ISSN 2278-3091
Volume 9, No.5, September - October 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse102952020.pdf
https://doi.org/10.30534/ijatcse/2020/102952020

MULTI-AGENT identity combined key Signature authentication PROTOCOL based schnorr signature with provable security under AVISPA

Sanae Hanaoui [1*], Jalal Laassiri[1], Yousra Berguig[1]
[1] Ibn Tofail University, faculty of science Kenitra Morocco, Informatics research laboratory (LRI),
[1*]sanae.hanaoui@uit.ac.ma,yousra.berguig@gmail.com, LAASSIRI@uit.ac.ma

ABSTRACT

In recent years, Mobile agent technology has been significantly used in different domain applications. Yet because of its strong mobility via network caring its code and data, the agent technology faces some security issues as verifying the integrity and authenticity of information carried by the agent or its code while its migration over the network. therefore, we propose as an alternative solution as a protocol for agent authentication security by adopting a new Identity combined distributed key-based schnorr signature inspired from the secure ID-based signature scheme and the schnorr signature, which could be applied to the agent technology as well to any distributed architecture technology thus to validate its authentication as a digital transaction which is an alternative for certificate-based proxy signatures to grant its data integrity, authentication and non-repudiation in digital communications. The scheme security overall based on the hardness assumption of solving the discrete log problem (DLP). We show through security analysis that our protocol is secure against possible attacks. Furthermore, through the formal security analysis using the AVISPA tool, we justify that our protocol is also secure against passive and active attacks.

Key words: Mobile Agent, cryptography, Identity-based cryptosystem, authenticated encryption, secure transaction, AVISPA, schnorr signature, electronic commerce security.

1. INTRODUCTION

Computer science and network, information integrity, and authenticity check are a prime necessity [1][2]. Therefore, it's one of the security requirements we must grantee in the mobile agents. As a result, we considered the agent validated as authentic (unmodified) by any other parties while its migration from a platform to another one to execute a distributed task assigned by a user via validation of the transaction. However, mobile agents are susceptible to several attacks, particularly by malicious hosts due to their ability to execute mobile code in a remote host. There has been massive work to solve the agents' security lacks. One of the reasonable and practical approaches to solve these issues is to provide a software-based mechanism to prevent any kind of vulnerability. Nevertheless, it's very difficult to opt for implementing any kind of secure function in mobile agents since all the code and data of mobile agents are exposed to the remote host[3]. To emphasis the enforcement of its security, we will propose an identity combined key Signature authentication PROTOCOL based schnorr signature to grantee its authenticity. To guarantee authentication, non-repudiation: and integrity of the agent in transit to the recipient host. Our protocol is different from the other proposed identity based schnorr schemes. In our protocol, we adopted a scheme, in which we define a set of designated cosigners in a selected directory selected by the System Authority. Based on the original signer request the system authority selects a N random number of disturbed registered cosigners' identities from that set based on their availability and workload who can contribute in signing with the original signer, then communicate their set of identities to the Private key Generator (PKG), to generate a new combined secret key for the original signer associated with its corresponding public key, To make it hard to forge the agent signature as well to guess the private key of the original singer. In our protocol, the public key of the original signer is embedded into the signature so that any verifier can verify that the right person only has signed the Agent. We show through analysis

and simulation using the AVISPA tool that our protocol is secure against possible attacks. The proposed protocol is conducted to secure mobile agents technology because of the agent autonomy as well its distributed architecture that makes the implementation of the protocol efficient and fast. Yet its applicability is not limited to agents only it can be integrated into any distributed entities or architecture. The rest of this paper is organized as follows. In Section 2, an overview is given about the different preliminaries that our scheme is based on as well as the motivation of using the schnorr signature. Related works in Section 3. The proposed identity combined key-based schnorr scheme in Section 4. Its security analysis is presented in Section 5. Simulation results for formal security analysis in Section 6. The conclusions of the paper and future work are presented in Section 7.

2. PRELIMINARIES

2.1 Identity-based signature

Digital signature grants in digital communications authentication, data integrity also non-repudiation[4][5]. Diffe and Hellman (1976) were the first who described the notion of a digital signature scheme, and they introduced the traditional Public Key Cryptography (PKC). In a multi-user environment, authentication, revocation, storage of public keys leads to a lot of key management problems. To simplify public-key and certificate management in PKC, Shamir (1984) came up with the Identity-based Public Key Cryptography)idea, **by using a user's** unique **''identity'' information (e.g., its email address) as its public key,** while the user's secret keys are generated by a trusted third party known as Private Key Generator (PKG) from the user's identity besides its ''master secret''. In an identity-based signature (IBS) scheme the verification information does not include any certificate or any individual public key for the signer [6].

