Journal of Naval Science and Engineering
2016, Vol. 12, No.1, pp.33-50
Bilgisayar Mühendisliği/Computer Engineering

HYBRID NON-REPUDIATION PROTOCOL WITH ALL TYPES OF PAIRINGS

Yusuf KAVURUCU [1]
Ömer SEVER [2]

[1]*Ass.Prof.Computer Engineering Department,*
Turkish Naval Academy, Naval Sciences and Engineering Institute, Tuzla, Istanbul
[1]*ykavurucu@dho.edu.tr*

[2] *Cryptography Department, METU, Ankara*
[2]*severomer@gmail.com*

Date of Receive: 19.01.2016 **Date of Acceptance: 31.03.2016**

ABSTRACT

As a solution to fair exchange problem, non-repudiation protocols are being widely used over digital environment. Applications of non-repudiation protocols are spreaded over Electronic Contract Signing, Certified E-mail, electronic payment and e-commerce. In this paper we present a strong fair hybrid non-repudiation protocol which works with all types of pairings. The protocol is modeled with an on-line TTP in the first round and then works optimistic in next rounds. The protocol offers stronger security by integration of Joux tri-partite key exchange and uses certificateless ID based signature and encryption methods. All the cryptographic methods used in the protocol are based on pairing based cryptography which can be implemented on all three types of pairings.

TÜM ÇİFTLER İÇİN HİBRİD İNKAR EDİLEMEZLİK PROTOKOLÜ

ÖZ

Dürüst veri alışverişi problemine çözüm olarak, inkar edilemezlik protokolleri sayısal ortamlarda yaygın olarak kullanılmaktadır. İnkar edilemezlik uygulamaları Elektronik Sözleşme İmzalanması, Sertifakalı E-posta, Elektronik Ödeme ve Elektronik Ticarette yaygınlaşmıştır. Bu çalışmada, tüm çiftler için çalışabilecek hibrid bir inkar edilemezlik protokolü sunulmaktadır. Bu protocol, ilk turda çevrimiçi TTP ile modellenmiş ve müteakip turlarda optimistik çalışacak şekilde geliştirilmiştir. Önerilen protocol, Joux üç taraflı anahtar değişimi ile entegre edilerek daha güvenli bir model sunmakta ve sertifikasız kimlik tabanlı

imza ve şifreleme teknikleri kullanmaktadır. Bu protokolde kullanılan tüm kriptografi metodları kriptografi bilimindeki teknikler üzerine geliştirilmiş ve tüm çift kombinasyonları için kullanılabilmektedir..

Keywords: *Crytography, Non-repudiation, Security, Digiral signature*
Anahtar Kelimeler: *Kriptografi, İnkar edilemezlik, Güvenlik, Sayısal imza*

1. INTRODUCTION

Non-repudiation protocols are used for exchange of information with evidence of non-repudiation. Applications of Non-repudiation protocols are spreaded over Electronic Contract Signing, Certified E-mail, electronic payment and e-commerce.

Although there are many different types of Non-repudiation protocols such as Certified E-mail, Contract signing, fair exchange, differing in their goals; they are related with each other and share the properties Non-repudiation and fairness in common. To show these differences with an example; when non-repudiation protocol is based on message delivery like in Certified E-mail, receiver has to provide NRR in order to get the message and obtain the NRO for that message. But when non-repudiation protocol is based on exchange of evidence of non-repudiation not the message itself like in contract signing, obtaining message content is not important but exchanging signed message/contract fairly is the main goal of the application.

2. GENERAL DESCRIPTION

2.1 Non-Repudiation Protocols

Non-repudiation is defined as a security service by which the entities involved in a communication can not deny having participated, specifically, the sender can not deny having sent a message and the receiver can not deny having received a message [1].

Non-repudiation is primarily depending on asymmetric cryptography specifically to signatures which are accepted as evidences. Regarding how used

in a protocol, evidence of origin supplies Non-repudiation of Origin and evidence of receipt supplies Non-repudiation of Receipt.

