Improved Contract Signing Protocol Based on Certificateless Hybrid Verifiably Encrypted Signature Scheme

Ömer Sever, Ersan Akyıldız
Institute of Applied Mathematics, Middle East Technical University, Ankara, Turkey
e-mail: severomer@gmail.com, ersan@metu.edu.tr

***Abstract*—Contract signing protocols are being widely used over digital environment and treated as an application of non-repudiation protocols. As a kind of non-repudiation protocols, the most important property of contract signing protocols is fairness. In this paper we analyze a recent contract signing protocol based on hybrid verifiably encrypted signature scheme (HVESS) and show a common attack. Further we propose improvement to the protocol, adaptation of certificateless public key cryptography to HVESS (CL-HVESS) and expansion of CL-HVESS to Type-III pairings.**

***Keywords*—Contract Signing protocols, Non-Repudiation protocols, Pairing Based Cryptography, Certificateless Cryptography, verifiably encrypted signature scheme**

I. Introduction

Non-repudiation protocols are used for exchange of information with evidence of non-repudiation. Application of non-repudiation protocols are spreaded over Certfied E-mail, Electronic Contract Signing, e-commerce and electronic payment. E-contract is any kind of contract formed in the course of e-commerce by the interaction of two or more individuals using electronic means, such as e-mail, the interaction of an individual with an electronic agent, such as a computer program, or the interaction of at least two electronic agents that are programmed to recognize the existence of a contract [7]. There are many examples of e-contract platforms over the Internet, some of them serve for general purpose contracts [8], [9] and some of them serve for specific purposes like real estates [10] or like telecommunication suppliers [11].

II. General Description

A. Non-Repudiation and Contract Signing 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 (NRO) and evidence of receipt supplies Non-Repudiation of Receipt(NRR).

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], [21] [25] and [5].

As a kind of non-repudiation protocol, contract signing share similar properties with other protocols. The goal of contract signing protocols is exchange of evidence of non-repudiation not the message itself. Differing from certified e-mail or fair exchange is that obtaining message content is not important but exchanging signed message/contract fairly is the main goal of the contract signing protocol.

B. Pairing Based Cryptography

Public key cryptography (PKC) is generally based on certificates binding identities with public keys which are approved by Certificate Authorities. Differing from classical PKC, in ID-Based Cryptography public keys are dependant on user identities and/or identifiers. This difference brings advantages and disadvantages together as discussed in [14]. The advantages of ID-Based Cryptography are mainly achieving different encryption and signature schemes like ID-Based encryption [15], blind [16], short [17], ring [18] and verifiably encrypted [19], [26] 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 paper we adapted the certificateless public key cryptography described in [24] to the hybrid verifiably encrypted signature scheme [26].

1) 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, divisiors

and curve selection can be found in [6] as a summary and in [27] in more details.

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be additive abelian group of order q and $\mathbb{G}_3$ be multiplicative group of order q, a pairing is a function

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3 \quad (1)$$

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 $P, S \in \mathbb{G}_1$ and $Q, T \in \mathbb{G}_2$ 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 $P \in \mathbb{G}_1$, with $P \neq 0$ there is some $Q \in \mathbb{G}_2$ such that $e(P, Q) \neq 1$ and for all $Q \in \mathbb{G}_2$, with $Q \neq 0$ there is some $P \in \mathbb{G}_1$ such that $e(P, Q) \neq 1$

Consequtive 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 $a \in \mathbb{Z}$

In Section IV we will use this notation to expand HVESS to Type-III pairings.

2) Modified Pairings: In Section III we will use Type I [28] supersingular curves for pairing instantiation in which $\mathbb{G}_1 = \mathbb{G}_2$, to show how we adapted certificateless ID-Based PKC [24] to HVESS. In this type $\mathbb{G}_1$ is a subgroup of $E(\mathbb{F}_q)$. There is a distortion map ψ which maps $\mathbb{G}_1$ into $E(\mathbb{F}_{q^k})$ and the modified pairing $\hat{e}(P, Q) : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_3$ for $P, Q \in \mathbb{G}_1$ is defined by:
$\hat{e}(P, Q) = e(P, \psi(Q))$ as shown in section X in [27].

