Verifiably Encrypted Signcryption Scheme Based on Pairings

Ömer Sever

Institute of Applied Mathematics, Middle East Technical University.
Ankara, Turkey e-mail: severomer@gmail.com

Abstract—Signcryption as a cryptographic method combines signing and encryption usually in sing then encrypt order. Verifiably encrypted signatures are used mainly for fair exchange and contract signing protocols. In this paper we propose a new scheme (up to our knowledge the first) that combines signcryption and verifiably encrypted signatures which we call VESigncrypt.

Keywords—Verifiably Encrypted Signcryption, Contract Signing protocols, Non-Repudiation protocols, Pairing Based Cryptography, Verifiably Encrypted Signatures, Signcryption.

1. Introduction

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. Verifiably encrypted signatures are used mainly for fair exchange and contract signing protocols to sustain fairness in cryptograhic manner. Although cofidentiality of the message is not as important as fairness for ordinary contracts, in the case of secret contracts confidentiality will be as important as fairness. Signcryption as a cryptographic method combines signing and encryption usually in sing then encrypt order. In this paper we propose a new scheme (up to our knowledge the first) that combines signcryption and verifiably encrypted signatures which we call VE-Signcrypt.

2. General Description

2.1. Signcryption

Signcryption was first introduced by Zheng [20] and then accrued many different signcryption methods [21]. Signcryption can be constructed in different orders as; sign-then-encrypt, encrypt-then-sign, commit-then-encrypt-and-sign paradigms. Also signcryption can be performed basically for single recepient or for multi-recipients. It is applicable in a wide area where both confidentiality and authenticity is required like e-voting.

2.2. Verifiably Encrypted Signatures

Verifiably encrypted signature was first introduced by Boneh et al [13] as a cryptographic primitive to satisfy mainly fairness in fair exchange, contract signing [16], [9] and certified electronic mail protocols [3]. By using verifiably encrypted signatures in a protocol sender S can send an encrypted signature to a receiver R. The receiver R can check that signature validity but can not get the actual signature without help of an adjudicator. When the receiver requests from the adjudicator with valid reasons to adjudicate, he can recover the actual signature from verifiably encrypted signature.

2.3. Pairing-Based Cryptography

Pairings were first introduced into cryptography to break elliptic curve discrete logarithm problem. Consequently, they are used to construct cryptographic schemes as building stones which we call pairing-based cryptography (PBC). PBC has made many cryptographic mechanisms easier to be implemented and thus attracted many cryptographers attention. Also it provides design of new schemes more effectively and simple. ID-Based cryptography including encryption [10] and signatures [8] is the first application area of PBC. Afterwards, signature schemes with different properties like verifiably encrypted [15], [17], short [12], [6], [5], ring [14] and blind [11] have been proposed and implemented which are summarized in [2]. Here we will focus and combine two signature schemes namely verifiably encrypted signatures and signcryption.

2.3..1 Bilinear Pairings

To define pairings, we start with three groups; $\mathbb{G}_1$ and $\mathbb{G}_2$ are additive abelian group of order q and $\mathbb{G}_3$ is a multiplicative group of order q. A pairing e is a function which maps two elliptic curve points which are elements of $\mathbb{G}_1$ and $\mathbb{G}_2$ to one element of a finite field $\mathbb{G}_3$.

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

e is used in cryptographic schemes when it satisfies the following properties:

a) e is bilinear: For all $P_1, Q_1 \in \mathbb{G}_1$ and $P_2, Q_2 \in \mathbb{G}_2$ we have $e(P_1 + Q_1, P_2) = e(P_1, P_2)e(Q_1, P_2)$ and $e(P_1, P_2 + Q_2) = e(P_1, P_2)e(P_1, Q_2)$
b) e is non-degenerate: For all $P_1 \in \mathbb{G}_1$, with $P_1 \neq 0$ there is some $P_2 \in \mathbb{G}_2$ such that $e(P_1, P_2) \neq 1$ and for all $P_2 \in \mathbb{G}_2$, with $P_2 \neq 0$ there is some $P_1 \in \mathbb{G}_1$ such that $e(P_1, P_2) \neq 1$

Consequtive properties of bilinearity are:

- $e(P_1, 0) = e(0, P_2) = 1$
- $e(-P_1, P_2) = e(P_1, P_2)^{-1} = e(P_1, -P_2)$
- $e([a]P_1, P_2) = e(P_1, P_2)^a = e(P_1, [a]P_2)$ for all $a \in \mathbb{Z}$

Above 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 [4] as a summary, information about pairing friendly field arithmetics in [7] and much more details in [18].