Non-Repudiation Protocols can satisfy various properties in different ways like:

- Fairness: Strong, weak, light
- Non-Repudiation: NRO, NRR, NRS, NRD
- State storage: Statefull, stateless
- Timeliness: Synchronous, Asynchronous
- TTP Inclusion: In-line, On-line, Off-line, Probabilistic

These properties and non-repudiation protocols have been studied in [2,3,5,6,17].

Public key cryptography is generally based on certificates binding identities with public keys which are approved by Certificate Authorities. What is different in ID-Based Cryptography is public keys are dependant on user identities and/or identifiers. This difference brings advantages and disadvantages together as discussed in [10]. The advantages of ID-Based Cryptography are mainly achieving different encryption and signature schemes like ID-Based encryption [11], blind [12], short [13], ring [14] and verifiably encrypted [15], [22] signatures which are summarized in [4]. The disadvantage of ID-Based cryptography is if the public key is dependant only on identity of a user, key generator knows the private keys of users when generation. In this work which is an expansion of [9], we used certificateless public key cryptography described in [20].

2.2 Bilinear Pairings

Pairings in elliptic curve cryptography are functions which map a pair of elliptic curve points to an element of the multiplicative group of a finite field. Below is the simple definition of a bilinear pairing , more information on pairings like Weil or Tate pairings, divisors and curve selection can be found in [6] as a summary and in [23] in more details.

Let G_1 and G_2 be additive abelian group of order q and G_3 be multiplicative group of order q, a pairing is a function

$$e : G_1 \times G_2 \rightarrow G_3 \quad (1)$$

e is suitable for cryptographic schemes when it is an efficiently computable bilinear pairing which satisfies the following properties:

a) e is bilinear: For all $P, S \in G_1$ and $Q, T \in G_2$ we have $e(P+S,Q) = e(P,Q)\, e(S,Q)$ and $e(P,Q+T) = e(P,Q)\, e(P,T)$

b) e is non-degenerate: For all $P \in G_1$, with $P \neq 0$ there is some $Q \in G_2$ such that $e(P,Q) \neq 1$ and for all $Q \in G_2$, with $Q \neq 0$ there is some $P \in G_1$ such that $e(P,Q) \neq 1$.

Consecutive properties of bilinearity are:

- $e(P,0) = e(0,Q) = 1$
- $e(-P,Q) = e(P,Q)^{-1} = e(P,-Q)$
- $e([a]P,Q) = e(P,Q)^a = e(P,[a]Q)$ for all $a \in Z$

As an expansion to previous work [9], here we can use all three types of pairings.

3. Protocol Definition

We present an ID-based hybrid non-repudiation protocol using the Joux tri-partite key exchange scheme. Our protocol is hybrid because in the first round of exchange TTP is on-line but in the next rounds with same entities TTP works off-line. TTP in the protocol also acts as PKG. If we had used traditional ID-Based encryption and signature methods, TTP can generate and escrow private keys of all users. But in certificateless scheme of [20] users can generate their own private keys. Also revocating a disclosed or lost private key in pure ID-Based crypto systems is difficult because you have to change the corresponding public key and so the ID of that user depends on. Using schemes of [20] TTP can not escrow keys but can revocate keys easily which is important for our non-repudiation protocol depending on pairings. All the

cryptographic methods used in the protocol are based on pairing based cryptography which can be implemented on all three types of pairings.

