A Novel Strong Designated Verifier Signature Scheme without Random Oracles

Maryam Rajabzadeh Asaar[1], Mahmoud Salmasizadeh [2]
[1] Department of Electrical Engineering, [2] Electronics Research Institute (Center), Sharif University of Technology, Tehran, Iran.
asaar@ee.sharif.ir,salmasi@sharif.edu

Abstract. In this study, a novel pairing based strong designated verifier signature scheme based on non-interactive zero knowledge proofs is proposed. The security of the proposal is presented by sequences of games without random oracles; furthermore, this scheme has a security proof for the property of privacy of the signer's identity in comparison with the scheme proposed by Zhang et al. in 2007. In addition, this proposal compared to the scheme presented by Huang et al. in 2011 supports non-delegatability. The non-delegatability of our proposal is achieved since we do not use the common secret key shared between the signer and the designated verifier in our construction. Furthermore, if a signer delegates her signing capability which is derived from her secret key on a specific message to a third party, then, the third party cannot generate a valid designated verifier signature due to the relaxed special soundness of the non-interactive zero knowledge proof. To the best of our knowledge, this construction is the first attempt to generate a designated verifier signature scheme with non-delegatability in the standard model, while satisfying of non-delegatability property is loose.

Keywords : strong designated verifier signature, registered public key model, random oracle model.

1 Introduction

Jakobsson et al. [13] introduced the notion of designated verifier proofs (DVP) in 1996. These proofs allow a signer (Alice) to designate a verifier (Bob) and prove the validity of a statement only to Bob; while Bob cannot use this transcript to convince anyone else. This motivates non-transferability and is generally achieved by proving either the validity of the statement or the knowledge of Bob's secret key. Consequently, Bob can always generate the same transcript. A designated verifier signature (DVS) is the non-interactive version of the DVP. A DVS is publicly verifiable and a valid DVS is generated by Alice or Bob. The DVS is applied in various cryptographic schemes such as voting [13], undeniable signature [5, 7, 9], deniable authentication [25] where it is required that only designated entities can be convinced of several statements. It is desirable that a third party except Alice and Bob cannot tell whose signature is sent to Bob. A DVS with this property is called a strong designated verifier signature (SDVS)[13]. The strength of a SDVS as privacy of a signer's identity (PSI) is formalized by Laguillamie and Vergnand in 2004 [16]. A valid designated verifier signature for Bob on behalf of Alice is generated if and only if the secret key of either Alice or Bob is known. This property means non-delegatability for signing and is introduced by Lipmaa et al. [18] in 2005.

1.1 Related Work

Several variants for DVS such as ring signatures [19, 20], universal designated verifier signatures (UDVS) [8, 9, 14, 21, 24, 27], multi-designated verifier signatures [13, 15], and identity-based designated verifier signatures (IBDVS)[4, 10, 11, 23], and (SDVS)[4, 10] are proposed. Several DVS schemes [16, 17, 21, 22] are shown to be delegatable since the notion of non-delegatability [18] is introduced, while there are a few non-delegatable DVS schemes [11, 18, 28] in the random oracle model [2]. Since 2007, two SDVS schemes in the standard model are proposed in [12] and [28], respectively. To present a non-delegatable SDVS scheme without random oracles is an open problem as aforementioned in [12].

1.2 Contribution

In this paper, a novel strong designated verifier scheme without random oracles is proposed. We use the paradigm which is slightly analogous to the Bellare and Goldwasser's paradigm [1]; this paradigm combines the ordinary signature proposed by Waters [26] and non-interactive zero knowledge (NIZK) proof without random oracles [6]. Our proof of non-delegatability evidently is not a proof of knowledge, the proof with standard special soundness, of the secret key of the signer or the designated verifier. The non-delegatability of our proposal is achieved since we do not use the common secret key shared between the signer and the designated verifier in our construction. Furthermore, if a signer delegates her signing capability which is derived from her secret key on a specific message to a third party, then, the third party cannot generate a valid designated verifier signature due to the relaxed special soundness of the NIZK proof. To the best of our knowledge, this is the first attempt to generate a designated verifier signature scheme with non-delegatability in the standard model, while satisfying of non-delegatability property is due to the relaxed special soundness of the NIZK proof instead of standard special soundness of these proofs [6]. Moreover, we prove the security of the proposal, i.e. unforgeability and privacy of the signer's identity using sequences of games without random oracles. This scheme has a security proof for the property of privacy of the signer's identity in comparison with the scheme proposed by Zhang et al. in 2007 [28], and this proposal compared to the scheme presented by Huang et al. in 2011 [12] supports non-delegatability as well.

1.3 Outline of the paper

The rest of this manuscript is organized as follows. Section 2 presents a number of preliminaries, bilinear pairings and complexity assumptions, as the signature foundation. The model of SDVS including outline of the SDVS scheme and its security properties are described in section 3. The proposed scheme and its formal security proofs are presented in section 4. Section 5 presents the conclusion. Appendices are given in sections 6 and 7.