2.3..2 Modified Pairings [18]

In [19], pairings are classified into three types. Here we will use Type I [19] supersingular curves for pairing instantiation in which $\mathbb{G}_1 = \mathbb{G}_2$. Today Type II and Type III pairings are more popular but since our reference signcryption method [1] is implemented in Type I we also used them. Let $\mathbb{G}_1$ be a subgroup of $E(\mathbb{F}_q)$. There is a distortion map φ which maps $\mathbb{G}_1$ into $E(\mathbb{F}_{q^k})$ and the modified pairing $\hat{e}(P_1, P_2) : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_3$ for $P_1, P_2 \in \mathbb{G}_1$

is defined by:
$\hat{e}(P_1, P_2) = e(P_1, \varphi(P_2))$ as shown in section X in [18].

3. Verifiably Encrypted Signcryption Scheme

Y.Han et. al. [1] have developed a signcryption method which was also extended to multi-recipient environment. We adapted this scheme to the verifiably encrypted signature scheme and called it shortly as VE-Signcrypt. The **Setup**, **Extract**, **Signcrypt**, **DeSigncrypt** steps are same as the original work [1]. Here are all the steps;

Setup : Let $\mathbb{G}_1$ be additive group of prime order q which is an n-bit prime 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 $s_T \in \mathbb{Z}_q^*$. l is the bit length of elements in $\mathbb{G}_1$. Set $Y_T = [s]P$ choose cryptographic hash functions $H_1 : \{0,1\}^m X\mathbb{G}_1 \rightarrow \mathbb{G}_1$ and $H_2 : \mathbb{G}_1^3 \rightarrow \{0,1\}^{m+l}$. Publish the system parameters $(\mathbb{G}_1, \mathbb{G}_3, q, \hat{e}, P, Y_T, H_1, H_2)$

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

- TTP or PKG computes $Y_T = [s]P$ as public key and s as private key.
- User ID computes $Y_{ID} = [X_{ID}]P$ as public key and $X_{ID} \in \mathbb{Z}_q^*$ as private key.

Signcrypt : Sender ID=S with key pair (Y_S, X_S) sends a signcrypted message m to receiver ID=R with public key Y_R. Sender S picks a random $r \in \mathbb{Z}_q^*$, computes $U = [r]P, V = X_S H_1(m, rY_R), Z = (m||V) \oplus H_2(U, Y_R, rY_R)$ and output the signcryption $(U, Z) \in \mathbb{G}_1 X\{0,1\}^*$.

DeSigncrypt : Given a signcryption (U, Z) and public key of sender S, receiver R computes $(m||V) = Z \oplus H_2(U, Y_R, X_R U), h = H_1(m, X_R U)$ and then check if $\hat{e}(P, V) = \hat{e}(Y_S, h)$
if check passes, output $< m, (U, V, Y_R, X_R U), Y_S >$ as signature.

The correction of verification for a valid signcryption (U, Z) is as follows;
Since $X_R U = X_R rP = rY_R, Z$ signcryption can be decrypted successfully, then,
$\hat{e}(P, V) = \hat{e}(P, X_S H_1(m, rY_R))$
$= \hat{e}(P, V) = \hat{e}(X_S P, H_1(m, rY_R))$
$= \hat{e}(P, V) = \hat{e}(Y_S, h)$

VE-Signcrypt : Sender ID=S with key pair (Y_S, X_S) sends a verifiably encrypted signcrypted message m to receiver ID=R with public key Y_R and with public key Y_T of adjudicator . Sender S picks two random $r_1$and $r_2 \in \mathbb{Z}_q^*$, computes $U_1 = [r_1]P, U_2 = [r_2]P, V = X_S H_1(m, r_1 Y_R) + r_2 Y_T, Z = (m||V) \oplus H_2(U_1, Y_R, r_1 Y_R)$ and output the signcryption $(U_1, U_2, Z) \in \mathbb{G}_1^2 X\{0,1\}^*$.

De-VE-Signcrypt : Given a verifiably encrypted signcryption (U_1, U_2, Z) and public key of sender S, receiver R computes $(m||V) = Z \oplus H_2(U_1, Y_R, X_R U_1), h = H_1(m, X_R U_1)$ and then check if $\hat{e}(P, V) = \hat{e}(Y_S, h)\hat{e}(U_2, Y_T)$
if check passes, output $< m, (U_1, U_2, V, Y_R, X_R U_1), Y_S >$ as verifiably encrypted signature.

The correction of verification for a valid verifiably encrypted signcryption (U_1, U_2, Z) is as follows;
Since $X_R U = X_R rP = rY_R, Z$ signcryption can be decrypted successfully, then,
$\hat{e}(P, V) = \hat{e}(P, [X_S H_1(m, r_1 Y_R) + r_2 Y_T])$
$= \hat{e}(P, V) = \hat{e}(P, X_S H_1(m, r_1 Y_R))\hat{e}(P, r_2 Y_T)$
$= \hat{e}(P, V) = \hat{e}(X_S P, H_1(m, r_1 Y_R))\hat{e}(r_2 P, Y_T)$
$= \hat{e}(P, V) = \hat{e}(Y_S, h)\hat{e}(U_2, Y_T)$