3.1 Notation

Description of notation is as follows:

A: Sender
B: Receiver
TTP: Trusted Third Party
M_i: Message labeled i; 1 - 6
Sig_X{M}: Message M signed by agent X's private key by ID-Based Signature Scheme
(M)_k: Message M symmetrically encrypted by key k
{M}_X: Message M encrypted for agent X's public key by ID-Based Encryption Scheme
S_id: Session identifier
EOO: Evidence Of Origin
EOR: Evidence Of Receipt
EOS: Evidence Of Submission of key
EOD: Evidence Of Delivery
h(M): Hash of message M
M_id: Message identifier is equal to h(h(M),S_id)
kek_sid: Key encryption key which is equal to h(e(P,Q)^x.y.z , s_id)

3.2 Protocol Description

The protocol starts with an initialization and registration at the beginning. **Initialization:** TTP generates *setup* phase shown in Section 5 and publishes system parameters G_1,G_2, G_3, e, P, Q, P_pub, Q_pub, H_1, H_2. TTP generates $s \in Z^*q$ where P_pub = [s]P, Q_pub = [s]Q and keeps s secret, TTP also generates its own public key P_pub_TTP, Q_pub_TTP and corresponding private key.
Registration: A user with identity ID registers to the TTP. First TTP sends the partial key to user ID, then user ID computes public key P_pub_ID =

[X_ID]P_pub, Q_pub_ID = [X_ID]Q_pub where [X_ID] $\in Z^*_q$ and sends to TTP over authentic channel. User ID computes his private key as shown in Section \ref{Modified}.

Execution: The sender A with public key Q_pub_A, private key d_A_1 computes [x]P and [x]Q where x $\in Z^*_q$ chosen randomly for Joux tri-partite scheme. The receiver B with public key Q_pub_B, private key d_B_1 computes [y]P and [y]Q where y $\in Z^*_q$ random element. TTP with public key Q_pub_TTP, private key d_TTP_1 computes [z]P and [z]Q where z $\in Z^*_q$ chosen randomly.

For the first time of exchange between the participants A, B and TTP round 1 procedure is executed, for next exchanges with the same participants round 2 procedure is executed.

3.2.1 Online Round

Round 1 is the online mod of the hybrid protocol. Main protocol of Round 1, shown in Figure 1 below is as follows:

Step 1
A→B: M_1 = Sig_A{A,B,TTP,S_id, h(M), [x]P, [x]Q, (A,B,TTP,{M}_B)_k}
A→TTP: M_2 = Sig_A{M_1, k_TTP}
Step 2
B→A: M_3 = Sig_B{A,B,TTP,S_id, h(M),[y]P, [y]Q, (A,B,TTP,{M}_B)_k}
B→TTP: M_4 = M_3
Step 3
TTP→B: M_5 = Sig_TTP{A,B,TTP,S_id, h(M),[z]P, [z]Q, k_{kek_sid}}
TTP→A: M_6 = M_5 + M_4

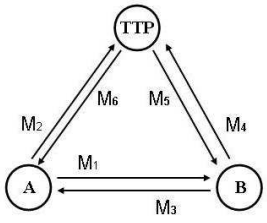

Figure 1. First Round Message Flow

Here the critical point in the protocol is the usage of signed Joux tri-partite key exchange, after the Step 3 of the Round 1, A, B and TTP has [x]P,[y]P,[z]P, [x]Q,[y]Q,[z]Q in common. This means that they can compute $e([y]P,[z]Q)^{x}) = e([x]P,[z]Q)^{y} = e([x]P,[y]Q)^{z} = e(P,Q)^{x.y.z}$.
The steps defined above follow previous one after some checks, as;
In Step 2 receiver B checks the identities, signature of sender A in M_1.
In Step 3 TTP checks:
First, the identities, session identifier and signature of sender A in message M_2.
Secondly, checks if the key k, which was sent in Step 1 by A is working properly. TTP decrypts the encrypted part (A,B,TTP,M_B)_k in message M_1 by the key k and checks the ID's are correct.
Thirdly, checks the identities, session identifier and signature of receiver B in message M_4 which is equal to M_3.
Finally, cross-checks the encrypted part in M_3 is same as the encrypted part in M_1.