2 Preliminaries

In this section, we review several fundamental backgrounds employed in this research, including bilinear pairings and complexity assumptions.

2.1 Bilinear pairings

Let G and G_T be two cyclic multiplicative groups of prime order p; furthermore, let g be a generator of G. The map $e : G \times G \longrightarrow G_T$ is said to be an admissible bilinear pairing if the following conditions hold true.

1. e is bilinear, i.e. $e(g^a, g^b) = e(g, g)^{ab}$ for all a and $b \in Z_p$
2. e is non-degenerate, i.e. $e(g, g) \neq 1_{G_T}$
3. e is efficiently computable.

We refer readers to [3] for more details on the construction of bilinear pairings.

2.2 Definitions and complexity assumptions

Definition 1 (Problem generator $\widetilde{g}_{dl} = (G_{dl}, g_{dl})$). G_{dl} outputs an instance $t = (p, \gamma, h = \gamma^x)$, where p is prime, p is k-bit long, and γ is an element of Z_p and $h = \gamma^x \bmod p$. In this case, the solution is $g_{dl}(t) = x$.
An algorithm A is said to completely break $\widetilde{g}_{dl}$ if it is able to solve any instance t with non-negligible probability [6].

Definition 2 (Problem generator $\widetilde{H}_{Paillier} = (H_{Paillier}, h_{Paillier})$). $H_{Paillier}(k')$ outputs an instance (n, c), where n is k'-bit RSA modules along with $c = (1 + n)^{\alpha} r^n \bmod n^2$ (i.e. c is a paillier encryption of α), where α is chosen at random in some given interval. The solution is $h_{Paillier}(n, c) = \alpha$.
An algorithm A is said to completely break $\widetilde{H}_{Paillier}$ if it is able to solve any instance (n, c) with non-negligible probability [6].

Assumption 1 . $\widetilde{H}_{Paillier}$ is 2-harder than $\widetilde{g}_{dl}$. This assumption is presented in [6]; furthermore, the reasonability of this assumption is discussed in [6].

Assumption 2 (Decisional Diffie-Hellman (DDH)assumption). Given $(g, g^a, g^b, Z \in Z_p)$ for some unknown a, $b \in_R Z_p$, there is no probabilistic polynomial-time algorithm A that can decide if $Z = g^{ab}$ or Z is a random element from $Z \in Z_p$ with non-negligible probability ϵ_{ddh}.

Assumption 3 (Semantic security, security for indistinguishable chosen plaintext attack (IND-CPA Security) of $\widetilde{H}_{Paillier} = (H_{Paillier}, h_{Paillier})$). A cryptosystem with a security parameter k' is semantically secure if there is no probabilistic polynomial time algorithm A on inputs (α_0, α_1) and $c^* = E_{pk_{1V}}(\alpha_b)$, where $b \in_R \{0, 1\}$ can output the correct plaintext inside c^* with non-negligible probability ϵ_{ind} (a negligible quantity in k' for all sufficiently large k').

3 Model of strong designated verifier signature schemes

In this section, we review the outline and security properties of the strong designated verifier signature schemes.

3.1 Outline of designated verifier signature schemes

There are two participants in a designated verifier signature scheme, the signer S and the designated verifier V. A designated verifier signature scheme consists of five algorithms as follows.

- Setup: Given a security parameter k, this algorithm outputs the system parameters.
- Key generation: It takes the security parameter k as its input and outputs the secret-public key (sk_i, pk_i) for $i \in \{S, V\}$.

- Signing: This algorithm (signing oracle O_s) takes the signer's secret key sk_S, the designated verifier's public key pk_V, and a message M as its inputs to generate a signature θ.

- Verification: This algorithm (verification oracle O_v) takes the designated verifier's secret key sk_V, the signer's public key pk_S, the message M, and the signature θ as its inputs and returns 1 if the signature is valid, otherwise returns 0 indicating the signature is invalid.

- Transcript simulation: This algorithm (simulation oracle O_{sim}) takes the designated verifier's secret key sk_V, the signer's public key pk_S, and a message M as its inputs to output an identically distributed transcript θ' which is indistinguishable from the one generated by the signer.

3.2 Security properties of designated verifier signature schemes

A SDVS scheme ought to be unforgeable, non-transferable, and satisfy the privacy of the signer's identity. An SDVS is said to be non-delegatable if it satisfies non-delegatability. Informal definitions of these properties are expressed as follows.

1. Correctness: A properly formed SDVS must be accepted by the verifying algorithm. Formally, the correctness of the SDVS requires that for any (pk_S, sk_S), (pk_V, sk_V) and any message $M \in \{0,1\}^*$, we have $pr[ver(sk_V, pk_S, pk_V, M, \theta = sign(sk_S, pk_S, pk_V, M)) = 1] = 1$.