Adjudication : Given the adjudicator's pri-

vate 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) The correction of adjudication for a valid verifiably encrypted signature (U_1, U_2, V) is as follows;
$V_1 = V - [s_T]U_2 = V - [s_T][r_2]P = V - [r_2]Y_T = X_S H_1(m, r_1 Y_R) + [r_2]Y_T - [r_2]Y_T = X_S H_1(m, r_1 Y_R)$
Here the receiver can not send the original verifiably encrypted signcryption (U_1, U_2, Z) to adjudicator since Z is encrypted for receiver. The adjudication process can be done by only (U_1, U_2, V) provided, but in a fair protocol adjudicator shall make some verifications, in that case De-VE-Signcrypted tuple $< m, (U_1, U_2, V, Y_R, X_R U_1), Y_S >$ as verifiably encrypted signature can be sent to adjudicator.

4. Multi-Recipient Verifiably Encrypted Signcryption Scheme

In this section we extended the verifiably encrypted signature scheme described in the previous section to multi-recipient environment and called it shortly as MR-VE-Signcrypt. The **Setup**, **Extract**, **MR-Signcrypt**, **MR-DeSigncrypt** steps are the same as in the original work [1]. Here are all the steps;

Setup : Let $\mathbb{G}_1$ be an additive group of prime order q and $\mathbb{G}_3$ be a 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 $Y_T = [s]P$ choose cryptographic hash functions $H_1 : \{0,1\}^* X \mathbb{G}_1^2 \rightarrow \mathbb{G}_1$ and $H_2 : \mathbb{G}_1^3 \rightarrow \{0,1\}^*$. Publish the system parameters $(\mathbb{G}_1, \mathbb{G}_3, q, \hat{e}, P, Y_T, H_1, H_2)$

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

- TTP or PKG computes $Y_T = [s]P$ as a public key and s as a private key.
- User ID computes $Y_{ID} = [X_{ID}]P$ as a public key and $X_{ID} \in \mathbb{Z}_q^*$ as a private key.

MR-Signcrypt : Sender ID=S with key pair (Y_S, X_S) sends messages m_i to receivers ID=$R_i, i = 1, ..., n$ with public keys Y_{R_i}. Sender S picks a random $r \in \mathbb{Z}_q^*$, computes $U = [r]P$
For i=1 to n;

- $V_i = X_S H_1(m_i, rY_{R_i})$,
- $Z_i = (m_i || V_i) \oplus H_2(U, Y_{R_i}, rY_{R_i})$

End For
Finally output the signcryptions $(U, Z_i) \in \mathbb{G}_1 X \{0,1\}^{n+1}$.

MR-DeSigncrypt : Given a signcryption for receiver $R_i, (U, Z_i)$ and public key of sender S, receiver R_i computes $(m_i || V_i) = Z_i \oplus H_2(U, Y_{R_i}, X_{R_i} U), h_i = H_1(m_i, X_{R_i} U)$ and then check if $\hat{e}(P, V_i) = \hat{e}(Y_S, h_i)$
if check passes, output $< m, (U, V_i, Y_{R_i}, X_{R_i} U), Y_S >$ as signature.

The correction of verification for a valid signcryption (U, Z_i) is as follows;
Since $X_{R_i} U = X_{R_i} rP = rY_{R_i}, Z_i$ signcryption can be decrypted successfully, then,
$\hat{e}(P, V_i) = \hat{e}(P, X_S H_1(m_i, rY_{R_i}))$
$= \hat{e}(P, V_i) = \hat{e}(X_S P, H_1(m_i, rY_{R_i}))$
$= \hat{e}(P, V_i) = \hat{e}(Y_S, h_i)$

MR-VE-Signcrypt : Sender ID=S with key pair (Y_S, X_S) sends verifiably encrypted messages m_i to receivers ID=$R_i, i = 1, ..., n$ with public keys Y_{R_i} and public key Y_T of adjudicator. Sender S picks two random r_1 and $r_2 \in \mathbb{Z}_q^*$, computes $U_1 = [r_1]P, U_2 = [r_2]P, V = X_S H_1(m, r_1 Y_R) + r_2 Y_T, Z = (m || V) \oplus H_2(U_1, Y_R, r_1 Y_R)$
For i=1 to n;

- $V_i = X_S H_1(m_i, rY_{R_i}) + r_2 Y_T$,

- $Z_i = (m_i||V_i) \oplus H_2(U, Y_{R_i}, rY_{R_i})$

EndFor
Finally output the verifiable encrypted signcryptions $(U_1, U_2, Z_i) \in \mathbb{G}_1^2 X\{0,1\}^{n+1}$.

MR-De-VE-Signcrypt : Given a verifiably encrypted signcryption (U_1, U_2, Z_i) and public key of sender S, receiver R_i computes $(m_i||V_i) = Z_i \oplus H_2(U_1, Y_{R_i}, X_R U_1), h_i = H_1(m_i, X_{R_i} U_1)$ and then check if $\hat{e}(P, V_i) = \hat{e}(Y_S, h_i)\hat{e}(U_2, Y_T)$
if check passes, output $< m_i, (U_1, U_2, V_i, Y_{R_i}, X_{R_i} U_1), Y_S >$ as verifiably encrypted signature.