Cancellation Sub-protocol

After Step 1 sender A can cancel the protocol by sending TTP a cancellation message. The TTP confirms the Cancellation request if the signature is valid and the request is coming from the sender of the message. The cancellation sub-protocol works as follows;

If any of these checks fail then TTP cancels the protocol. Otherwise TTP continues to Step 3, calculates the kek_sid the key encryption key which is equal to $h(e([x]P,[y]Q)^{z}, s\_id)$, encryptes the key k with kek_sid and sends the messages M_5 and M_6.

Step 1: A→B, TTP: M'_1 = Sig_A{Cancel,M_2}
Step 2: TTP→A, B: M'_2 = Sig_TTP{Cancel-Confirm,S_id,M'_1}

If A sends Cancellation request to only TTP and B sends M_3 and M_4 meanwhile, TTP gets both Cancellation request and M_4. TTP aborts the protocol in this case also. But any Cancellation request from sender after Step 3

is not accepted. Cancellation confirmation is not valid without M'_2. By this way A cannot repudiate M_1 and M_2.

After Step 1 before Step 2 receiver B can also cancel the protocol by sending TTP a Cancellation request. The TTP confirms the Cancellation request if the signature is valid and the request is coming from the receiver of the message. The Cancellation sub-protocol works as follows;

Step 1: B→A, TTP: M'_1 = Sig_B{Cancel,M_1}
Step 2: TTP→A, B: M'_2 = Sig_TTP{Cancel-Confirm,S_id,M'_1}

Dispute Resolution
After Step 2 if the receiver B did not get the key from TTP, recipient B can run Resolve sub-protocol. This is a case if the message M_3 has reached to sender, but message M_4 has not reached to TTP, because of network error or sender A blocks it as an active attack. The Resolve sub-protocol works as follows;

Step 1: B→A, TTP: M'_1 = Sig_B{Resolve,M_1,M_4}
Step 2: TTP→B: M'_2 = M_5
TTP→A: M'_3 = M_6

Before confirmation for resolve request TTP checks the same points as done in main protocol at Step 3.

3.2.2 Off-line Round
Round 2 is the off-line mod of the hybrid protocol.
After Round 1 with online TTP users can pass to Off-line TTP. A, B and TTP has [x]P,[y]P,[z]P.
A,B and TTP have previously computed $e([y]P,[z]Q)^x$), $e([x]P,[z]Q)^y$ and $e([x]P,[y]Q)^z$ respectively.
Now they can use this saved pairing for computing new kek_sid with new sid.
Main protocol of Round 2, shown in picture below is as follows:

Step 1: A→B: M_1 =

Sig_A{A,B,TTP,S_id,M_id,(M_Subj,h(M),h(M,S_id))_kek_sid},
{{A,B,TTP,S_id,{M}_kek_sid}}_TTP}

Step 2: B→A: M_2 = Sig_A{M_1, M_Subj,M_id,h(M),h(M,S_id)}

Step 3: A→B: M_3 = Sig_A{A,B,TTP,S_id,M_id,(M)_kek_sid}

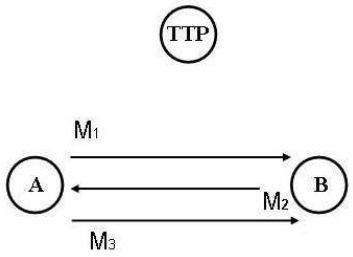

Figure 2. Second Round Message Flow

The steps defined above follow previous one after some checks, as;
In Step 2 receiver B checks the identities, signature of sender A and kek_sid is working properly by decrypting the message identifier encrypted in M_1.
In Step 3 sender A checks the identities, session identifier, signature of sender B and message subject M_Subj has been properly decrypted by B.
If any of these checks fail then TTP cancels the protocol.