2.2 Review of the identity-based signature scheme

In general, an identity-based signature scheme consists of 4 algorithms; is a tuple IBS= (Setup; Extract; Sign; Veri). It could be summarized on 3 algorithms where the Setup and the Extract algorithms concluded in one algorithm named the PKG [7][8].

Setup: This algorithm is run by the third-party PKG on input a security parameter $1^K \in N$, and generates the public parameters ***params*** of the scheme and a master secret key ***s***. The PKG publishes ***params*** and keeps the master secret to itself.

Extract: Given an identity ***ID***, the PKG secret key ***s,*** and ***params***, this algorithm generates the private key d_{ID} of ***ID***. The PKG uses this algorithm to generate private keys for all the participants in the scheme and distribute the private keys to their respective owners through a secure channel.

Sign: This is a probabilistic polynomial-time signature issuing algorithm, which takes input public parameters ***params***, message ***m***, signer's identity ***ID*** and his private key d_{ID}, outputs a signature σ on message ***m***.

Veri: This is a deterministic verification algorithm. On input public parameters ***params***, signer's identity ***ID***, message ***m,*** and a candidate signature σ for ***m***, its outputs either **1** if σ is a valid signature on ***m*** for identity ***ID***, or **0** otherwise.

3. MOTIVATION

Compared to traditional methodologies, we can find Schnorr signatures offering manifold benefits, as fellow:

- They have stronger security proof.
- They're considered the simplest form of a digital signature.
- They can be implemented in blindingly quick ways on Intel hardware therefore they are considered Fast & Efficient.

As well in Schnorr, you can have native k-of-k multi-signatures, in which we can get a bunch of keys together and have a single signature that proves that all of them sum. Multi-signing is considered a big advantage for Schnorr, where a group of people can jointly create a signature that is valid for the sum of their keys.[9] Therefore, we opted for the schnorr signature as its suitable for our proposition and it might be very efficient for the case of multi-agent technology for its distributed architecture.

4. RELATED WORK

In the literature, we find various researches about the identity-based signature as will multi-signature schemes either new contributions or improvements of old propositions. As for the most conducted identity-based signature propositions, are based on the Shamir signature scheme which assumes the existence of trusted key generation centers that give all the users in the network a personalized smart card, using the information embedded in the card enables the user to sign and encrypt the messages he sends, as well to decrypt and verify the messages he receives independently, despite the other party identity[6]. This scheme opened a wild area of research: (Qin et al. 2016) [10] has proposed an ID-based digital signature scheme on the elliptic curve cryptosystem which is integrated with the identification scheme by Popescu using a one-way hash function. to make the trade-off of performance and security stand and most beneficial, their scheme was constructed on the elliptic curve cryptosystem. which protects the signer from the

chosen-message attack and also identifies a forged signature. For (Singh and Verma 2012) [11] have worked on a provably secure ID-based. (Kwon 2014) [12] has introduced a strongly unforgeable identity-based (ID-based) signature (IBS) scheme in the standard model whose security is reduced to the hardness of the computational Diffie-Hellman (CDH) problem in bilinear groups. They have used Waters's system parameters and construction to keep the key pair corresponding to each identity unchanged, their scheme is profitable for devices with low storage capacity due to a smaller number of public parameters. [13].

5. SUGGESTED CONTRIBUTION

5.1 Preliminaries

Notation Used.

The general notation of mathematical expressions will be used in the paper are listed in Table I. These notations are important pre-knowledge for the remainder of the paper.

Table 1: Notations used in the proposed scheme.

Symbol	Description
p	large prime number
q	larger prime such that q/ p-1.
G	the cyclic group of the prime order p.
Z_p	the ring of integer modulo p.
g	a generator of G, exists a number $g \in G$ such that $G = \{1, g, g^2, g^3, \ldots, g^{p-1}\}$.
$H(.)$	Secure one-way hash function.
x_i	the private key of a specific signer.
$x_2, \ldots, x_n$	Private keys of other cosigners.
id_i	a message digest of a corresponding identity to a co-signer or the signer himself.
$Y = \{y_1 = g^{x1}, \ldots, y_n = g^{xn}\}$	the multi-set of all public keys.
$X = \{x_1, \ldots, x_n\}$	the multi-set of all private keys.
CoSigner	A set of other helping trusted signers
M	the message that will be signed in our case it will be the agent.