The correction of verification for a valid verifiably encrypted signcryption (U_1, U_2, Z_i) is as follows;
Since $X_{R_i} U = X_{R_i} rP = rY_{R_i}, Z_i$ signcryption can be decrypted successfully, then,
$\hat{e}(P, V_i) = \hat{e}(P, [X_S H_1(m_i, r_1 Y_{R_i}) + r_2 Y_T])$
$= \hat{e}(P, V_i) = \hat{e}(P, X_S H_1(m_i, r_1 Y_{R_i}))\hat{e}(P, r_2 Y_T)$
$= \hat{e}(P, V_i) = \hat{e}(X_S P, H_1(m_i, r_1 Y_{R_i}))\hat{e}(r_2 P, Y_T)$
$= \hat{e}(P, V_i) = \hat{e}(Y_S, h_i)\hat{e}(U_2, Y_T)$

Adjudication : Given the adjudicator's private key s_T and a valid verifiably encrypted signature (U_1, U_2, V_i) for a message m_i, compute $V_{1_i} = V_i - [s_T]U_2$ and output the original signature (U_1, V_{1_i}) The correction of adjudication for a valid verifiably encrypted signature (U_1, U_2, V_i) is as follows;
$V_{1_i} = V_i - [s_T]U_2 = V_i - [s_T][r_2]P = V_i - [r_2]Y_T = X_S H_1(m_i, r_1 Y_{R_i}) + [r_2]Y_T - [r_2]Y_T = X_S H_1(m_i, r_1 Y_{R_i})$
Here the receiver can not send the original verifiably encrypted signcryption (U_1, U_2, Z_i) to adjudicator since Z_i is encrypted for receiver. The adjudication process can be done by only (U_1, U_2, V_i) provided, but in a fair protocol adjudicator shall make some verifications, in that case MR-De-VE-Signcrypted tuple $< m, (U_1, U_2, V_i, Y_{R_i}, X_{R_i} U_1), Y_S >$ as verifiably encrypted signature can be sent to adjudicator.

As stated in [1], this scheme supports multi message to multi recipient. When $m_1 = m_2 = ...m_n = m$ then this scheme becomes a single message to multi recipient. When $R_1 = R_2 = ...R_n$ then this scheme becomes a single message to single recipient.

5. Fair Two-Party Secret Contract Signing Protocol

In this section we propose a fair two-party optimistic secret contract signing protocol. We propose two alternative ways to define protocol; in the first case we use single recipient verifiably encrypted signcryption and in the second case we use multi recipient verifiably encrypted signcryption defined in previous sections.

5.1. First Case

Here is the steps for the first case with single recipient verifiably encrypted signcryption.

Step 1 $S \rightarrow R$:
$ID_S, ID_R, VESigncrypt\{ID_S, ID_R, m\}$

Step 2 $R \rightarrow S$:
$ID_R, ID_S, Signcrypt\{ID_R, ID_S, m\}$

Step 3 $S \rightarrow R$:
$ID_S, ID_R, Signcrypt\{ID_S, ID_R, m\}$

- Step 1: Sender S computes verifiably encrypted signcryption (U_1, U_2, Z) of $\{ID_S, ID_R, m\}$ where m is the single message as secret contract. And sends to receiver $R < ID_S, ID_R, (U_1, U_2, Z) >$
- Step 2: Receiver R checks the validity of $< ID_S, ID_R, (U_1, U_2, Z) >$ by De-VE-Signcrypt (U_1, U_2, Z). If De-VE-Signcrypt successes then output and keeps

$< ID_S, ID_R, m, (U_1, U_2, V, Y_R, X_R U_1), Y_S >$ as verifiably encrypted signature and sends back to Sender S $< ID_R, ID_S, Signcrypt\{ID_R, ID_S, m\}$, otherwise aborts the protocol.

- Step 3: Sender S checks the validity of $< ID_R, ID_S, (U, Z) >$ by De-Signcrypt (U, Z). if check passes, output and keeps $< m, (U, V, Y_R, X_R U), Y_S >$ as signature and sends back to Receiver R $ID_S, ID_R, Signcrypt\{ID_S, ID_R, m\}$, otherwise aborts the protocol.