III. Adaptation of Certificateless Public Key Cryptography to HVESS

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

Chen and Gu have developed and used HVESS [26] which is a pure ID based scheme. To eliminate some of the above mentioned disadvantages, we adapted Riyami and Paterson's [24] certificateless public cryptography scheme to HVESS and call the adapted scheme shortly as CL-HVESS. Most of the parts of our scheme is similar to the original work [26] naturally.

Setup : Let $\mathbb{G}_1$ be additive group of prime order q and $\mathbb{G}_3$ be multiplicative group of prime order q. Choose an arbitrary generator $P \in \mathbb{G}_1$, a random secret PKG master key $s \in \mathbb{Z}_q^*$ and a random secret adjudicator master key $s_T \in \mathbb{Z}_q^*$. Set $P_{pub} = [s]P$ and $P_{adj} = [s_T]P$ choose cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1$ and $H_2 : \{0,1\}^*$ X $\mathbb{G}_1 \to \mathbb{Z}_q^*$. Publish the system parameters $(\mathbb{G}_1, \mathbb{G}_3, q, \hat{e}, P, P_{pub}, P_{adj}, H_1, H_2)$

Extract : Public and private key pair for user ID is computed as follows:

- TTP or PKG computes $P_{pub} = [s]P$ and $[s]H_1(ID)$ as the partial private key then send to user ID.
- User ID computes $P_{pub\_ID} = [X_{ID}][s]P$ and $R_{pub\_ID} = [X_{ID}]P$ as public keys then computes $d_{ID} = [X_{ID}][s]H_1(ID)$ as private key.

Sign : Given a private key d_{ID} and a message m, pick a random $r \in \mathbb{Z}_q^*$, compute $U = [r]P, h = H_2(m, U), V = [r]H_1(ID) + [h]d_{ID}$ and output a signature (U, V).

Verify : Given a signature (U, V), of an identity ID and public keys P_{pub_ID}, R_{pub_ID} first check certificateless public keys as $\hat{e}(R_{pub\_ID}, P_pub) \stackrel{?}{=} \hat{e}(P_{pub\_ID}, P)$ then compute $h = H_2(m, U)$, and accept the signature and return 1 if and only if $\hat{e}(P, V) = \hat{e}(U + [h]P_pub\_ID, H_1(ID)$. The proof of verification for a valid signature (U, V) is as follows;
$\hat{e}(P, V) = \hat{e}(P, [r]H_1(ID) + [h]d_{ID})$
$= \hat{e}(P, [r]H_1(ID) + [h][X_{ID}][s]H_1(ID))$
$= \hat{e}(P, ([r] + [h][X_{ID}][s])H_1(ID))$
$= \hat{e}(([r] + [h][X_{ID}][s])P, H_1(ID))$
$= \hat{e}([r]P + [h][X_{ID}][s]P, H_1(ID))$
$= \hat{e}(U + [h]P_{pub\_ID}, H_1(ID))$

SignVE : Given a private key d_{ID} and a message m, pick randomly $r_1, r_2 \in \mathbb{Z}_q^*$, compute $U_1 = [r_1]P, U_2 = [r_2]P, h = H_2(m, U_1), V = [r_1]H_1(ID) + [h]d_{ID} + [r_2]P_{adj}$, and output a verifiably encrypted signature (U_1, U_2, V)

