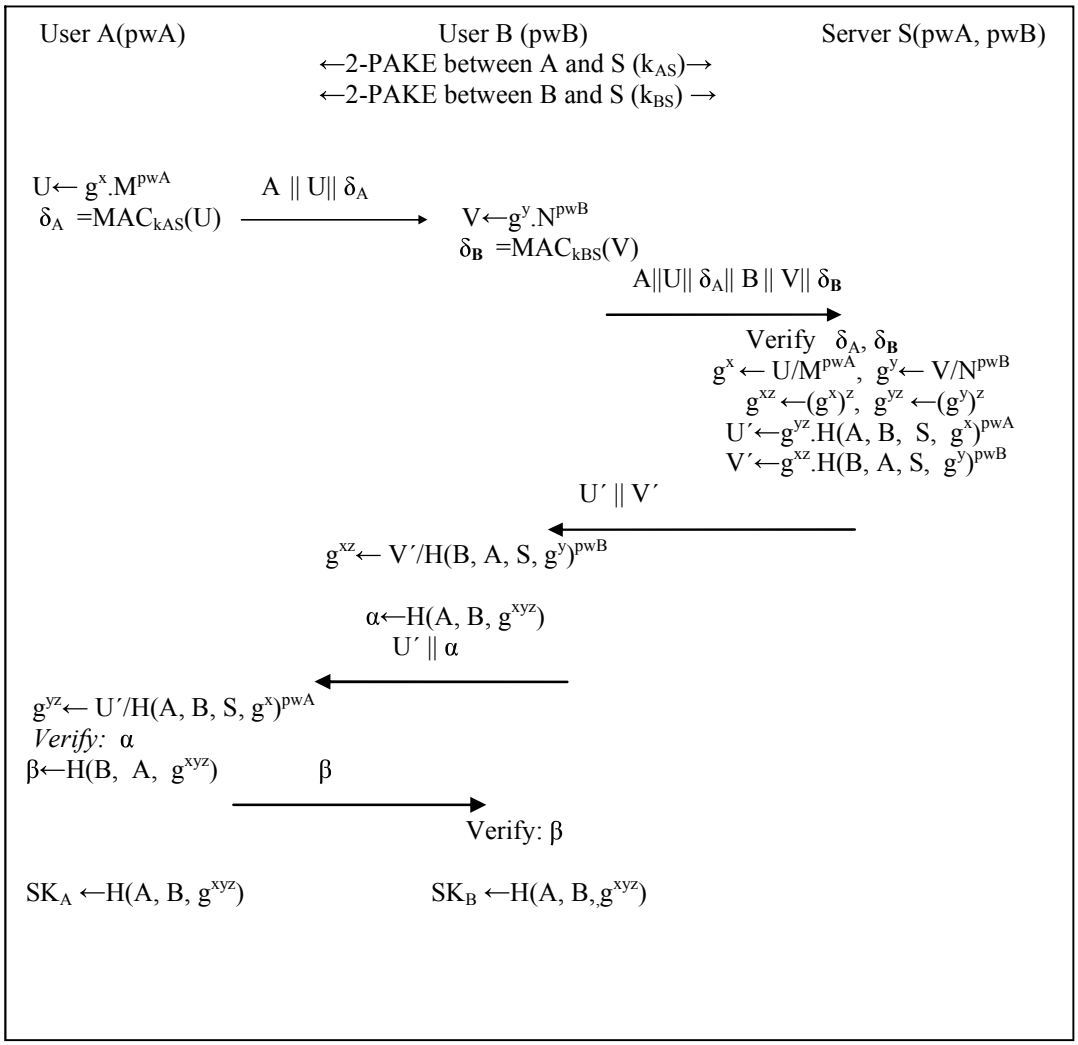

Figure2. G-3PAKE protocol

4. Proposed protocol

Recently, Choi and Yoon have demonstrated that G-3PAKE protocol still falls prey to undetectable on-line password guessing attack by any other client and hence is insecure for practical application. In this Section an improved G-3PAKE protocol using $\oplus$ an exclusive -or operation has been proposed which can resist all the known attacks. The proposed protocol do not require the execution of 2-PAKE protocol.