Cancellation Sub-protocol
After Step 1 sender A can cancel the protocol by sending TTP a cancellation message. The TTP checks first if the signature is valid and the request is coming from the sender of the message. The TTP confirms the Cancellation request if the status of the session is not Resolved. The cancellation sub-protocol works as follows;

Step 1: A→TTP, B: M'_1 = Sig_A{Cancel,M_1}
Step 2: If (Status(S_id)==Resolved)
Step 2.a Then TTP→A: M'_2 = Sig_TTP{Cancel-Reject,S_id,M_2}
TTP→B: M'_3 = Sig_TTP{Cancel-Reject,(M)_kek_sid}
Step 2.b Else TTP→A, B: M'_2 = Sig_TTP{Cancel-Confirm,S_id,M'_1}

(Status(S_id)=Cancelled)

Dispute Resolution

After Step 2 if receiver B does not get message M_3 or the hash of the message does not match with the hash in the first message, receiver B runs Resolve sub-protocol. The Resolve sub-protocol works as follows;

Step 1: B→TTP, A: M'_1 = Sig_B{Resolve,M_1,M_2}
Step 2: If (Status(S_id)==Cancelled)
Step 2.a: Then TTP→B,A: M'_2 = Sig_TTP{Resolve-Reject,S_id, Sig_TTP{Cancel-Confirm,S_id,M'_1}}
Step 2.b: Else TTP→A,B: M'_2 = Sig_TTP{Resolve-Confirm,S_id,M_id,(M)_kek_sid,M_1,M2}
(Status(S_id)=Resolved)

4. PROTOCOL ANALYSIS

4.1 Fairness and Non-Repudiation

Proposed non-repudiation protocol satisfies fairness in both rounds. By inclusion of online TTP in first round, TTP checks the previous messages, identities, signatures and finally send complementary evidences for both sender and receiver. This achieves strong fairness at the end of the protocol as either each party gets the expexted items (NRO,NRR,Message) or none of them gets a valuable information. If the sender denies, having sent a message M, the receiver can show NRO = M_1 + M_6 and adjudicator rejects denial unless the protocol is cancelled by TTP. In case of cancellation, the sender should show a confirmed cancellation.
If the receiver denies, having received a message M, the sender can show NRR = M_3 + M_5 and adjudicator rejects denial unless the protocol is cancelled by TTP. In case of cancellation, the receiver should show a confirmed cancellation. Since *Cancellation* requests after Step 2 is not accepted, cancellation confirmation and messages M_5 and M_6 can not be present at same time.

For the second round, strong fairness is achieved by help of dispute resolution sub-protocols. Dishonest users can try to get non-repudiation evidences hindering other party to get respective evidence. As a case for dishonest sender; after Step 2, A gets successfully EOR, but can misbehave as sending a cancellation request before a resolve request. In this case since the exchange will be cancelled by TTP and confirmation of cancellation is sent to both parties. Receiever can show to adjudicator that the exchange with S_id is cancelled and EOR in his message M_2 is not valid anymore. As a case for dishonest receiver; after Step 1, B gets successfully EOO, but can misbehave as sending a resolve request to only TTP. In this case the TTP will resolve the issue only if user B sends valid EOR, and this EOR in M_2 will be forwarded to sender A also.

4.2 Timeliness

Asynchronous timeliness is achieved in the proposed protocol by means of cancellation sub-protocols without any time constraint.

4.3 TTP State

TTP works in a statefull manner as has to keeps states of protocol with respect to session identifiers. TTP also keeps securely keys for respective participating parties.

4.4 Efficiency and Comparison

The communication and computation bottleneck of the protocol is TTP for the first round. Since TTP in our protocol acts also as PKG, this situation naturally increased the burden of TTP. But this is not a necessity, PKG and TTP can be different. In that case users should get both PKG parameters and TTP pairing parameters which requires two registration. For the next rounds pairing computations on both sides seems as the reason of computational burden when compared to traditional PKI signatures and encryption.
The proposed protocol has inevitably common characteristics with previously proposed non-repudiation protocols stated in [7], [8] and [2]. It satisfies the

required properties as NRO, NRR, strong fairness and asynchronous timeliness but lacks in efficiency because of pairing computation, online TTP and statefull structure.
The advantage of using a hybrid protocol over other types (pure in-line, on-line or offline) is a kind of optimization between the security and performance. First online round embedded with Joux Tri-partite key exchange scheme enhances the security and next rounds give better performance as being off-line. Our new design does not contribute new capabilities over previous protocols at the moment but it shows that non-repudiation protocols can be built on pairing based cryptography and it is possible to extend this work by using unique properties of identity based cryptography.