VerifyVE : Given a verifiably encrypted signature (U_1, U_2, V) of a user ID for a message m, compute $h = H_2(m, U_1)$, accept the signature if and only if $\hat{e}(P, V) = \hat{e}(U_1 + [h]P_pub\_ID, H_1(ID).\hat{e}(U_2, P_{adj})$. The proof of verification for a valid verfiably encrypted signature (U_1, U_2, V) is as follows;
$\hat{e}(P, V) = \hat{e}(P, [r_1]H_1(ID) + [h]d_{ID} + [r_2]P_{adj})$
$= \hat{e}(P, [r_1]H_1(ID) + [h][X_{ID}][s]H_1(ID)).\hat{e}(P, [r_2][s_T]P)$
$= \hat{e}(P, ([r_1] + [h][X_{ID}][s])H_1(ID)).\hat{e}(U_2, P_{adj})$
$= \hat{e}(([r_1] + [h][X_{ID}][s])P, H_1(ID)).\hat{e}(U_2, P_{adj})$
$= \hat{e}([r_1]P + [h][X_{ID}][s]P, H_1(ID)).\hat{e}(U_2, P_{adj})$
$= \hat{e}(U_1 + [h]P_pub\_ID, H_1(ID)).\hat{e}(U_2, P_{adj})$

Adjudication : Given the adjudicator's private key s_T and a valid verifiably encrypted signature (U_1, U_2, V) for a message m, compute $V_1 = V - [s_T]U_2$ and output the original signature (U_1, V_1) Validation requires first verification of verifiably encrypted signature (U_1, U_2, V) and then verification of adjudicated verifiably encrypted signature (U_1, V_1) as an original signature. First part is same procedure as $VerifyVE(U_1, U_2, V)$, for the validation of second part: $V_1 = V - [s_T]U_2 = V - [s_T][r_2]P = V - [r_2]P_{adj}$ $= [r_1]H_1(ID) + [h]d_{ID} + [r_2]P_{adj} - [r_2]P_{adj} = [r_1]H_1(ID) + [h]d_{ID}$ so $\hat{e}(P, V_1) = \hat{e}(P, [r_1]H_1(ID) + [h]d_{ID}) = \hat{e}(U_1 + [h]P_pub\_ID, H_1(ID)$

IV. Expansion of CL-HVESS to Type-III Pairings

In the previous section we have adapted Certificateless PKC to HVESS on Type-I pairings in which $\mathbb{G}_1 = \mathbb{G}_2$. Since Type-I pairings are susceptible to recent quasi-polynomial attacks [30], [31], here we expanded CL-HVESS to Type-III pairings. Type-II pairings are not suitable for CL-HVESS because there is not efficiently computable hash function to $\mathbb{G}_2$.

Setup : Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be additive abelian group of order q and $\mathbb{G}_3$ be multiplicative group of order q. Choose arbitrary generators $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$ a random secret PKG master key $s \in \mathbb{Z}_q^*$ and a random secret adjudicator master key $s_T \in \mathbb{Z}_q^*$. Set $P_{pub} = [s]P$, $Q_{pub} = [s]Q$, $P_{adj} = [s_T]P$ and $Q_{adj} = [s_T]Q$ choose cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \{0,1\}^* \rightarrow \mathbb{G}_2$ and $H_3 : \{0,1\}^*$ X $\mathbb{G}_1 \rightarrow \mathbb{Z}_q^*$. Publish the system parameters $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, q, e, P, Q, P_{pub}, Q_{pub}, P_{adj}, Q_{adj}, H_1, H_2, H_3)$

Extract : 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 $[s]H_1(ID), [s]H_2(ID)$ as the partial private keys then send to user ID.
- User ID computes $P_{pub\_ID} = [X_{ID}][s]P, Q_{pub\_ID} = [X_{ID}][s]Q$ and $R\_P_{pub\_ID} = [X_{ID}]P, R\_Q_{pub\_ID} = [X_{ID}]Q$ as public keys then computes $d\_P_{ID} = [X_{ID}][s]H_1(ID), d\_Q_{ID} = [X_{ID}][s]H_2(ID)$ as private keys.

Sign : Given a private key d_Q_{ID} and a message m, pick a random $r \in \mathbb{Z}_q^*$, compute $U = [r]P, h = H_3(m, U), V = [r]H_2(ID)+[h]d\_Q_{ID}$ and output a signature (U, V).