Step 1: A chooses 2 random numbers x, $p_1 \in$ Zp and computes $U = g^x \oplus M^{pwA}$, g^{p1} and sends $A \parallel U \| g^{p1}$ to B.

Step 2: B also chooses random numbers y, p2 $\in$ Zp and computes $V = g^y \oplus M^{pwB}$, g^{p2} and sends $A \parallel U \| g^{p1} \| B \| V \| g^{p2}$ to S.

Step 3: Upon receiving the messages, S finds $g^x = U \oplus M^{pwA}$, $g^y = V \oplus N^{pwB}$, chooses a random number z $\in$ Zp and computes $g^{xz} = (g^x)^z$, $g^{yz} = (g^y)^z$, $g^{p1z} = (g^{p1})^z$ and $g^{p2z} = (g^{p2})^z$. Finally, S computes $U' = g^{yz}$

$\oplus$ H(pwA, A, B, g^x,g^{p1z}) and V′ = g^{xz} $\oplus$ H(pwB, A, B, g^y,g^{p2z}) and sends (U′||g^z||B), (V′||g^z||A) to B.

Step 4: Upon receiving the message from S, B computes g^{xz} = V′ $\oplus$ H(pwB, A, B, g^y,g^{p2z}) and uses y to compute g^{xyz} = $(g^{xz})^y$. Then B computes α = H(A, B, g^{xyz}) and forwards U′||g^z||B||α to A.

Step 5: A computes g^{yz} = U′ $\oplus$ H(pwA, A, B, g^x,g^{p1z}) and uses x to compute g^{xyz} = $(g^{yz})^x$ and verifies α. If the verification fails, A terminates the protocol, otherwise A computes the session key SK_A= H (A, B, g^{xyz}) and sends β = H(B, A, g^{xyz}) to B.

Step 6: B verifies β. If it holds, B computes the session key SK_B = H(A, B, g^{xyz}).