If Receiver gets signcryption computed in step three and verification that signcryption passes than the protocol ends by success, otherwise Receiver can request arbitrament from adjudicator. Here is the steps for Adjudication;

- Step 1: Receiver R sends De-VE-Signcrypted $< ID_S, ID_R, m, (U_1, U_2, V, Y_R, X_R U_1), Y_S >$ to adjudicator as verifiably encrypted signature. And computes an ordinary signature as $U = [r]P, V = X_R H_1(m, rY_S)$ and sends also ordinary signature $< m, (U, V, Y_S, rY_S), Y_R >$
- Step 2: Adjudicator checks the validity of ordinary signature $< m, (U, V, Y_S, rY_S), Y_R >$ as $\hat{e}(P, V) = \hat{e}(Y_R, h)$ where $h = H_1(m, rY_S)$ if check fails then aborts the protocol. Otherwise adjudicator $(U_1, U_2, V, Y_R, X_R U_1)$ outputs the original signature (U_1, V_1) and checks the contract and identities and sends back to Receiver R (U_1, V_1). Then sends to Sender S ordinary signature $< m, (U, V, Y_S, rY_S), Y_R >$

5.2. Second Case

Here is the steps for the second case with multi recipient verifiably encrypted signcryption.

Step 1 $S \rightarrow R : ID_S, ID_R,$
$MR - VESigncrypt\{ID_S, ID_R, m\},$
$MR - VESigncrypt\{ID_S, ID_ADJ, m\}$

Step 2 $R \rightarrow S : ID_R, ID_S,$
$MR - Signcrypt\{ID_R, ID_S, m\},$
$MR - Signcrypt\{ID_R, ID_ADJ, m\}$

Step 3 $S \rightarrow R : ID_S, ID_R,$
$MR - Signcrypt\{ID_S, ID_R, m\},$
$MR - Signcrypt\{ID_S, ID_ADJ, m\}$

- Step 1: Sender S computes multi-recipient verifiably encrypted signcryption (U_1, U_2, Z_1, Z_2) of $\{ID_S, ID_R, ID_{ADJ}, m\}$ where m is the single message as secret contract. And sends to receiver R $< ID_S, ID_R, (U_1, U_2, Z_1, Z_2) >$
- Step 2: Receiver R checks the validity of $< ID_S, ID_R, (U_1, U_2, Z_1, Z_2) >$ by MR-De-VE-Signcrypt (U_1, U_2, Z_1, Z_2). If MR-De-VE-Signcrypt successes then output and keeps $< ID_S, ID_R, m, (U_1, U_2, V_1, Y_R, X_R U_1), Y_S >$ as verifiably encrypted signature and sends back to Sender S $< ID_R, ID_S, MR - Signcrypt\{ID_R, ID_S, ID_{ADJ}, m\}$, otherwise aborts the protocol.
- Step 3: Sender S checks the validity of $< ID_R, ID_S, ID_{ADJ}, (U, Z_1, Z_2) >$ by MR-De-Signcrypt (U, Z_1, Z_2). if check passes, output and keeps $< m, (U, V_1, Y_R, X_R U), Y_S >$ as signature and sends back to Receiver R $ID_S, ID_R, ID_{ADJ}, MR - Signcrypt\{ID_S, ID_R, ID_{ADJ}, m\}$, otherwise aborts the protocol.

If Receiver gets signcryption computed in step three and verification that signcryption passes than the protocol ends by success, otherwise Receiver can request arbitrament from adjudicator. Here is the steps for Adjudication;

- Step 1: Receiver R sends original message sent in Step 1 to adjudicator as (U_1, U_2, Z_1, Z_2)

of $\{ID_S, ID_R, ID_{ADJ}, m\}$. And also sends original message sent in step 2 to adjudicator as $< ID_R, ID_S, MR - Signcrypt\{ID_R, ID_S, ID_{ADJ}, m\}$

- Step 2: Adjudicator checks the validity of $< ID_S, ID_R, ID_{ADJ}, (U_1, U_2, Z_1, Z_2) >$ by MR-De-VE-Signcrypt (U_1, U_2, Z_1, Z_2) and checks the validity of $< ID_R, ID_S, ID_{ADJ}, (U, Z_1, Z_2) >$ by MR-De-Signcrypt (U, Z_1, Z_2). If the second check fails then aborts the protocol but if the first check fails or the contract in two messages are different than requests MR-De-VE-Signcrypted version of (U_1, U_2, Z_1, Z_2) from Receiver. If MR-De-VE-Signcrypted version as $< ID_S, ID_R, m, (U_1, U_2, V_1, Y_R, X_R U_1), Y_S >$ validates then adjudicates the first message $< ID_S, ID_R, ID_{ADJ}, (U_1, U_2, Z_1, Z_2) >$ outputs the original signature (U_1, V_1) and checks the contract and identities and sends back to Receiver R (U_1, V_1). Then sends to Sender S ordinary signature $< m, (U, V, Y_S, rY_S), Y_R >$

6. Security and Performance Analysis

There are three security notions that a verifiably encrypted signcryption should satisfy, namely confidentiality, unforgeability and opacity. Confidentiality and unforgeability is required for both signcryption and verifiably encrypted signcryption while opacity is required for only verifiably encrypted signcryption.

Confidentiality and unforgeability for signcryption has been shown in the random oracle model under the hardness of CDH in [1]. Since our scheme's signcryption part is same as the original work, we will present security analysis regarding confidentiality and unforgeability for verifiably encrypted signcryption.