Verify : Given a signature (U, V), of an identity ID and public keys $P_{pub\_ID}, R\_P_{pub\_ID}$ first check certificateless public keys as $\hat{e}(R\_P_{pub\_ID}, Q_p ub) \stackrel{?}{=} \hat{e}(P_{pub\_ID}, Q)$ then compute $h = H_3(m, U)$, and accept the signature and return 1 if and only if $\hat{e}(P, V) = \hat{e}(U + [h]P_p ub\_ID, H_2(ID)$. The proof of verification for a valid signature (U, V) is as follows;
$\hat{e}(P, V) = \hat{e}(P, [r]H_2(ID) + [h]d\_Q_{ID})$
$= \hat{e}(P, [r]H_2(ID) + [h][X_{ID}][s]H_2(ID))$
$= \hat{e}(P, ([r] + [h][X_{ID}][s])H_2(ID))$
$= \hat{e}(([r] + [h][X_{ID}][s])P, H_2(ID))$
$= \hat{e}([r]P + [h][X_{ID}][s]P, H_2(ID))$
$= \hat{e}(U + [h]P_{pub\_ID}, H_2(ID))$

SignVE : Given a private key d_Q_{ID} and a message m, pick randomly $r_1, r_2 \in \mathbb{Z}_q^*$, compute $U_1 = [r_1]P, U_2 = [r_2]Q, h = H_3(m, U_1), V = [r_1]H_2(ID) + [h]d\_Q_{ID} + [r_2]Q_{adj}$, and output a verifiably encrypted signature (U_1, U_2, V)

VerifyVE : Given a verifiably encrypted signature (U_1, U_2, V) of a user ID for a message m, compute $h = H_3(m, U_1)$, accept the signature if and only if $\hat{e}(P, V) = \hat{e}(U_1 + [h]P_p ub\_ID, H_2(ID).\hat{e}(P_{adj}, U_2)$. The proof of verification for a valid verfiably encrypted signature (U_1, U_2, V) is as follows;
$\hat{e}(P, V) = \hat{e}(P, [r_1]H_2(ID) + [h]d\_Q_{ID} + [r_2]Q_{adj})$
$= \hat{e}(P, [r_1]H_2(ID) + [h][X_{ID}][s]H_2(ID)).\hat{e}(P, [r_2][s_T]Q)$
$= \hat{e}(P, ([r_1] + [h][X_{ID}][s])H_2(ID)).\hat{e}([s_T]P, [r_2]Q)$
$= \hat{e}(([r_1] + [h][X_{ID}][s])P, H_2(ID)).\hat{e}(P_{adj}, U_2)$
$= \hat{e}([r_1]P + [h][X_{ID}][s]P, H_2(ID)).\hat{e}(P_{adj}, U_2)$
$= \hat{e}(U_1 + [h]P_p ub\_ID, H_2(ID)).\hat{e}(P_{adj}, U_2)$

Adjudication : Given the adjudicator's private key s_T and a valid verifiably encrypted signature (U_1, U_2, V) for a message m, compute $V_1 = V - [s_T]U_2$ and output the original signature (U_1, V_1) Validation requires first verification of verifiably encrypted signature (U_1, U_2, V) and then verification of adjudicated verifiably encrypted signature (U_1, V_1) as an original signature. First part is same procedure as $VerifyVE(U_1, U_2, V)$, for the validation of second part: $V_1 = V - [s_T]U_2 = V - [s_T][r_2]Q = V - [r_2]Q_{adj} = [r_1]H_2(ID)+[h]d\_Q_{ID}+[r_2]Q_{adj}-[r_2]Q_{adj} = [r_1]H_2(ID)+[h]d\_Q_{ID}$ so $\hat{e}(P, V_1) = \hat{e}(P, [r_1]H_2(ID) + [h]d\_Q_{ID}) = \hat{e}(U_1 + [h]P_p ub\_ID, H_2(ID)$

V. Attack and Improvement to Fair Contract Signing Protocol

A. Attack to Contract Signing Protocol

Here we show a replay attack to Chen and Gu protocol [26], in which the responder site could get the adjudicated contract but the initiator A, can not get the contract signed by the intended responder B, instead get the contract signed by a colluder C. The attack of the scenerio is figured in Fig.1 and then explained further below.