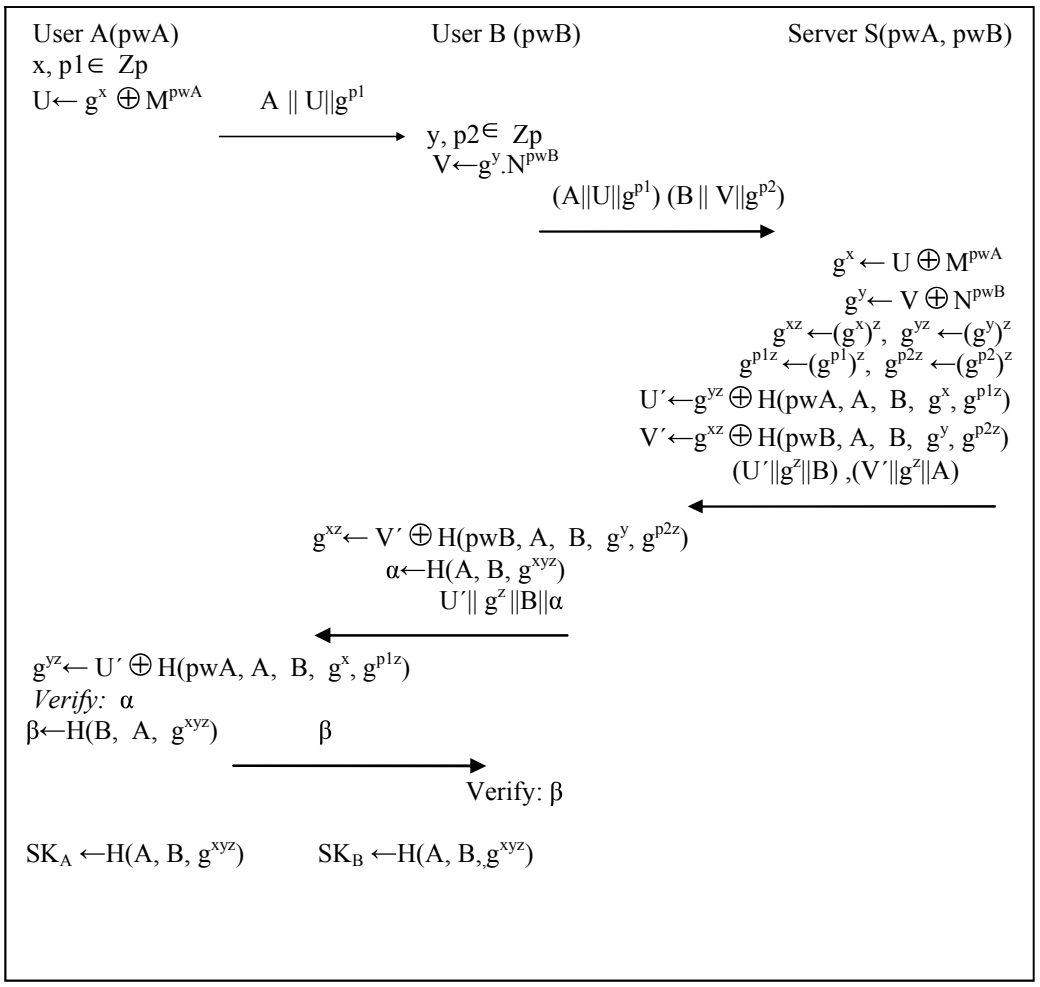

Figure 3. Proposed protocol

5. Formal Verification and Validation of Proposed Protocol

Formal verification of protocols is necessary to get user confidence. A number of tools are available for formal verification of protocols. The security of the proposed protocol is verified using AVISPA tool [12]. Automated validation of internet security protocols and applications (AVISPA) is a push button tool for the automated validation of security protocols. A modular and expressive formal language called HLPSL [13] (High level protocols specification language) is used by AVISPA to specify the security protocol and their properties. HLPSL is a role-based language, meaning that we first specify the sequence of actions of each kind of protocol participant in a module, which is called a basic role. This specification can later be instantiated by one or more agents playing the given role, and we further specify how the resulting participants interact with one another by combining multiple basic roles together into a composed role. HLPSL specification is translated into the Intermediate Format (IF), using hlpsl2if. The IF specification is then processed by model-checkers to analyze if the security goals are violated. There are four different verification back end tools use to analyze the IF specification namely, OFMC (On-the-Fly Model- Checker), CL-AtSe (Constraint-Logic-based Attack Searcher), SATMC (SAT-

based Model-Checker), TA4SP (Tree Automata-based Protocol Analyser). Possible flaws in a protocol can be identified using these back end tools. As, exponential and XOR operations are supported by OFMC and CL-AtSe back ends, OFMC back end tool with AVISPA has been used to analyze the proposed protocol.

Initially three basic roles are defined which are played by Alice (A), Bob (B) and Server (S). Then the composed roles describing the sessions of the protocol and finally the top level role" environment role" are defined.. For analyzing the protocol using AVISPA tool the following notations have been used.

g → G(value of g is stored in G)
x → X
y → Y
z → Z

The HLPSL specification for the proposed protocol is given below.

```
%% PROTOCOL: Improved G-3PAKE protocol
%% ALICE BOB SERVER:
%% Macros:
%% FM1: H(PwA, A, B, exp(G,X),exp(exp(G,P1),Z))
%% FM2: H(PwB, A, B, exp(G,Y),exp(exp(G,P2),Z))
%% Key: exp(exp(GY,Z),X) = exp(exp(GX,Z),Y)
%% GX : exp(G,X)
%% GY: exp(G,Y)
%% U: xor(exp(G,X),exp(M, PwA))
%% V: xor(exp(G,Y),exp(N, PwB))
%% U′: xor(exp(G,YZ),FM1)
%% V′: xor(exp(G,XZ),FM2)
%% α: H(B, A, Key)
%% β: H(A, B, Key)
%% Key: exp(exp(GX,Z),Y) = exp(exp(GY,Z),X)
%% 1. A → B : A||U||g^P1
%% 2. B → S : A||Ug^P1, B||V ||g^P2
%% 3. S → B : (U′||g^z ||B), (V′||g^z ||A)
%% 4. B → A : (U′||g^z||B||α)
%% 5. A→ B : β
```