2. Unforgeability: It requires that no one other than the signer S and the designated verifier V can produce a valid designated verifier signature. The formal definition of unforgeability [13] is expressed in Definition 3, Appendix A.

3. Non-transferability: This property means that it should be infeasible for any PPT distinguisher to tell whether σ on a message M was generated by the signer S or simulated by the designated verifier V. The formal definition of non-transferability [13] is expressed in Definition 4, Appendix A.

4. Privacy of the Signer's Identity (PSI): A SDVS has the property of PSI if no one can tell signatures generated by the signer S_0 for a V is different from signatures generated by the signer S_1 for the V in case of not knowing the secret key of the V. The formal definition of this property [16] is given in Definition 5, Appendix A.

5. Non-delegatability: It requires that if one generates a valid designated verifier signature on a message, it must "know" the secret key of either S or V. Therefore, a signature is a proof of knowledge of secret key of either S or V. The formal definition of non-delegatability [18] is presented in Definition 6, Appendix A.

4 Our designated verifier signature scheme

In this section, we describe our designated verifier signature scheme. There are two participants in the system the signer S and the designated verifier V. In the following, all the messages to be signed will be represented as bit strings of length y. Our scheme consists of five algorithms as follows.

1. Setup: The system parameters are as follows. Let (G, G_T) be bilinear groups where $|G| = |G_T| = p$ for some prime p with k-bit length; further, let g be the generator of G. e denotes an admissible pairing $e : G \times G \longrightarrow G_T$. Pick $m' \in G$, and a vector $\boldsymbol{m} = (m_i)$ of length y, whose entries are random elements from G. The public parameters are $(G, G_T, e, m', \boldsymbol{m})$.

2. Key generation: The signer S picks randomly x_{1S} and $x_{2S} \in Z_p^*$, then, the signer S computes her public key $pk_S = (pk_{1S}, pk_{2S}) = (g^{x_{1S}}, g^{x_{2S}})$, where the signer secret key is $g^{x_{1S}x_{2S}}$. Furthermore, it is assumed that every designated verifier has a secret-public key pair $(sk_{1V}, pk_{1V}) = ((p', q'), n) \leftarrow_R H_{Paillier}(k')$, where $n = p'q'$ is a k'-bit RSA modulus with two large primes p' and q' for the paillier's cryptosystem with the condition $k' > l_z(k) + 1$, where $l_z(k)$ is the bit length of the plaintext inside the paillier's cryptosystem.
   Moreover, the designated verifier V chooses another secret key $sk_{2V} = \alpha$ with $l_\alpha(k)$-bit length and sets the corresponding public key to the encryption of the message α under the public key pk_{1V} i.e., $pk_{2V} = c = E_{pk_{1V}}(\alpha) = (1 + n)^\alpha r^n \bmod n^2$, where r is uniformly selected in Z_n^*. $D_{sk_{1V}}(E_{pk_{1V}}(\alpha)) = \alpha$ is the decryption of $E_{pk_{1V}}(\alpha)$ and the output is α.
   The public keys of the designated verifier are $(pk_{1V}, pk_{2V}) = (n, c)$ and corresponding secret keys are $(sk_{1V}, sk_{2V}) = ((p', q'), \alpha)$.

   Before we explain the details of the proposal, the overview of the proposal is presented: a signature of a user on a message M is $\theta = (\sigma_1, \sigma_2) = (g^{x_{1S}x_{2S}}(M^\beta), g^\beta)$, where $g^{x_{1S}x_{2S}}$ is the secret key of the signer. The signature θ of the signer with the public key (pk_{1S}, pk_{2S}) is verified as $e(\sigma_1, g) = e(\sigma_2, g)e(pk_{1S}, pk_{2S})$. To convert the signature to a designated verifier signature, we do not include σ_2 in the signature; however instead, we set σ_2 to be a NIZK proof showing that σ_1 is binding to either the signer or the designated verifier. The non-delegatability of our proposal is achieved since we do not use the common secret key shared between the signer and the designated verifier in our construction. However, if a signer delegates her signing capability which is derived from her secret key on a specific message to a third party, then, the third party cannot generate a valid designated verifier signature due to the relaxed special soundness of the NIZK proof.

3. Signing. Let M be an y-bit message to be signed by the signer S and M_i denotes the i-bit of M, and $\widetilde{M} \subseteq \{1, 2, ..., y\}$ be the set of all i for which $M_i = 1$, the designated

verifier signature is generated as follows. First, the signer S picks random values β and $r \in_R Z_p^*$, then, the designated verifier signature $\theta = (\sigma_1, \sigma_2 = (a_1, \dot{z_1} = E_{pk_{1V}}(z_1)))$ on M is constructed as expressed in Eq.(1).

$$\begin{aligned} \sigma_1 &= g^{x_{1S}x_{2S}}(m' \prod_{i \in \widetilde{M}} m_i)^{\beta} \\ a_1 &= e((m' \prod_{i \in \widetilde{M}} m_i), g)^r \\ \dot{z_1} &= E_{pk_{1V}}(z_1) = E_{pk_{1V}}(r) pk_{2V}^{\beta} \end{aligned} \tag{1}$$

4. Verifying. To check whether θ is a valid designated verifier signature on the message M, the designated verifier V uses his secret key sk_{1V} to decrypt $\dot{z_1}$; then, he checks whether Eq.(2) holds.

$$e((m' \prod_{i \in \widetilde{M}} m_i), g)^{z_1} = a_1 (\frac{e(\sigma_1, g)}{e(pk_{1S}, pk_{2S})})^{\alpha} \tag{2}$$

If the equality holds, the designated verifier V accepts the signature θ; otherwise, the designated verifier V rejects it.