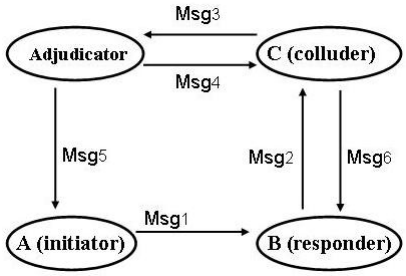

Fig 1. replay attack on protocol

Msg 1 $A \rightarrow B : ID_A, C, SignVE\{d_A, ID_A, C, P_{Adj}\}$

Msg 2 B colludes with C and sends verifiably encrypted signed contract to C
$B \rightarrow C : ID_A, C, SignVE\{d_A, ID_A, C, P_{Adj}\}$

Msg 3 C signs the contract by his private key and request from the adjudicator to resolve the dispute.
$C \rightarrow Adj : (ID_A, C, SignVE\{d_A, ID_A, C, P_{Adj}\})$
and $Sign(d_C, ID_C, C)$

Msg 4 After the adjudicator verifies the signed and verifiably encrypted signed contract, delivers the adjudicated contract to C.
$Adj \rightarrow C : Adjudication(SignVE\{d_A, ID_A, C, P_{Adj}\})$

Msg 5 After the adjudicator verifies the signed and verifiably encrypted signed contract, delivers the signed contract to A.
$Adj \rightarrow A : Sign(d_C, ID_C, C)$

Msg 6 Colluder C returns the adjudicated contract to B.
$C \rightarrow B : Adjudication(SignVE\{d_A, ID_A, C, P_{Adj}\})$

This contract signing protocol is based on HVESS as cryptographic signature scheme, which was proven as secure in [26]. Besides the cryptographic security, information sent in the protocol / signature scheme and control checks are also very important for a security protocol. In this attack we exploited a security flaw of missing information, namely identifier of responder, in the signature scheme. It may be claimed as the contract is suited for A and B but this would not be a formal security check for a protocol.

B. Improvement to Contract Signing Protocol

The improvement to the protocol is very easy as to include the identifier of responder to the signed message and check this before any response. For adding the CL-HVESS, we include the public keys of the sender to the message. The improved protocol is shown below; note that Signed or verifiably signed messages also include the original messages.

Msg 1 $A \rightarrow B : SignVE\{d_A, ID_A, ID_B, C, P_{pub\_A}, R\_P_{pub\_A}, P_{Adj}, Q_{Adj}\}$

Msg 2 $B \rightarrow A : Sign\{d_B, ID_B, ID_A, C, P_{pub\_B}, R\_P_{pub\_B}, P_{Adj}, Q_{Adj}\}$

Msg 3 $A \rightarrow B : Sign\{d_A, ID_A, ID_B, P_{pub\_A}, R\_P_{pub\_A}, Msg2\}$

C. Analysis of Protocol

Although there is not a formal security proof for CL-HVESS, we can make an informal comparison between original protocol and our work. When you use traditional ID-Based encryption and signature methods, as done in the original scheme, TTP can generate and escrow private keys of all users. But in certificateless scheme of [24] 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 [24] TTP can not escrow keys but can revocate keys easily which is important for contract signing protocols depending on pairings. Addition to security analysis we can say the improved protocol is resistant to replay attacks. When we compare our adapted protocols with original version in view of efficiency, there is not so much difference between them. Both Type I and Type III versions of CL-HVESS have same calculations except setup phase which is done for only once. Below is the comparison of efficiency:

- **Sign** same as original; 3 scalar multiplication.
- **Verify** extra two pairings to check certificateless public keys; in total 4 pairings, 1 scalar multiplication.
- **SignVE** same as original; 5 scalar multiplication.
- **VerifyVE** extra two pairings to check certificateless public keys; in total 5 pairings, 1 scalar multiplication.
- **Adjudication** same as original; 1 scalar multiplication.