Table 2: Notations used for computational costs in the proposed scheme.

Symbol	**Description**
t_{add}	Taken time by one modular multiplication operation
t_{exp}	Taken time by one modular exponentiation operation
t_h	Taken time to compute one hash value
t_{mul}	Taken time by one modular addition operation

5.2 The Protocol integrated algorithm

In our protocol, we adopted the four algorithms that generally respect an identity-based signature scheme and we will add our fifth algorithm which will be in charge to combine the keys coming from the cosigners to generate a combined private-public key. Our protocol consists of the following five algorithms. The flow of functionalities of these algorithms of the proposed protocol as follows:

Setup: Given security parameters $\tau, \lambda \in Z(\tau > \lambda)$ as input, PKG runs this algorithm to generate system parameters.

Extract: Given a user's identity ID, PKG runs this algorithm to generate an initial private key.

Combine: Given the output of the Extract algorithm, the PKG runs this algorithm to combine the partial initial shares $\{CSigner_i\}_i \in CoSigners$ from cosigners in the set $CoSigners$ to generate $S \leftarrow (Skey, Pkey)$.

Signature Generation: To generate a signature on a message m in our case it would be the agent using the combined key received from the PKG after running the combined algorithm. This algorithm is run by the original signer.

Verification: Given the system parameters and a signature tuple; any verifier can check the validity of the signature using this algorithm.

5.3 Our Id-based combined key Schnorr signature

In this section, we put forward our main protocol construction of Id-based combined key algorithms, under Schnorr signature (see Figure. 1). The signature uses a cyclic group G of prime order p, a generator of a multiplicative subgroup Z_p^* g of G, and a collision-resistant cryptographic hash function H. The full description of the construction is provided as fellow:

Setup

The PKG picks a security parameter k and generates the system's public parameters and its master-key and the other signer keys. It works as follows:

Key Generation

Given two following: security parameters $\tau, \lambda \in Z(\tau > \lambda)$ as input, the PKG do:

1. Generate a random $\lambda - bit$ prime q.
2. Generate a random $\tau - bit$ prime p such that q divides $p - 1$.
3. Pick an element $g \in Z_p^*$ of order q.
4. Pick a random integer as master-key $\alpha \in [1, q]$ and set $P_{pub} = g^{\alpha} \in Z_p^*$.

5. Let H_1 and H_2 be two hash function $H_1: \{0,1\}^* \rightarrow Z_q$ and $H_2: \{0,1\}^* \rightarrow Z_q$.
6. The PKG publishes systems public parameters $\{p, q, g, P_{pub}, H_1, H_2\}$ and keeps the master-key α secret.

Extract

The signers submit their identities to the PKG:

The PKG generates to each co-signer a private key $x_i \in \{0, \ldots, p-1\}$ where

1. Computes the private key for the $cosigner_i$ with its corresponding identity: $x_i = \alpha P_i$ and we donate $P_i = H_1(id_i)$ associated with original singer identity (id_0).

Key Combining

1. Combines original singer private key associated with the cosigner's private key generated by the PKG: $Y \leftarrow g^X$ and $X \leftarrow \prod_{i=0}^{i} x_i \ mod\ p$. And returns the private combined Id private key associated with its public key

Sends to the original signer the couple $S \leftarrow (X, Y) \in \{0, \ldots, p-1\} \times G$

X : $secretkey$, Y: $publickey$

Signing

To sign a message $m \in \{0,1\}^*$ using the private key X the original signer does:

1. Pick a random k $\in Z_p^*$.
2. Compute $r = g^k \in Z_p^*$ set $c = H_2(Y, r, m)$ and $s = Xc + k \in Z_q$.
3. Output the pair $(s, c) \in Z_q^2$.

Verifying

To verify a message/signature pair (m, (s, c)) using the public key Y the verifier does:

1. Compute $v = g^s Y^{-c} \in Z_p$.
2. Accept the signature if $c = H_2(m \| v)$

. Otherwise, reject.