Opacity means that, given a verifiably encrypted signature, it is not possible to get a valid signature on the same message and the same recepient. By this respect we can define opacity for verifiably encrypted signcryption scheme as; given a verifiably encrypted signcryption text, it is not possible to get a valid signcryption on the same message and the same recepient.

6.1. Confidentiality of VE-Signcrypt

Theorem 6.1. *In the random oracle model, if there is an adversary $\mathbb{A}_0$ that performs an attack against IND-CCA2 of our VE-Signcrypt with non-negligible advantage ϵ running time in t and performing q_{VeSC} verifiably encrypted signcryption queries, q_{DeVeSC} verifiably encrypted designcryption queries, and q_{H1} and q_{H2} queries to oracles H_1 and H_2, then there is an algorithm $\mathbb{A}_1$ that solves the CDH problem in G_1 with probability $\epsilon' \geq \epsilon - q_{DeVESC}(q_{H_1}/2^{n-1} + q_{H_2}/2^{m+l})$ with running time $t' = t + (5q_{De-VESC} + 2q_{H2})t_p + 4q_{VESC}t_{sm}$.*

Proof: With the help of $\mathbb{A}_0$ we can construct an adversary $\mathbb{A}_1$ for solving the CDH problem. When $\mathbb{A}_1$ is given with (P, aP, bP), he runs $\mathbb{A}_0$ as a subalgorithm to find the solution abP. Since VE-Signcrypt processes are based on signcryption processes of [1], hash, VE-Signcrypt and De-VE-Signcrypt queries are similar to work [1]. $\mathbb{A}_1$ constructs three lists L_1, L_2, L_3 for oracle queries H_1, H_2 and to simulations of VE-Signcrypt and De-VE-Signcrypt.
H_1 and H_2 simulations are same as [1] except when returning hP from H_1 oracle, $\mathbb{A}_1$ maintains another list L_3 as (h, r_2P, r_2Y_T) as r_2 picked randomly for each query.
VE-Signcrypt Simulation: When a VE-Signcrypt query for (m, Y_R) chosen by $\mathbb{A}_0$, $\mathbb{A}_1$ checks first if $Y_R \notin \mathbb{G}_1$ or $Y_R = Y_S$ or $Y_R = Y_T$,

then rejects the query. Otherwise $\mathbb{A}_1$ picks randomly $r_1 \in \mathbb{Z}_q^*$, computes the result of $U_1 = r_1P$, then simulates $H_1(m, r_1Y_R)$ and gets hP from list L_1 and (r_2P, r_2Y_T) from L_3. Sets $U_2 = r_2P$ and $V = X_SH_1(m, r_1Y_R) + r_2Y_T = hY_S + r_2Y_T = h(bP) + r_2Y_T$ and computes the result of $Z = (m||V) \oplus H_2(U_1, Y_R, r_1Y_R)$ and output the signcryption (U_1, U_2, Z) with sender's public key $Y_S = bP$.

De-VE-Signcrypt Simulation: When a VE-Signcrypted test (U_1, U_2, Z) arrives, $\mathbb{A}_1$ checks first if (U_1, Y_R, F_i, v_i) is in the list L_2, for $0 \leq i \leq q_{H_2}$, such that $Z \oplus v_i = m_i||V_i$ for the corresponding elements (m_i, F_i, h_i) in list L_1 and corresponding r_2Y_T in list L_3, which satisfies $V_i = h_ibP + r_2Y_T$. If one of them satisfies $\hat{e}(P, F_i) = \hat{e}(U_1, Y_R)$ and $\hat{e}(P, V_i) = \hat{e}(Y_{S_i}, h_i)\hat{e}(U_2, Y_T$ then returns (m_i, U_1, U_2, V_i) to $\mathbb{A}_0$, else reutrns 0.

Second stage of proof is same as [1] except the probability and running time as follows. For the queries on H_1 the probability is no more than $q_{H_1}/2^n + q_{H_1}/2^n = q_{H_1}/2^{n-1}$ and for the queries on H_2 the probability is no more than $q_{H_2}/2^{m+l}$. Hence the probability of adversary $\mathbb{A}_1$ wins is $\epsilon' \geq \epsilon - q_{DeVESC}(q_{H_1}/2^{n-1} + q_{H_2}/2^{m+l})$

For the running time of adversary $\mathbb{A}_1$, we only count pairing and scalar multiplication operations. Its running time is evaluated as, 5 pairing operations for each De-VE-Signcrypt simulation, 2 pairing operation 4 scalar multiplication operations for each VE-Signcrypt simulation which includes H_1 and H_2 oracles. so the overall running time is $t' = t + (5q_{De-VESC} + 2q_{H2})t_p + 4q_{VESC}t_{sm}$ where t_p stands for pairing evaluation time and t_{sm} stands for scalar multiplication evaluation time. □