4.5 Key escrow and Revocation

Generally key escrow is accepted as a positive capability for authorized third party to gain access to keys needed to decrypt encrypted data. But from view of non-repudiation key escrow property of full Identity-based cryptosystems is regarded as a negative capability. That is why we used identifier based encryption and signature schemes (Certificateless PKC) and TTP can not hold an escrow capability for the private keys of users A and B as stated in [20]. Key revocation can not be handled properly by PKG in a full Identity-based cryptosystem. But by using identifier based encryption and signature schemes (Certificateless PKC) this problem is also eliminated.

4.6 Confidentiality

Confidentiality of the message is ensured in both rounds against eavesdroppers. In the first round message is kept secret even to TTP, but in the second round message can be decrypted by TTP if the cancellation or dispute resolution sub-protocols executed. This property is inserted to improve the efficiency and generally TTP will not be joining the communication. If required this property can be changed as done in the first round.

5. CERTIFICATELESS ID-BASED SIGNATURE AND ENCRYPTION SCHEME

ID-Based signature verification and encryption schemes use publicly known variable such as identity or e-mail of a user to derive public key without any key distribution for public keys. For signing and decrypting user contacts to a Private Key Generator (PKG, CA etc.) to derive the private key which is dependant on the identity and master key of the PKG.

This scheme has some disadvantages stated in [4]

- The PKG can calculate users private keys which is a problem for confidentiality in non-rep protocols
- User has to authenticate himself to PKG
- PKG needs a secure channel to send users private key
- User has to publish PKG's public parameters

To ensure non-repudiation in our protocol we modified and used Riyami and Paterson's certificateless ID-Based encryption and signature schemes described in [20] to eliminate some of these disadvantages.

The original work of Riyami and Paterson's certificateless ID-Based encryption and signature schemes are based on only Type-I pairings. Since Type-I pairings are susceptible to recent quasi-polynomial attacks [26], [27], here we expanded their certificateless PKC to Type-II and Type-III pairings. Here we present our modification to their work.

The *setup* phase is same for both encryption and signature scheme:

Setup: Let G_1 and G_2 be additive group of prime order q and G_3 be multiplicative group of prime order q. Choose an arbitrary generator $P \in G\_1$ and $Q \in G\_2$ and a random secret master key $s \in Z_q^*$.
Set P_pub = [s]P and Q_pub = [s]Q choose cryptographic hash functions H_1 : {0,1}* → G_1 and H_3 : G_3 → {0,1}*. Public and private key pair for user ID is computed as follows:

TTP or PKG computes P_pub = [s]P, Q_pub = [s]Q and [s]H_1(ID), then send to user ID.

User ID computes P_pub_ID = [X_ID][s]P, Q_pub_ID = [X_ID][s]Q and send as public keys then computes d_ID_1= [X_ID][s]H_1(ID) as private key. Our scheme does not need to compute [s]H_2(ID) and d_ID_2= [X_ID][s]H_2(ID) and thus does not need a hash function to G_2 such that H_2 : {0,1}* → G_2 . This gives us the ability to use Type-II pairings.

5.1 Certificateless ID-Based Encryption

We adapted Riyami and Paterson [20] ID-Based Encryption Scheme to all types of pairings.