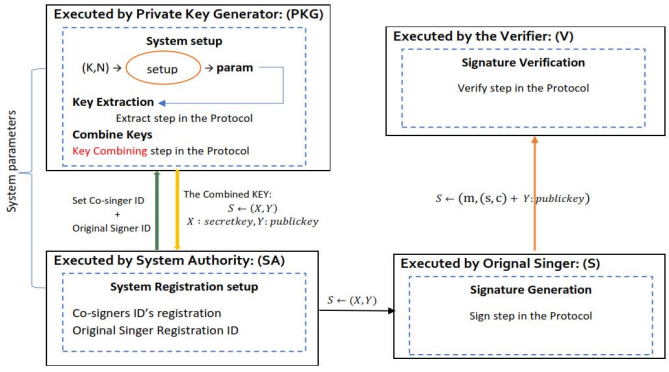

Figure. 1. Principle of the Id-based combined key Schnorr Protocol.

6. ANALYSIS OF THE PROPOSED SCHEME

Firstly we demonstrate the correctness of the scheme. Then we evaluate the computational overhead required for our protocol. Finally, we present that it can tolerate different security attacks.

6.1 Formal security proof for the proposed scheme

It is famously and widely known that the typical version of the El Gamal family signature schemes is provably unforgeable under adaptive chosen-message attack. As well, our scheme is proved unforgeable under an adaptive chosen message attack in Theorem 1.

6.2 Formal security proof for the proposed scheme

It is famously and widely known that the typical version of the El Gamal family signature schemes is provably unforgeable under adaptive chosen-message attack. As well, our scheme is proved unforgeable under an adaptive chosen message attack in Theorem 1.

Theorem 1: Our scheme is secure if and only if the DL problem is believed to be hard in a large finite field.

Proof: To forge a message-signature pair (m; (s; c)), two approaches may be used by an attacker:

Approach 1: Pick Approach 1: Pick k $\in Z^*$ and computes the valuer = $g^k \in Z_p^*$ and c = H_2 (Y, r, m). then s generated and must satisfy $g^s Y^{-c} = g^k$ When c and Y are given, to obtain **s,** he must solve the DL problem.

Approach 2: First, an attacker has to forge s where s = Xc + k and c = H_2 (m||r) hold. It is a DL problem and it must satisfy the one-way hash function because (g, s, m, y) is given but (r, k, α) is unknown and uniformly random, all on the tune of that plausible attack works but with cost at least 2^t hashes, Therefore a conclusion is given that our scheme is secure if and only if the DL problem is believed to be hard in a large finite field.

6.3 Efficiency

In our protocol, the keys generation phase takes on modular exponentiation, as for the signature phase there are two modular exponentiations, one for computing r, and the other for computing s. In the signature verification phase, there are two modular exponentiations for computing r'. So, the computation consumption in our scheme is low. As well for the size of the signature is smaller because of the advantage of the use of schnorr signature which is very useful for our agent technology to migrate with via the network or to handle with distributed servers, hence more the complexity of forging the private key as we adopted the multiplication of secret key based distributed identities generated by a third trusted party (PKG) combined with original signer identity and since it is

protected by a collision-resistant one-way hash function h(Δ). Therefore, the proposed scheme can resist the insider attack for the alteration of the agent while its execution in other platforms.

6.4 Algorithm correctness

For a valid message/signature pair we have:
$v = g^s Y^{-c} = (g^{xc+k})Y^{-c} = (g^{xc}g^k)Y^{-c} = (Y^c g^k)Y^{-c} = g^k$
Therefore $H_2(m \parallel v) = H_2(m \parallel g^k) = c$. It follows that the theorem is proved.

6.5 Computational overhead

For analyzing the computational costs, we use the notations described in **Error! Reference source not found.**. From the **Extract and key combining** phases described in Section 4.3., it is clear that the PKG requires the computational complexity $nt_{mul} + t_{exp} + nt_h$ during these two phases. The original signer signature generation phase described in the same Section 4.3 requires the computational complexity $t_{mul} + t_{exp} + t_h + t_{add}$. As for the verification phase, it requires computational complexity $t_{mul} + 2t_{exp} + t_h$.
Comparing our proposal computational costs which work on the hardness of private key forgery by adopting a new algorithm that takes as input a set of the private keys generated by the system third trusted party associated with an original signer private key, to ID-based Signature Under the Schnorr Signature [14] see (**Error! Reference source not found.**, we could consider the computation consumption in our protocol is low.