6.2. Unforgeability of VE-Signcrypt

Theorem 6.2. *In the random oracle model, if there is a forger $\mathbb{F}_0$ that forges a valid VE-Signcryption text with non-negligible advantage ϵ running time in t and performing q_{VeSC} verifiably encrypted signcryption queries, q_{DeVeSC} verifiably encrypted designcryption queries, and q_{H1} and q_{H2} queries to oracles H_1 and H_2, then there is an algorithm $\mathbb{F}_1$ that solves the CDH problem in G_1 with probability $\epsilon' \geq \epsilon - (q_{VESC}(q_{H_1} + 1)/2^n$ with running time $t' = t + q_{VESC}(2q_{H2})t_p + (2q_{VESC} + 3q_{VESC}q_{H1})t_{sm}$.*

Proof:

With the help of $\mathbb{F}_0$ we can construct an adversary $\mathbb{F}_1$ for solving the CDH problem. When $\mathbb{F}_1$ is given with (P, aP, bP), he runs $\mathbb{F}_0$ as a subalgorithm to find the solution abP. $\mathbb{F}_1$ constructs three lists L_1, L_2, L_3 for oracle queries H_1, H_2 and to simulation of VE-Signcrypt except H_1 returns haP instead of hP. In the second stage $\mathbb{F}_0$ produces signcryption text (U_1', U_2', Z'). $\mathbb{F}_1$ validates the text as $\hat{e}(P, V') = \hat{e}(Y_S, H')\hat{e}(U_2', Y_T$ if it is a valid verifiably encrypted signcryption text. And if $H_1(m', r_1Y_R)$ is in the list L_1 and (r_2P, r_2Y_T) is in the list L_3 it is easy to see that $V' = habP + r_2Y_T$, then $\mathbb{F}_1$ can compute $abP = h^{-1}(V' - r_2Y_T)$.

The probability of adversary $\mathbb{F}_1$ wins is not different than the probability of [1] as $\epsilon' \geq \epsilon - (q_{VESC}(q_{H_1} + 1)/2^n$. The running time of adversary $\mathbb{F}_1$ sums up, 2 pairing operation for each H_2 query, 3 scalar multiplication operations for each H_1 query and 2 scalar multiplication operations for each VE-Signcrypt simulation. So the running time of $\mathbb{F}_1$ is $t' = t + q_{VESC}(2q_{H2})t_p + (2q_{VESC} + 3q_{VESC}q_{H1})t_{sm}$ where t_p stands for pairing evaluation time and t_{sm} stands for scalar multiplication evaluation time.

□

6.3. Opacity of VE-Signcrypt

Since adjudication can only be applied to verifiably encrypted signature (U_1, U_2, V) we can consider opacity attack like forgery in Theorem 6.2

except list L_3 is not provided and $Y_T = aP$, $U_2 = (bP - hP)$, then $V' = V - r_2 P_T = V - s_T U_2$ and $V' = haP - a(bP - hP)$ so $\mathbb{F}_1$ can compute $abP = -1(V')$ with same propability and running time as in Theorem 6.2.

6.4. Performance Analysis

We compare our VE-Signcrypt with [1] to give computational overheads of adding verifiably encryption to signcryption. Here SM, PC, PA, FM, H_1, H_2 denotes scalar multiplication, pairing computation, point addition in $\mathbb{G}_1$, field multiplication in $\mathbb{G}_3$, hash functions 1 and 2, respectively. In Table 1 and Table 2 there is a minor computational overhead of adding verifiably encryption. Since the **Setup**, **Extract**, **Signcrypt**, **DeSigncrypt**, **MR-Signcrypt**, **MR-DeSigncrypt** steps are same as the original work there is no overhead in these steps. For single recipient case extra 2 SM, 1 PA and 1 PC, 1 FM is added for **VE-Signcrypt** and **VE-DeSigncrypt**, respectively. For multi-recipient case extra 2 SM, n PA and n PC, n FM is added for **MR-VE-Signcrypt** and **MR-VE-DeSigncrypt**, respectively.

TABLE 1
Comparison of our scheme with [1] for single recipient

	[1]	Proposed
Key Gen	1 SM for each user	1 SM for each user
Sign	3 SM, 1 H1, 1 H2	3 SM, 1 H1, 1 H2
Design	1 SM, 1 H1, 1 H2, 2 PC	1 SM, 1 H1, 1 H2, 2 PC
VE-Sign	-	5 SM, 1 H1, 1 H2, 1 PA
VE-Desig	-	1 SM, 1 H1, 1 H2, 3 PC, 1 FM
Adj	-	1 SM, 1 PA

TABLE 2
Comparison of our scheme with [1] for multi-recipient

	[1]	Proposed
Key Gen	n SM	n SM
Sign	(2n+1) SM, n H1, n H2	(2n+1) SM, n H1, n H2
Design	n SM, n H1, n H2, 2n PC	n SM, n H1, n H2, 2n PC
VE-Sign	-	(2n+3) SM, n H1, n H2, n PA
VE-Design	-	n SM,n H1,n H2,3n PC,n FM
Adj	-	1 SM, n PA

7. Conclusion

In this paper we propose a new scheme (up to our knowledge the first) which combines signcryption and verifiably encrypted signatures which we call VESigncrypt, extent it to multi-recipient environment and called it shortly as MR-VE-Signcrypt and use this scheme in a fair two-party optimistic secret contract signing protocol. Implementation of the proposed scheme is left as a future work to see the real-time performance results for different elliptic curves based on security bit lengths.