5.1.1 Encryption

- First choose a random $r \in Z_q^*$
- Message M encrypted by symmetric key k which is ciphered as C = < [r]Q, $k \oplus H\_3(g\_ID)^r$ > where g_ID = e(H_1(ID),Q_pub_ID)

5.1.2 Decryption

C=< U,V> compute k as $k = V \oplus H\_3(e(d\_ID\_1,U))$

5.1.3 Proof of Decryption

Decryption works because;

$V \oplus H\_3(e(d\_ID,U))$

$= V \oplus H\_3(e(d\_ID,[r]Q))$

$= V \oplus H\_3(e([X\_ID][s]H\_1(ID),[r]Q))$

$= V \oplus H\_3(e(H\_1(ID),Q)^{X\_ID \,.\, s \,.\, r}$

$= V \oplus H\_3(e(H\_1(ID),[X\_ID][s]Q)^r$

$= V \oplus H\_3(g\_ID)^r$

5.2 Certificateless ID-Based Signature

We also adapted Riyami and Paterson [20] ID-Based Signature Scheme to all types of pairings.

5.2.1 Signature

For signing message M user ID, chooses an arbitrary $P\_1 \in G^*\_1$ and a random $k \in Z_q^*$

First compute $r = e(P\_1,Q)^k$

$v = H(M,r)$

$u = [v]d\_ID\_1 + [k]P\_1$

The signature is the pair $\langle u,v \rangle \in \langle G\_1, Z_q \rangle$

5.2.2 Verification

When receiving a message M and signature $\langle u,v \rangle \in \langle G\_1, Zq \rangle$ verifier computes

$r = e(u,Q) \cdot e(H\_1(ID), -P\_pub\_ID\_2)^v$

Accept the signature iff $v = H(M,r)$

5.2.3 Proof of Verification

Check if $r = e(P\_1,Q)^k$:

$r = e(u,Q) \cdot e(H\_1(ID), - Q\_pub\_ID)^v$

$= e([v]d\_ID\_1+[k]P\_1,Q) \cdot e(H\_1(ID), - Q\_pub\_ID)^v$

$= e([v][X\_ID][s]H\_1(ID) +[k]P\_1,Q) \cdot e(H\_1(ID), - Q\_pub\_ID)^v$

$= e([v][X\_ID][s]H\_1(ID),Q) \cdot e([k]P\_1,Q) \cdot e(H\_1(ID), - Q\_pub\_ID)^v$

$= e(H\_1(ID),Q)^{v.X\_ID.s} \cdot e([k]P\_1,Q) \cdot e(H\_1(ID), - [X\_ID][s]Q)^v$

$= e(H\_1(ID),Q)^{v.X\_ID.s} \cdot e([k]P\_1,Q) \cdot e(H\_1(ID), Q)^{-X\_ID.s.v}$

$= e([k]P\_1,Q)$

$= e(P\_1,Q)^k$

6. CONCLUSION

We proposed a non-repudiation protocol which has new structure based on pairing based cryptography. The hybrid structure consists of two rounds described in previous sections, first round runs with an online TTP then second and next rounds run with offline TTP. Although online TTP has been regarded as a bottle-neck for security protocols, this is not a big challenge nowadays with usage of high available servers and broad band internet connection. Our main contribution here is the modification of certificateless PKC to all types of pairings. Previous works on non-repudiation protocols have used pairing based cryptography to take advantages of different properties but they also used traditional PKI for encryption and signatures. Differently our protocol is fully based on pairing based cryptography, especially certificateless ID based encryption and signature schemes which prevents some problems of pure ID-based systems.

REFERENCES

[1] NIST Glossary of Key Information Security Terms, FIPS 191.

[2] S.Kremer, O.Markowitch, J.Zhou An Intensive Survey of Non repudiation Protocols, Computer Communications 25 (2002)1606-1621, 2002.

[3] J.L.F.Gomilla, J.A.Onieva, M.Payeras Certified Electronic Mail: Propersties Revisited, Computer & Security (2009) 1-13, 2009.

[4] R.Dutta, P.Barua, P.Sarkar :Pairing Based Cryptography: A Survey, 2004.

[5] C.Calik, O.Sever, H.M.Yildirim, Z.Yuce :A Survey of Certified Electronic Mail Protocols. 4th ISC Turkey, 2010.