Table 3. Notations are used for computational costs in the proposed scheme.

	Schnorr	Our Proposition
Steup Extraction step Cost	$nt_h + t_{exp}$	nt_h
Combine step cost	N	$nt_{mul} + t_{exp}$
Signature cost	$t_{mul} + 2t_{exp} + t_h + t_{add}$	$t_{mul} + t_{exp} + t_h + t_{add}$
Verification cost	$t_{mul} + 2t_{exp} + t_h$	$t_{mul} + 2t_{exp} + t_h$

6.6 Unforgeability

Let an attacker try to forge an original signer signature on any arbitrary message m. Suppose the attacker chooses m the attacker knows the public params as well the public Key Y of the original signer, and try to find s. Therefore, to retrieve X the attacker needs the partial secret keys x_i of the cosigners which were generated by the PKG and their identities were picked randomly from a list of trusted cosigners in the system including the original singer secret key, as well the PKG master-key α which is a computationally infeasible problem due to the difficulty of solving DLP also because of the property of the hash function H(.) for the identities of both cosigners and the original signer. And to determine directly X from $Y \leftarrow g^x$ is a computationally infeasible problem due to the difficulty of solving DLP. Moreover, computing the secret key X from the hash value $H_2(m \parallel v)$ knowing m is computationally infeasible due to the one-way property of the hash function H(.). Hence, the attacker does not have any ability to recompute $H_2(m \parallel v)$ and as a result, the attacker cannot forge the original signer signature. Considering the case where the attacker is one of the users in the set of cosigners chosen by the PKG. the attacker still needs to know the other cosigners secret key as well the secret master key of the PKG and the original signer secret key without forgetting he will have to know the number of cosigners chosen by the system. However, the original signer and the other cosigners send their identities via a secure channel and the System authority verifies the information before communicating it to the PKG. As a result, the attacker does not have any ability to recreate a valid signature and he/she cannot forge the original signer signature. in case all the set of cosigners are not being honest they still need to have both the PKG and the original signer secret key to forge the original signature. As a result, the attackers do not have any ability to recreate a valid signature.

6.7 Verifiability

In the signature generation phase of our scheme, after receiving the tuple (X, Y) from the PKG, the verifier using the public key Y of the original signer, and other information params verifies the condition $c = H_2(m \parallel v)$. Thus, as a result, our proposed scheme is verifiable.

6.8 Secrecy

Note that during the combined key generation phase of our protocol, the PKG generates a private key X and computes the corresponding public key Y for the original signer as it sent to the signer using a secure channel. After that, the original signer selects a random integer $k \in Z_p^*$ and computes the public value $r = g^k \in Z_p^*$. Finally, the original signer computes s and c and sends the message. Now, deriving k, from $r = g^k$, and X from $Y = g^X$ is computationally infeasible due to the difficulty of solving DLP. Thus, the original signer's private key X cannot be derived from any public information by an attacker, and as a result, the secrecy property is also preserved by our scheme.

7. SIMULATION RESULTS

7.1 AVISPA model

Recently, to analyze security systems formally, the automated security validation tool for internet protocols and its applications has become increasingly used especially for cryptographic protocols. AVISPA (Automated Validation of Internet Security Protocols and Applications) is one of the commonly used automated security validation tools, which is a push-button tool that was developed based on the Dolev-Yao intruder model (1983)[15]. Cryptographic protocols analyzed by the AVISPA requires to be specified in a language called HLPSL (High-Level Protocol Specification Language), which is a rule-based language. In HLPSL AVISPA supports four model checkers, called the back-ends. namely OFMC (On-the-fly Model-Checker), CL-AtSe (Constraint-Logic-based Attack Searcher), SATMC (SAT-based Model-Checker) and TA4SP (Tree Automata-based Protocol Analyzer)[16].