Acknowledgments

We would like to thank to Prof. Ersan AKYILDIZ and Assoc. Prof. Sedat AKLEYLEK for their valuable guidance.

References

[1] Y. Han, X. Gui and X. Wang, ”Multi-Recipient Signcryption for Secure Wireless Group Communication”, *Cryptology ePrint Archive,* 2008/253, https://eprint.iacr.org/2008/253, 2008.
[2] R. Dutta, P. Barua, P.Sarkar, ”Pairing Based Cryptography: A Survey”, *Cryptology ePrint Archive,* 2004/064, https://eprint.iacr.org/2004/064, 2004.
[3] C. Calik, O. Sever, H.M. Yildirim, Z. Yuce, ”A Survey of Certified Electronic Mail Protocols”, *Proceedings of ISCTurkey 2010,* Ankara, Turkey, pp.45-50 06-08 May 2010.

[4] S. Akleylek, B.B. Kirlar, O. Sever, Z. Yuce, ”Pairing Based Cryptography: A Survey”, *Proceedings of ISCTurkey 2008,* Ankara, Turkey, pp.121-125, 25-27 December 2008.

[5] S. Akleylek, B.B. Kirlar, O. Sever, Z. Yuce, ”A New Short Signature Scheme with Random Oracle from Bilinear Pairings”, *Journal of Telecommunications and Information Technology,* Vol.3, No.1, pp.5-10, 2011.

[6] S. Akleylek, B.B. Kirlar, O. Sever, Z. Yuce, ”Short Signature Scheme from Bilinear Pairings”, *Information Assurance and Cyber Defense (IST-091),* Tallinn, Estonia, 2010.

[7] S. Akleylek, B.B. Kirlar, O. Sever, Z. Yuce, ”Arithmetic on Pairing-Friendly Fields”, *Proceedings of ISCTurkey 2008,* Ankara, Turkey, pp. 115-120, 25-27 December 2008.

[8] K.G. Paterson, ”ID-Based Signatures from Pairings on Elliptic Curves”, *Cryptology ePrint Archive,* 2002/004, https://eprint.iacr.org/2002/004, 2002.

[9] O. Sever, E. Akyildiz, ”Improved Contract Signing Protocol Based on Certificateless Hybrid Verifiably Encrypted Signature Scheme”, *Proceedings of ISCTurkey 2015,* Ankara, Turkey, 30-31 October 2015.

[10] D. Boneh, M. Franklin, ”Identity Based Encryption from Weil Pairing” *SIAM J.of Computing,* Vol.32, No.3, pp.586-615, 2003.

[11] 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.

[12] D. Boneh, B. Lynn, H. Shacham, ”Short Signatures from the Weil Pairing”, *Proceedings of Asiacrypt* https://www.iacr.org/archive/asiacrypt2001/22480516.pdf, 2001.

[13] D.Boneh, C.Gentry, B.Lynn, and H.Shacham, ”Aggregate and Verifiably Encrypted Signatures from Bilinear Maps”, *Advances in Cryptology - Eurocrypt 2003,* LNCS Vol.2656, Springer, pp.416-432, 2003.

[14] F. Zhang, K. Kim. ”ID-Based Blind Signature and Ring Signature from Pairings”, *Advances in Cryptology in AsiaCrypt,* 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,* Springer-Verlag, 2003.

[16] C. Bamboriya, S.R. Yadav, ”A Survey of Different Contract Signing Protocols”, *IJETAE* Vol.1, No.4, January 2014.

[17] L. Chen, C. Gu, ”Optimistic Contract Signing Protocol Based on Hybrid Verifiably Encrypted Signature” *Advances in Information Sciences and Service Sciences(AISS)* Vol.4, No.12, 2012.

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

[19] S.D. Galbraith, K.G. Paterson, N.P. Smart, ”Pairings for Cryptographers”, Cryptology ePrint Archive, Report, 165, https://eprint.iacr.org/2006/165, 2006.

[20] Y. Zheng, ”Digital Signcryption or How to Achieve Cost”, *Advances in Cryptology - CRYPTO97,* (B.S.Kaliski Jr.,ed.), LNCS Vol.1294, Springer, pp.165-179, 1997.

[21] I. Jeong, H. Jeong, H. Rhee, D. Lee and J. Lim, ”Provably Secure Encrypt-then-Sign Composition in Hybrid Signcryption”, *Information Security and Cryptology*, ICISC 2002, pp.16-34, 2002.