5. Simulation of a transcript. The designated verifier V can use his secret key α to simulate a signature which is indistinguishable from the one generated by the signer S on an arbitrary message M. He picks random values σ_1 and $z_1 \in_R Z_p^*$, then, the designated verifier signature $\theta = (\sigma_1, \sigma_2 = (a_1, \dot{z_1} = E_{pk_{1V}}(z_1)))$ on M is constructed as expressed in Eq. (3).

$$\begin{aligned} \dot{z_1} &= E_{pk_{1V}}(z_1) \\ a_1 &= \frac{e(m' \prod_{i \in \widetilde{M}} m_i, g)^{z_1}}{(\frac{e(\sigma_1, g)}{e(pk_{1S}, pk_{2S})})^{\alpha}} \end{aligned} \tag{3}$$

4.1 Analysis of the scheme

In this section, we will primarily show the correctness of the proposed scheme. Subsequently, we prove that the proposal is secure without random oracles.

Correctness. The correctness of the scheme is clear by inspection.
$D_{sk_{1V}}(\dot{z_1})$ equals the correct value $z_1 = r + \beta\alpha$ since $k' > l_{z_1}(k)$ which ensures that $z_1 < n$. Moreover, we have

$$\begin{aligned} & e((m' \prod_{i \in \widetilde{M}} m_i), g)^{z_1} \\ &= e((m' \prod_{i \in \widetilde{M}} m_i), g)^{r+\beta\alpha} \\ &= e((m' \prod_{i \in \widetilde{M}} m_i), g)^r e((m' \prod_{i \in \widetilde{M}} m_i), g)^{\beta\alpha} \\ &= a_1((e((m' \prod_{i \in \widetilde{M}} m_i), g))^{\beta})^{\alpha} \\ &= a_1 (\frac{e(\sigma_1, g)}{e(pk_{1S}, pk_{2S})})^{\alpha} \end{aligned} \tag{4}$$

As we shall see later (Theorem 2), the scheme is perfectly non-transferable. Hence, making query to O_{sim} is equivalent to making query to signing oracle O_s in the games of unforgeability and privacy of the signer's identity.

Theorem 1. If there exists an adversary A who can $(t, q_s, q_{sim}, q_v, \epsilon)$ forge the designated verifier signature scheme, then there exists another algorithm A_1 that can use A to break the DDH assumption with probability ϵ_{ddh} in time $t_1 \simeq t$ and an algorithm A_2 that breaks the semantic security of the public-key cryptosystem $H_{Paillier}$ with probability ϵ_{ind} in time $t_2 \simeq t$, where $\epsilon < \epsilon_{ddh} + \epsilon_{ind} + (q_s + q_{sim} + q_v)\frac{1}{p}$.

Proof. The theorem is proved by a series of games; it is supposed that A is an adversary which can violate the unforgeability of the scheme. Let G_i be the i-th game, and X_i be the event that A outputs a successful forgery in game G_i.

- G_0: This is the original game. The challenger C selects α, a and $b \in_R Z_p^*$; furthermore, C chooses two large numbers p' and q' such that $n = p'q'$, then, it sets $(pk_{1S}, pk_{2S}, pk_{1V}, pk_{2V}) = (g^a, g^b, n, c)$. (sk_{1V}, pk_{1V}) is given to the verification oracle O_v. A_1 invokes A on inputs $(pk_{1S}, pk_{2S}, pk_{1V}, pk_{2V}) = (g^a, g^b, n, c)$. The secret key employed to generate a designated verifier signature is $K_s = g^{ab}$, while the secret key employed to simulate a designated verifier signature is $K_{sim} = \alpha$. For each signature query M, C generates the corresponding answer with K_s instead of $g^{x_{1S}x_{2S}}$ as aforementioned in Eq.(1). For each simulation query M, C simulates the corresponding answer as aforementioned in Eq.(3). For each verification query (M, θ), C responses with the appropriate bit indicating the validity of the signature, 1 if the signature is valid, 0 otherwise. It should be noted that O_v verifies the signature (M, θ) as aforementioned in Eq.(2) and outputs the correct bit. The adversary A finally outputs a successful forgery (M^*, θ^*) with probability $pr[X_0] = \epsilon$, which M^* is new and θ^* satisfies the Eq.(2).