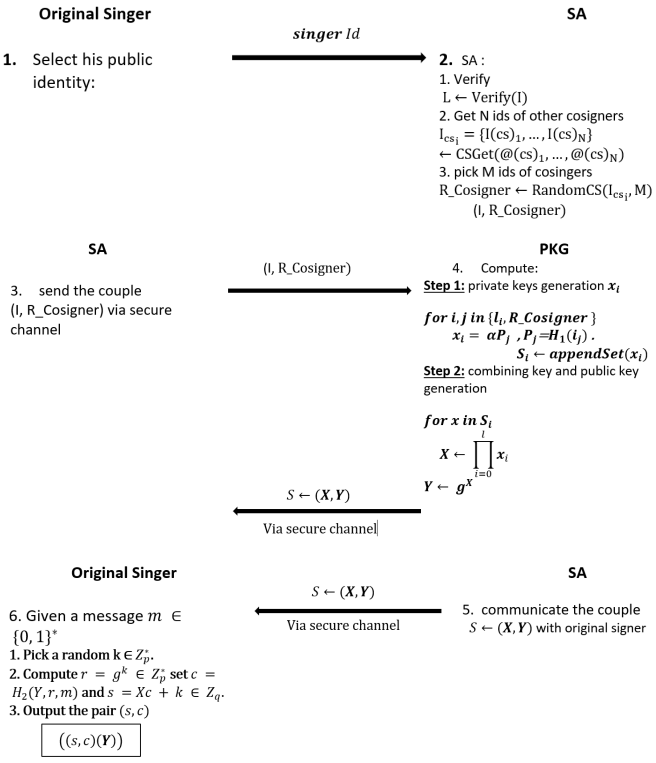

Figure. 2. ID-Combined schnorr key Protocol.

7.2 Specification of the protocol

We have implemented our protocol algorithm (see Figure. 2) under the AVISPA model checkers for formal security analysis to verify whether there is a suspectable attack or not. We have set up three basic roles respectively the original signer A, the system authority S, and the private key generator PKG B in HLPSL related to the key generation phase, signing phase of our identification protocol. Besides these roles, the roles for the session, goal, and environment in HLPSL must be specified for our scheme (see Figure.7).

```
role
orignalsinger(A:agent,B:agent,S:agent,Kat:symmetric_key,SND,RCV:channel(dy)
)
played_by A
def=
        local
                State:nat,
          Na,Nb,X,Y,P,R,M,K,C,D,V,G,Cosigner,Idi:text,
                                H,F,Union : hash_func,
          Kab:symmetric_key
        init
                State := 0
        transition
                1. State=0 ∧ RCV(start) =|>
          State':=1 ∧ Na':=new() ∧ Kab':=new() ∧ SND({B.Kab'}_Kat) ∧
secret(Kab',sec_1,{A,B,S})
                                ∧ Idi':=new()∧ SND({S.Idi'}_Kat)
                2. State=1 ∧ RCV({S.{X'.Y'}_Kab}_Kat) =|>
                        State' := 2 ∧ K':=new()
          ∧ secret(K', sec_2, A)
          ∧ R':=exp(G,K')
          ∧ C':=F(Y,R',M)
          ∧ S':=Union(X',C',K')
          ∧ SND(S',C',Y')

                3. State=2 ∧ RCV({B.Nb'}_Kab) =|> State':=3 ∧
SND({Nb'}_Kab)

          %% A checks that B uses the same key
          %% that he sent at step 1.
          ∧ request(A,B,auth_1,Kab)

          %% A hopes that Nb will permit to authenticate him
          ∧ witness(A,B,auth_2,Nb')

end role
```

Figure. 3. Our protocol role specification for the original signer in HLPSL.

```
role
systemauthority(S:agent,A:agent,B:agent,Kat,Kbt:symmetric_key,SND,RCV:chan
nel(dy))
played_by S
def=
        local
                State:nat,Na,X,Y,Cosigner,Idi:text,Kab:symmetric_key
        init
                State := 0
        transition
                1. State=0 ∧ RCV({B.Kab'}_Kat) =|>

           State':=1 ∧ SND({A.Kab'}_Kbt)
                2. State=1 ∧ RCV({S.Idi'}_Kat) =|>
           State':= 2 ∧ Cosigner':=new() ∧ SND({S.Idi'.Cosigner'}_Kbt) ∧
secret({Idi, Cosigner}, sec_3, {S,B})
                                ∧
SND({B.Idi'.Cosigner'}_Kbt)
                3. State=2 ∧ RCV({B.{X'.Y'}_Kab}_Kbt) =|>
                        State':=3 ∧ SND({A.{X'.Y'}_Kab}_Kbt)
end role
```