%% HLPSL:

```
role alice( A, B, S : agent,
        SND, RCV : channel(dy),
        H : hash_func,
        PWA : symmetric_key,
        M, G :text)
Played_ by A
def=
local State : nat,
        X, Z , P1 : text,
        GY, Key :message
        const sec_m_Key : protocol_id
init State := 0
transition
1. State = 0  ∧ RCV(start)= | >
   State':= 1 ∧ X' := new()
              ∧ P1' := new()
              ∧  SND(xor(exp(G,X'), exp(M,PWA)). exp(G,P1'))
2. State = 1  ∧ RCV(xor(exp(GY',Z'), H(PWA.A.B.exp(G,X).exp(exp(G,P1),Z')))
                  exp(G,Z').H(A.B.Key'))= | >
   State':= 2 ∧ Key' := exp(exp(GY',Z'), X)
              ∧ SND(H(A.B. Key'))
              ∧ witness(A,B,key1,Key')
              ∧ request(A,B,key,Key)
              ∧ secret(Key,sec_m_Key,A,B)
end role
```

```
role bob( A, B, S : agent,
          SND,RCV : channel(dy),
          H : hash_func,
          PWA,PWB : symmetric_key,
          M, N, G :text)
Played_by B
def =
local State : nat,
            X, Y, Z, P1, P2 : text,
            GX, GY : message,
             FM1, FM2, Key :message
             const sec_v_Key : protocol_id
init State := 0
transition
1.State = 0  ∧  RCV(xor(exp(G,X'),exp(M,PWA)).exp(G,P1'))= | >
  State':= 1 ∧  Y' := new()
               ∧  P2' := new()
               ∧  SND(xor(exp(G,X'), exp(M,PWA)).exp(G,P1').xor(exp(G,Y'),
                  exp(N,PWB)).exp(G,P2'))
2. State = 1   ∧  RCV(xor(exp(GY,Z'),H(PWA.A.B.exp(G,X').exp(exp(G,P1'),Z'))).exp(G,Z'),
                  xor(exp(GX',Z'),H(PWB.B.A.exp(G,Y).exp(exp(G,P2),Z'))).exp(G,Z'))= | >
   State':= 2  ∧  FM1' := H(PWA.A.B.exp(G,X').exp(exp(G,P1'),Z'))
               ∧  FM2' := H(PWB.B.A.exp(G,Y).exp(exp(G,P2),Z'))
               ∧  SND(xor(exp( GY,Z'),FM1').exp(G,Z').H(A.B.exp(exp(GX',Z'),Y)))
3. State = 2   ∧  RCV(H(A.B.exp(exp(GX',Z'),Y)))= | >
   State':= 3  ∧  Key' := exp(exp(GX',Z'),Y)
               ∧  request(B,A,key1,Key)
               ∧  secret(Key,sec_v_Key,B,A)
               ∧  witness(B, A, key, Key')
end role
```

```
role server ( A,B,S : agent,
              SND,RCV : channel(dy),
              H : hash_func,
              PWA,PWB : symmetric_key
              M, N,G : text)
Played_by S
def =
local State : nat,
            X,Y,Z :text,
            P1,P2 : text,
            GX,GY : message,
            FM1,FM2 : message
init State := 0
transition
1. State = 0  ∧  RCV(xor(exp(G,X'),exp(M,PWA).exp(G,P1')).xor(exp(G,Y'),
                 exp(N,PWB).exp(G,P2')))= | >
   State':= 1  ∧  Z' := new()
               ∧  GY' := new()
               ∧  GX' := new()
               ∧  FM1' := H(PWA.A.B.exp(G,X').exp(exp(G,P1'),Z'))
               ∧  FM2' := H(PWB.B.A.exp(G,Y').exp(exp(G,P2'),Z'))
               ∧  SND(xor(exp(GY',Z'),FM1').exp(G,Z'),xor(exp(GX',Z'),FM2').exp(G,Z'))
end role
```

```
role session( A,B,S : agent,
              H : hash_func,
              PWA,PWB : symmetric_key,
             M, N, G : text)
def =
local SND,RCV : channel (dy)
composition
```

```
alice(A,B,S,SND,RCV,H,PWA,M,G)
 ∧ bob(A,B,S,SND,RCV,H,PWA,PWB,M,N,G)
 ∧ server(A,B,S,SND,RCV,H,PWA,PWB,M,N,G)
end role
```

```
role environment()
def =
const a, b, s : agent,
        h : hash_func,
        key, key1 : protocol_id,
        pwa, pwb, pwi :symmetric_key,
        g, m, n : text
intruder_knowledge = a, b, s, g, h, pwi
composition
session(b,a,s,h,pwa,pwb,m,n,g)
 ∧ session(i,b,s,h,pwi,pwb,m,n,g)
 ∧ session(a,i,s,h,pwa,pwi,m,n,g)
end role
```

```
goal
authentication_on key
authentication_on key1
secrecy_of sec_m_Key, sec_v_Key
end goal
environment()
```