- G_1: This game is different from the game G_0 in which the secret key, K_s used to generate a designated verifier signature is chosen at random from Z_p, i.e. $K_s = K \in_R Z_p$. Moreover, the validity of the adversary's forgery is checked w.r.t. the random key, K. If successful probabilities of the adversaries in games G_0 and G_1 differs non-negligibly, it leads to an algorithm A_1 which breaks the DDH assumption with probability ϵ_{ddh} in time $t_1 \simeq t$. Consequently, we have $|pr[X_1] - pr[X_0]| \leq \epsilon_{ddh}$.

  To prove the above equation, it is assumed that a random instance of the DDH problem, i.e. g, g^a, g^b, and $K \in_R Z_p$ is given; A_1' goal is to decide whether $K = g^{ab}$ or K is a random element of Z_p. A_1 sets $(pk_{1S}, pk_{2S}, pk_{1V}, pk_{2V}) = (g^a, g^b, n, c)$ and invokes A on these inputs. A_1 computes Eq. (1) to answer a signing or (simulation) query, while $g^{x_{1S}x_{2S}}$ is replaced with K; A_1 responses to a verification query (M, θ) with the appropriate bit which O_v outputs w.r.t. K. Note that O_v computes Eq.(2), while $e(pk_{1S}, pk_{2S})$ is substituted with $e(g, K)$. Finally, A outputs a successful forgery (M^*, θ^*); A_1 checks whether (M^*, θ^*) is a valid signature w.r.t. K with the help of O_v. If $K = g^{ab}$, A_1 outputs 1 which O_v outputs; otherwise, it outputs 0 if K is randomly chosen from Z_p. Let j be a bit output by A_1; therefore, we have $pr[j = 1|K = g^{ab}] - pr[j = 1|K \in_R Z_p]$.

  If $K = g^{ab}$, the game simulated by A_1 is game G_0; then, success probability of A is $pr[X_0]$. If K is a random element of Z_p, the game simulated by A_1 is game G_1; hence, the success probability of A is $pr[X_1]$. Therefore, we have $pr[j = 1|K = g^{ab}] - pr[j = 1|K \in_R Z_p] =$

$|pr[X_1] - pr[X_0]|$; as a result, we have $|pr[X_1] - pr[X_0]| \leq \epsilon_{ddh}$ by the DDH assumption.

– G_2: This game is different from the game G_1 in which the secret key of the designated verifier inside c, α', is chosen at random by A_2. Besides, the validity of the forgery is checked w.r.t. the random keys K and α'. A_2 invokes A on inputs $(pk_{1S}, pk_{2S}, pk_{1V}, pk_{2V}) = (g^a, g^b, n, c)$. If success probabilities of the adversaries in games G_2 and G_1 differ non-negligibly, it leads to an algorithm for breaking semantic security of $H_{Paillier}$. As a result, we have $|pr[X_2] - pr[X_1]| \leq \epsilon_{ind}$.

To prove the above equation, it is assumed a random instance of the ciphertext created by the cryptosystem $H_{Paillier}$, i.e. c^* is given; A_2's goal is to decide if the plaintext inside c^* is α or α'. A_2 sets $(pk_{1S}, pk_{2S}, pk_{1V}, pk_{2V}) = (g^a, g^b, n, c^*)$ and invokes A on these inputs. To answer a signing or a simulation query, A_2 picks random values σ and $z_1 \in_R Z_p^*$ and generates (M, θ) as computed in Eq.(3). A_2 responses to a verification query (M, θ) with the help of O_v w.r.t. α' and K. Finally, A outputs a successful forgery (M^*, θ^*); A_2 checks whether (M^*, θ^*) is a valid signature w.r.t. K and α' with the help of O_v. If O_v outputs 1 meaning $D_{sk_{1V}}(c^*) = \alpha'$; otherwise, $D_{sk_{1V}}(c^*) = \alpha$. Therefore, we have
$pr[\alpha' \leftarrow A_2(c^*)] - pr[\alpha \leftarrow A_2(c^*)]$.

If $\alpha' = D_{sk_{1V}}(c^*)$, the game simulated by A_2 is game G_2; then, success probability of A is $pr[X_2]$. If $\alpha = D_{sk_{1V}}(c^*)$, the game simulated by A_2 is game G_1; hence, the success probability of A is $pr[X_1]$. Therefore, we have $pr[\alpha' \leftarrow A_2()c^*)] - pr[\alpha \leftarrow A_2(c^*)] = |pr[X_2] - pr[X_1]|$. As a result, we have $|pr[X_2] - pr[X_1]| \leq \epsilon_{ind}$ by the semantic security assumption.

In game G_2, the signatures on a message M is chosen at random from Z_p^* after issuing $q_s + q_{sim} + q_v$ queries because of knowing α'. As a result, the probability of a valid forgery output by A in game G_2 is upper bounded by $pr[X_2] \leq (\frac{1}{p} - q_s - q_{sim} - q_v)^{-1} < (q_s + q_{sim} + q_v)\frac{1}{p}$.