Figure. 4. The system authority role specification in HLPSL

```
role pkg(B:agent,A:agent,S:agent,Kbt:symmetric_key,SND,RCV:channel(dy))
played_by B
def=
        local
                State:nat,Na,Nb,D,Idi,X,Y,P,G,Cosigner:text,H,F,Union          :
hash_func,Kab:symmetric_key
        init
                State := 0
        transition
                1. State=0 /\ RCV({A.Kab'}_Kbt) =|>

            State':=1 /\ Nb':= new() /\ SND({B.Nb'}_Kab')
                                     /\ witness(B,A,auth_1,Kab')
                2. State =1 /\ RCV({Idi'.Cosigner'}_Kbt) =|>
                                        State':=2 /\ D':=new()
                                        /\ secret(D', sec_4, B)
                                        /\ P':=H(Idi',Cosigner')
                                        /\ secret(P', sec_5, B)
                                        /\  X':=Union(D',P')
                                        /\  Y':=exp(G,X')
                                        /\  secret({X',Y'},sec_6,{B,A,S})
                                        /\ SND({S.{A.X'.Y'}_Kab}_Kbt)

      3. State=2 /\ RCV({Nb}_Kab) =|> State':=3

          %% B checks that he receives the same nonce
          %% that he sent at step 1.

          /\ request(B,A,auth_2,Nb)
end role
```

Figure. 5. PKG Role specification in HLPSL

```
role session(A:agent,B:agent,S:agent,Kat,Kbt:symmetric_key)
def=
        local
                SND3,RCV3,SND2,RCV2,SND1,RCV1:channel(dy)
        composition
          orignalsinger(A,B,S,Kat,SND1,RCV1) /\
                pkg(B,A,S,Kbt,SND2,RCV2) /\
          systemauthority(S,A,B,Kat,Kbt,SND3,RCV3)
end role

role environment()
def=
        const
                kat,kbt,kit:symmetric_key,      %% we add a symmetric key: kit
shared between the intruder and S
          alice,bob,trusted:agent,
          sec_1,sec_2,sec_3,sec_4,sec_5,sec_6,auth_1,auth_2:protocol_id
        intruder_knowledge = {alice,bob,kit}   %% ... and we give it to the intruder
        composition
              %% We run the regular session
                session(alice,bob,trusted,kat,kbt)
            %% in parallel with another regular session
            /\ session(alice,bob,trusted,kat,kbt)

              %% and a session between the intruder (with key kit) and bob
            /\ session(i,bob,trusted,kit,kbt)
              %% and a session between alice and the intruder (with key kit)
            /\ session(alice,i,trusted,kat,kit)
end role

goal
        secrecy_of sec_1
        secrecy_of sec_2
        secrecy_of sec_3
        secrecy_of sec_4
        secrecy_of sec_5
        secrecy_of sec_6
    authentication_on auth_1
    authentication_on auth_2
end goal

environment()
```

Figure. 6. Session, goal, and environment role specification in HLPSL.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/idcombinedbaseschnorr.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 103.98s
  visitedNodes: 0 nodes
  depth: 1000000 plies
```

Figure. 7. Results of the formal security analysis of the proposed scheme using OFMC back- end.

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/idcombinedbaseschnorr.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 216 states
  Reachable  : 72 states
  Translation: 0.27 seconds
  Computation: 0.01 seconds
```

Figure. 8. Results of the formal security analysis of the proposed scheme using CL-AtSe back-end.

7.3 Results and Discussion

To validate and examine the security properties of our ID-Combined based schnorr protocol Figure. 3, we implemented it using the HLPSL language in the AVISPA tool, and the role specifications of the original signer (Alice), the pkg (Bob), and the designated system authority are given figure 4. We have simulated our scheme using the Security Protocol ANimator for AVISPA (SPAN). The proposed scheme is analyzed in the OFMC and CL-AtSe back-ends, we assume that the intruder knows all public parameters. From these simulation results, the proposed our ID-Combined based schnorr scheme indeed shows its strong security assurance against both passive and active attacks. The results

of the analysis using OFMC, CL-AtSe of our scheme are shown in figure 6 and 7. . The summary of the simulation results are as follows:

Figure.4 shows the specification in HLPSL language for the role of the initiator, the original signer A. A sends the message I di to system authority SA via a secure channel, who is a registered user in the hosted platform. After receiving the message S ←(X, Y) from The PKG, then system authority SA sends the message to original signer A via a secure channel.