VI. Conclusion

We proposed adaptation of certificateless public key cryptography to hybrid verifiably encrypted signature scheme [26] which we call CL-HVESS. Adaptation of certificateless PKC prevents some problems of pure ID based schemes especially generation of user private keys by PKG. Then we expanded CL-HVESS to Type-III pairings to mitigate the risks of recent attacks on Type-I pairings. We also presented a replay attack to Chen and Gu protocol [26], in which the responder site could get the adjudicated contract but the initiator A, can not get the contract signed by the intended responder B, instead get the contract signed by a colluder C. Then propose an improvement to the protocol which is resistant to replay attacks and also included the CL-HVESS to the improved protocol. But notice that this attack and CL-HVESS are independent. Formal security proof of CL-HVESS remains as a future work.

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: Properties 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* 4^{th} ISC Turkey, 2010.
[6] S. Akleylek, B.B. Kirlar, O. Sever, Z. Yuce, *Pairing Based Cryptography: A Survey* 3^{rd} ISC Turkey, 2008.
[7] US Legal Definition *http://definitions.uslegal.com/e/e-contract/*
[8] *https://www.e-contract.be/*
[9] *http://www.signable.co.uk/legal*
[10] *https://www.ctmecontracts.com/eContracts/wp/index.htm*
[11] *http://www.telefonica.com/en/about_telefonica/html/suppliers/soluciones/econtracts.shtml*
[12] 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.
[13] R. Oppliger, P. Stadlin *A certified mail system (CMS) for the Internet*, Comput.Commun.27 2004 12291235.
[14] M. Franklin, G. Price *A comparison between traditional Public Key Infrastructures and Identity-Based Cryptography*, 2002.
[15] D. Boneh, M. Franklin *Identity Based Encryption from Weil Pairing.* SIAM J.of Computing Vol.32 No.3, 2003, Extended Abstract in Crypto 2001.
[16] 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.
[17] D. Boneh, B. Lynn, H. Shacham. *Short Signatures from the Weil Pairing.* in Proceedings of Asiacrypt 2001.
[18] 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.
[19] 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.
[20] F. Hess, *Efficient Identity Based Signature Schemes Based on Pairings*, SAC 2002, LNCS 2595 Springer Verlag, 2000.
[21] J.A. Onieva, J. Zhou and J. Lopez *Multi-Party Non-Repudiation: A Survey* ACM Computing Surveys, 2008.

[22] C. Galdi, R. Giordano *Certified E-mail with temporal authentication: An improved optimistic protocol* LNCS Vol.3184, 2004.
[23] A. Joux *One Round Protocol for Tripartite Diffie Hellman* LNCS Vol.1838, 2000.
[24] S.S. Al-Riyami, K.G. Paterson *Certificateless Public Key Cryptography* AsiaCrypt 2003.
[25] C. Bamboriya, S.R. Yadav *A Survey of Different Contract Signing Protocols*, Ijetae V.1, I:4, January 2014.
[26] 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.
[27] 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.
[28] S.D. Galbraith, K.G. Paterson, N.P. Smart *Pairings for Cryptographers* Elsevier 2008, Cryptology ePrint Archive, Report 2006/165.
[29] E.R. Verheul *Evidence that XTR is more secure than supersingular elliptic curve cryptosystems.* in EuroCrypt 2001, 195-210.
[30] R. Barbulescu, P. Gaudry, A. Joux, E. Tomme *A Quasi-polynomial Algorithm for Discrete Logarithm in Finite Fields of Small Characteristic* in EuroCrypt 2014.
[31] R. Granger, T, Kleinjung, J. Zumbragel *Breaking 128 bit Secure Binary Curves*