Hence, we have $\epsilon = pr[X_0] \leq |pr[X_0] - pr[X_1]| + |pr[X_1] - pr[X_2]| + pr[X_2] \leq \epsilon_{ddh} + \epsilon_{ind} + (q_s + q_{sim} + q_v)\frac{1}{p}$.

Theorem 2. The proposal is non-transferable.
Proof. To prove non-transferability of the scheme, we show that the signature simulated by the designated verifier V is indistinguishable from that generated by the signer S. As a result, we have to show that the two following distributions are identical.

$$\theta = \begin{cases} \beta \in_R Z_p^* \\ r \in_R Z_p^* \\ a_1 = e((m' \prod_{i \in \widetilde{M}} m_i), g)^r \\ \dot{z_1} = E_{pk_{1V}}(z_1) = E_{pk_{1V}}(r) pk_{2V}^{\beta} \end{cases} \tag{5}$$

and

$$\theta' = \begin{cases} z'_1 \in_R Z_p^* \\ \sigma'_1 \in_R Z_p^* \\ \dot{z_1'} = E_{pk_{1V}}(z'_1) \\ a'_1 = \frac{e(m' \prod_{i \in \widetilde{M}} m_i, g)^{z'_1}}{(\frac{e(\sigma'_1, g)}{e(pk_{1S}, pk_{2S})})^{\alpha}} \end{cases} \tag{6}$$

Let $\overline{\theta}$ be a valid signature which is randomly chosen from the set of all valid signer's signatures intended to the verifier V. Subsequently, we have distributions of probabilities as follows:

$$Pr_{\theta} = Pr[\theta = \overline{\theta}] = \frac{1}{(p-1)^2}, \tag{7}$$

and

$$Pr_{\theta'} = Pr[\theta' = \overline{\theta}] = \frac{1}{(p-1)^2} \tag{8}$$

The analysis means both distributions of probability are the same. Hence, our proposal satisfies perfect non-transferability.

Theorem 3. If there exists an adversary D that can (t, q_s, q_v, ϵ) break the PSI of the scheme, then there exists another algorithm A_1 who can use D to break DDH assumption with probability ϵ_{ddh} in time $t_1 \simeq t$ and there is an algorithm A_2 who can use D to break the semantic security of the public-key cryptosystem $H_{Paillier}$ with probability ϵ_{ind} in time $t_2 \simeq t$, where $\epsilon \leq 2\epsilon_{ddh} + 2\epsilon_{ind} + \frac{1}{2}$.

Proof. The security proof is given in Appendix B due to the lack of space.

We first informally discuss the property of non-delegatability, then, theorem 4 is given for non-delegatability. If a signer would like to delegate her signing capability to a third party, she could give (σ_1, a_1) to the third party. However, the third party could not generate a valid designated verifier signature since the third party does not have β, having β is equivalent to having the secret key of the signer, or α, the secret key designated verifier.

Someone may claim that the signature is delegatable with the help of the designated verifier such that the designated verifier gives $(\frac{e(\sigma_1, g)}{e(pk_{1S}, pk_{2S})})^{\alpha}$ and $\sigma_1 \in_R Z_p^*$ to a third party. The third party computes the valid designated verifier signature $\theta = (\sigma_1, \sigma_2 = (a_1, \dot{z_1} = E_{pk_{1V}}(z_1)))$ on an arbitrary message M, where $z_1 \in_R Z_p^*$ and $a_1 = \frac{e(m' \prod_{i \in \widetilde{M}} m_i, g)^{z_1}}{(\frac{e(\sigma_1, g)}{e(pk_{1S}, pk_{2S})})^{\alpha}}$. However, the designated verifier can distinguish who generates the signature due to the existence of the σ_1 which he delegated before as soon as he receives the designated verifier signature.

Satisfying of non-delegatability property is loose (we use relaxed special soundness property) since it evidently is not a proof of knowledge (to have a proof of knowledge, we need standard special soundness property) of the secret key of the signer or the designated verifier. To prove

the proposed scheme is non-delegatable, we show that if there is a forger for the designated verifier signature, we can construct another algorithm that outputs the second secret key of the designated verifier in time $O(T(k)+poly(k))$ with the help of an algorithm that completely breaks $\widetilde{g}_{dl} = (G_{dl}, g_{dl})$ in $T(k)$, where $poly()$ is a polynomial. If $T(k)$ is polynomial in k, then $O(T(k) + poly(k)) = poly(k)$. Similarly, if $T(k)$ is superpolynomial, then $O(T(k) + poly(k))$ is superpolynomial in k.

Theorem 4. If there exists a probabilistic polynomial-time forger F that can generate valid signatures, then, the assumption 1 that $\widetilde{H}_{Paillier}$ is 2-harder than $\widetilde{g}_{dl} = (G_{dl}, g_{dl})$ is contradicted.