Figure.5 shows the specification in HLPSL language for the role of the system authority S. After receiving the message (Ii) from A it sends the message {li, Cosigner} to the PKG via a secure channel.

Figure.6 presents role specification in HLPSL language for the PKG B. After receiving the message {li, Cosigner} from S it sends the message S ← (X, Y) to the S via a secure channel.

Figure.7 presents the roles for the session, goal, and environment in HLPSL.

8. CONCLUSION AND FUTURE WORK

In this paper, we proposed an efficient ID-Combined key Signature under the schnorr scheme, which satisfies all the security requirements needed (provably secure with the hardness assumption to the difficulty of solving DLP as well due to one-way property of the hash function H (.). Additionally, the formal validation of the proposed ID-SDVPS scheme is performed by using an automated validation tool called AVISPA, and the simulation results show that the scheme is unforgeable against active and passive adversaries. After we tested the authentication protocol on Agent technology application to ensure the security of the agent, we are planning to work on the performance efficiency of the ID-combined key-based schnorr authentication protocol.

REFERENCES

[1] M. Bellare, R. Canetti, and H. Krawczyk, "**Keying hash functions for message authentication,? in BT - Proc. 16th Annu. Int. Cryptology Conf. Adv.,"** pp. 1–15, 1996.

[2] I. T. Plata, E. B. Panganiban, and B. B. Bartolome, **"A security approach for file management system using data encryption standard (DES) algorithm,"** *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 5, pp. 2042–2048, 2019.

[3] R. Steinfeld and P. Hawkes, **"Information Security and Privacy,"** *15th Australas. Conf. ACISP 2010*, no. January 2002, pp. 1–403, 2010.

[4] B. Vinoth Kumar, M. Ramaswami, P. Swathika, and A. Raymon, **"Wireless body area network with enhanced object identification, optimal storage and security for integrated healthcare system,"** *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 3, pp. 847–853, 2019.

[5] M. Thomas and V. S. Chooralil, **"Security and privacy via optimised blockchain,"** *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 3, pp. 415–418, 2019.

[6] D. Provisions and E. Resistance, **"I s r a e l,"** *Earthquake*, vol. 196, pp. 1–45, 1998.

[7] Y. Shi, J. Han, J. Li, G. Xiong, and Q. Zhao, "**Identity-based undetachable digital signature for mobile agents in electronic commerce,"** *Soft Comput.*, vol. 22, no. 20, pp. 6921–6935, Oct. 2018.

[8] D. Galindo and F. D. Garcia, **"A Lightweight Identity Based Signature Scheme,"** no. 7639, 2007.

[9] "Encryption." [Online]. Available: https://asecuritysite.com/encryption/. [Accessed: 12-Mar-2019].

[10] Z. Qin, C. Yuan, Y. Wang, and H. Xiong, **"On the security of two identity-based signature schemes based on pairings,"** *Inf. Process. Lett.*, vol. 116, no. 6, pp. 416–418, Jun. 2016.

[11] H. Singh and G. K. Verma, **"ID-based proxy signature scheme with message recovery,"** *J. Syst. Softw.*, vol. 85, no. 1, pp. 209–214, Jan. 2012.

[12] S. Kwon, **"An identity-based strongly unforgeable signature without random oracles from bilinear pairings,"** *Inf. Sci. (Ny).*, vol. 276, pp. 1–9, Aug. 2014.

[13] E. J. Yoon, Y. S. Choi, and C. Kim, **"New ID-based proxy signature scheme with message recovery,"** in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, vol. 7861 LNCS, no. 1, pp. 945–951.

[14] H. Zhonghua and G. Lina, **"ID-based signature under the schnorr signature,"** *2008 Int. Conf. Wirel. Commun. Netw. Mob. Comput. WiCOM 2008*, pp. 8–10, 2008.

[15] L. Viganò, **"Automated Security Protocol Analysis With the AVISPA Tool,"** *Electron. Notes Theor. Comput. Sci.*, vol. 155, no. 1 SPEC. ISS., pp. 61–86, 2006.

[16] S. K. H. Islam and G. P. Biswas, **"ORIGINAL ARTICLE A provably secure identity-based strong designated verifier proxy signature scheme from bilinear pairings,"** *J. KING SAUD Univ. - Comput. Inf. Sci.*, 2013.