[6] S.Akleylek, B.B.Kirlar, O.Sever, Z.Yuce, :Pairing Based Cryptography: A Survey. 3rd ISC Turkey, 2008.

[7] C.Galdi, R.Giordano :Certified email with temporal authentication: An improved optimistic protocol. Proceedings of International Conference on Trust

and Privacy in Digital Business (TrustBus04), LNCS, vol.3184, Springer, Berlin, 2004, pp.181-190.

[8] R.Oppliger, P.Stadlin :A certified mail system (CMS) for the Internet, Comput.Commun.27 2004 1229-1235.

[9] Ö.Sever, E.Akyıldız, Hybrid Non-Repudiation Protocol with Pairing Based Cryptography, 3rd ISDFS, 9-12 May 2015

[10] M.Franklin, G.Price :A comparison between traditional Public Key Infrastructures and Identity-Based Cryptography, 2002.

[11] D.Boneh, M.Franklin :Identity Based Encryption from Weil Pairing. SIAM J.of Computing Vol.32 No.3, 2003, Extended Abstract in Crypto 2001.

[12] A.Boldyreva, :Efficient Threshold Signature, Multisignature and Blind Signature Schemes Based on the Gap-Diffie-Hellman-Group Signature Scheme. PKC 2003,LNCS 2139, pp.31-46 Springer-Verlag 2003.

[13] D.Boneh, B.Lynn, H.Shacham. :Short Signatures from the Weil Pairing. in Proceedings of Asiacrypt 2001.

[14] F.Zhang, K.Kim. :ID-Based Blind Signature and Ring Signature from Pairings. Advances in Cryptology in AsiaCrypt 2002, LNCS Vol.2510, Springer-Verlag, 2002.

[15] F.Zhang, R.Safavi-Naini, W.Susilo. :Efficient Verifiably Encrypted Signature and Partially Blind Signature from Bilinear Pairings. In Proceedings of IndoCrypt 2003, Springer-Verlag, 2003.

[16] F.Hess, :Efficient Identity Based Signature Schemes Based on Pairings, SAC 2002, LNCS 2595 \relax Springer Verlag, 2000.

[17] J.A.Onieva, J.Zhou and J.Lopez :Multi-Party Non-Repudiation: A Survey , ACM Computing Surveys, 2008.

[18] C.Galdi, R.Giordano :Certified E-mail with temporal authentication: An improved optimistic protocol, LNCS Vol.3184, 2004.

[19] A.Joux :One Round Protocol for Tripartite Diffie Hellman, LNCS Vol.1838, 2000.

[20] S.S.Al-Riyami, K.G.Paterson :Certificateless Public Key Cryptography, AsiaCrypt 2003.

[21] C.Bamboriya, S.R.Yadav :A Survey of Different Contract Signing Protocols, Ijetae V.1, I:4, January 2014.

[22] L.Chen, C.Gu :Optimistic Contract Signing Protocol Based on Hybrid Verifiably Encrypted Signature, Advances in Information Sciences and Service Sciences(AISS) V.4, N:12, July 2012.

[23] I.Blake, G.Seroussi, N.Smart :Advances in Elliptic Curves in Cryptography Number 317 in London Mathematical Society Lecture Note Series. Cambridge University Press. ISBN 0-521-60415-X, 2005.

[24] S.D.Galbraith, K.G.Paterson, N.P.Smart :Pairings for Cryptographers Elsevier 2008, Cryptology ePrint Archive, Report 2006/165.

[25] E.R.Verheul :Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. in EuroCrypt 2001, 195-210.

[26] R.Barbulescu, P.Gaudry, A.Joux, E.Tomme. A Quasi-polynomial Algorithm for Discrete Logarithm in Finite Fields of Small Characteristic in EuroCrypt 2014.

[27] R.Granger, T,Kleinjung, J.Zumbragel. Breaking 128 bit Secure Binary Curves