Proof. It is supposed that there is a probabilistic polynomial-time forger F for the designated verifier signature scheme. We construct another algorithm B breaking $\widetilde{H}_{Paillier}$ on instances (pk_{1V}, pk_{2V}) of size $k' \geq 2k$ using F and any algorithm A_1 that completely breaks $\widetilde{g}_{dl} = (G_{dl}, g_{dl})$ in running time $t_1 = T(k)$, where $\gamma = e((m' \prod_{i \in \widetilde{M}} m_i), g)$.

B invokes F on $(pk_{1S}, pk_{2S}, pk_{1V}, pk_{2V})$ to obtain a valid signature $\theta = (\sigma_1, \sigma_2 = (a_1, \dot{z_1} = E_{pk_{1V}}(z_1)))$ on M; then, B will run A_1 on $a_1 = e((m' \prod_{i \in \widetilde{M}} m_i), g)^r$ to obtain r and will run A_1 on $\frac{e(\sigma_1, g)}{e(pk_{1S}, pk_{2S})} = e((m' \prod_{i \in \widetilde{M}} m_i), g)^{\beta}$ to attain β. On the other hand, the ciphertext $\dot{z_1} = E_{pk_{1V}}(r+\beta\alpha)$ is specified as a pair of integers (u, w) such that $\dot{z_1} = E_{pk_{1V}}(u) pk_{2V}^w \bmod n^2$. Finally, B outputs the secret key of the designated verifier $sk_{2V} = \alpha = \frac{r-u}{w-\beta}$ inside $pk_{2V} = c$ with non-negligible probability in time $O(T(k) + poly(k))$, where $poly(k)$ is the required time to forge a valid signature. Hence, the conclusion contradicts the assumption that $\widetilde{H}_{Paillier}$ is 2-harder than $\widetilde{g}_{dl} = (G_{dl}, g_{dl})$.

5 conclusion

We propose a novel pairing based strong designated verifier signature scheme based on non-interactive zero knowledge proofs. The security of the proposal is presented by sequences of games without random oracles; furthermore, this scheme has a security proof for the property of privacy of the signer's identity in comparison with the scheme proposed by Zhang et al. in 2007. In addition, this proposal compared to the scheme presented by Huang et al. in 2011 supports non-delegatability.To the best of our knowledge, this construction is the first attempt to generate a designated verifier signature scheme with non-delegatability in the standard model, while satisfying non-delegatability property is loose.

References

1. Bellare, M., Goldwasser, S., *New paradigms for digital signatures and message authentication based on non-interactive zero knowledge proofs,* Advances in Cryptology - Crypto 1989, Lecture Notes in Computer Science, pp. 194-211, Springer-Verlag (1990).
2. Bellare, M., Rogaway, P., *Random oracles are practical: a paradigm for designing efficient protocols,* ACM Conference on Computer and Communications Security, pp. 62-73, ACM (1993).