Running the AVISPA tool on the proposed protocol returns the following output.

```
% OFMC
% Version of 2006/02/13
SUMMARY
    SAFE
DETAILS
    BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
    C:\progra_1\testsuite\results\improved G-3PAKE.if
GOAL
   as_specified
BACKEND
   OFMC
COMMENTS STATISTICS
    parseTime: 0.00s
    searchTime: 0.15s
    visitedNodes: 14 nodes
    depth: 3 plies
```

References

[1] Y.Ding, P. Horster, Undetectable on-line password guessing attacks, ACM Operating Systems Review 29 (4) (1995) 77-86.
[2] SM. Bellovin, M. Merritt, Encrypted key exchange: password-based protocols secure against dictionary attacks, In: Proceedings of the 1992 IEEE symposium on research in security and privacy (1992) 72-84.
[3] M. Steiner, G. Tsudik, M. Waidner, Refinement and extension of encrypted key exchange, ACM Operating Systems Review 29 (3) (1995) 22-30.
[4] H.M. Sun, B.C. Chen, T. Hwang, Secure key agreement protocols for three-party against guessing attacks, The Journal of Systems and Softwar e 75 (12) (2005) 63-68.
[5] CL Lin, HM Sun, T. Hwang, Three party-encrypted key exchange: attacks and a solution, ACM Operating Systems Review 34 (4) (2000) 12-20.
[6] C.L. Lin, H.M. Sun, M. Steiner, T. Hwang, Three-party encrypted key exchange without Server public-keys, IEEE Communications Letters 5 (12) (2001) 97-499.
[7] C.C Chang, YF. Chang, A novel three-party encrypted key exchange protocol, Computer Standards and Interfaces 26 (5) (2004) 471-476.

[8] TF.Lee,T. Hwang, CL. Lin, Enhanced three-party encrypted key exchange without server public keys, Computers & Security 23 (7) (2004) 571-577.
[9] R.Lu, Z.Cao, Simple three-party key exchange protocol, Computers Security 26 (1)(2007) 94–97.
[10] Guo Hua, Li Zhoujun, Mu Yi, Zhang Xiyong, Cryptanalysis of simple three party key Exchange protocol, computers security, 27 (2008) 16–21.
[11] Sung-Bae Choi, Eun-Jun Yoon, Cryptanalysis of Guo et al.'s three-party password-based authenticated key exchange (G-3PAKE) protocol, Procedia Engineering, 24 (2011) 187 – 191.
[12] Avispa - a tool for Automated Validation of Internet Security Protocols. http:\\ www.avispa project.org.
[13] D6.2: Specification of the Problems in the High-Level Specification Language. http:\\www.avispa-project.org.

Bibliography of authors

Dr. H. K. Pathak received Post Graduate degree in Mathematics from Pt. Ravishanker Shukla University, Raipur. He was awarded Ph.D in 1988 by the same University. He has published more than 185 research papers in various international journals in the field of non linear analysis-Approximation and expansion, Calculus of variations and optimal controls Optimization, Field theory and polynomials, Fourier analysis, General topology, Integral equations, Number theory, Operations research, Mathematical programming, Operator theory, Sequences, Series, summability, cryptography. At present he is Professor and Head in S.o.S in Computer science & IT in Pt. Ravishanker Shukla University.

Mrs. Manju Sanghi received the post graduate degree in Mathematics from Ravishanker Shukla University Raipur in 1996. Since 2001 she has been working as Assistant professor in Rungta college of Engineering & Technology Bhilai. Currently she is pursuing PhD from School of studies in Mathematics Pt. Ravishanker Shukla University Raipur. Her research interests include Cryptography and Network security.