3. Boneh, D., Franklin, M., *Identity-based encryption from the Weil pairings,* Advances in Cryptology - Crypto 2001, vol. 3494 of Lecture Notes in Computer Science, pp. 213-229, Springer-Verlag (2001).
4. Bhaskar, R., Herranz, J., Laguillaumie, F. *Aggregate designated verifier signatures and application to secure routing,* International Journal of Security Network, vol. 2(3/4), pp.192-201, (2007).
5. Chaum, D., van Antwerpen, H., *Undeniable signatures,* Proceedings of Advances in Cryptology-CRYPTO 1989, vol. 435 of Lecture Notes in Computer Science, pp. 212-216. Springer (1989).
6. Damgard, I., Fazio, N., Nicolosi, A., *Non-interactive zero knowledge from homomorphic encryption,* Theory of Cryptography, Third Theory of Cryptography Conference, vol. 3876 of Lecture Notes in Computer Science, pp. 41-59. Springer (2006).
7. Huang, X., Mu, Y., Susilo, W., Wu, W., *Provably secure pairing based convertible undeniable signature with short signature length,* Proceedings of 1st International Conference on Pairing-Based Cryptography, Pairing 2007, vol. 4575 of Lecture Notes in Computer Science, pp. 367-391, Springer (2007).
8. Huang, X., Susilo, W., Mu, Y., Wu, W., *Universal designated verifier signature without delegatability,* Proceedings of 8th International Conference on Information and Communications Security, ICICS 2006, vol. 4307 of Lecture Notes in Computer Science, pp. 479-498, Springer (2006).
9. Huang, X., Susilo, W., Mu, Y., Wu, W., *Secure universal designated verifier signature without random oracles,* International Journal of Information Security, vol. 7(3), pp. 171-183, (2007).
10. Huang, X., Susilo, W., Mu, Y., Zhang, F., *Short designated verifier signature scheme and its identity-based variant,* International Journal of Network Security, vol. 6(1), pp.82-93, (2008).
11. Huang, Q., Yang, G., Wong, D. S., Susilo, W., *Identity-based strong designated verifier signature revisited,* International Journal of Systems and Software, vol.84(1), pp.120-129, 2011.
12. Huang, Q., Yang, G., Wong, D. S., Susilo, W., *Efficient strong designated verifier signature schemes without Random Oracle or with non-delegatability,* International Journal of Information Security, Springer, pp.373-385, 2011.
13. Jakobsson, M., Sako, K., Impagliazzo, R., *Designated verifier proofs and their applications,* Proceedings of Advances in Cryptology-EUROCRYPT 1996, vol. 1070 of Lecture Notes in Computer Science, pp. 143-154, Springer (1996).
14. Laguillaumie, F., Libert, B., Quisquater, J.-J., *Universal designated verifier signatures without random oracles or non-black box assumptions,* Proceedings of 5th International Conference on Security and Cryptography for Networks, SCN 2006, vol. 4116 of Lecture Notes in Computer Science, pp. 63-77, Springer (2006).
15. Laguillaumie, F., Vergnaud, D., *Multi-designated verifiers signatures,* Proceedings of 6th International Conference on Information and Communications Security, ICICS 2004, vol. 3269 of LectureNotes in Computer Science, pp. 495-507, Springer (2004b).
16. Laguillaumie,F., Vergnaud,D. , *Designated verifier signature: anonymity and efficient construction from any bilinear map*, Proceedings of 3th International Conference on Security and Cryptography for Networks, SCN 2004, Lecture Notes in Computer Science, pp. 105-119, Springer (2004).
17. Li, Y., Lipmaa, H., Pei, D., *On delegatability of four designated verifier signatures,* Proceedings of 7th International Conference on Information and Communications Security, ICICS 2005, vol.e 3783 of Lecture Notes in Computer Science, pp. 61-71, Springer (2005).
18. Lipmaa, H., Wang, G., Bao, F., *Designated verifier signature schemes: Attacks, new security notions and a new construction,* Proceedings of 32th International Colloquium on Automata, Languages and Programming, ICALP 2005, vol. 3580 of Lecture Notes in Computer Science, pp. 459-471, Springer (2005).
19. Rivest, R., Shamir, A., Tauman, Y., *How to leak a secret,* Boyd C. (ed.) Proceedings of Advances in Cryptology-ASIACRYPT 2001, vol. 2248 of Lecture Notes in Computer Science, pp. 552-565, Springer (2001).

20. Shacham, H.,Waters, B., *Efficient ring signatures without random Oracles,* Okamoto, T., Wang, X. (eds.) Proceedings of Public Key Cryptography 2007, vol. 4450 of Lecture Notes in Computer Science, pp. 166-180, Springer (2007).
21. Steinfeld, R., Bull, L., Wang, H., Pieprzyk, J., *Universal designated verifier signatures,* Proceedings of Advances in Cryptology-ASIACRYPT 2003, vol. 2894 of Lecture Notes in Computer Science, pp. 523-542, Springer (2003).
22. Steinfeld, R.,Wang, H., Pieprzyk, J. *Efficient extension of standard Schnorr/RSA signatures into universal designated verifier signatures,* Proceedings of Public Key Cryptography 2004, vol. 2947 of LectureNotes in Computer Science, pp. 86-100. Springer (2004).
23. Susilo, W., Zhang, F., Mu, Y., *Identity-based strong designated verifier signature schemes,* Proceedings of 9th Australasian Conference on Information Security and Privacy, ACISP 2004, vol. 3108 of LectureNotes in Computer Science, pp. 313-324, Springer (2004).
24. Vergnaud, D., *New extensions of pairing-based signatures into universal designated verifier signatures,* Proceedings of 33th International Colloquium on Automata, Languages and Programming, ICALP 2006, vol. 4052 of Lecture Notes in Computer Science, pp. 58-69, Springer (2006).
25. Wang, B., Song, Z., *A non-interactive deniable authentication scheme based on designated verifier proofs,* Information Sciences, Inf. Sci. 2009, vol. 179(6), pp. 858- 865, 2009.
26. Waters, B., *Efficient identity based encryption without random oracles,* Eurocrypt 2005, vol. 3494 of LectureNotes in Computer Science, pp. 114-127, Springer (2005)
27. Zhang, R., Furukawa, J., Imai, H., *Short signature and universal designated verifier signature without random oracles,* Proceedings of 3rd International Conference on Applied Cryptography and Network Security, ACNS 2005, vol. 3531 of LectureNotes in Computer Science, pp. 483-498, Springer (2005).
28. Zhang, J. and Ji, C., *An efficient designated verifier signature scheme without Random Oracles,* First International Symposium on Data, Privacy and E-Commerce, ISDPE 2007, pp.338-340, 2007.

6 Appendix A

6.1 Formal definitions of Security properties of designated verifier signature schemes

A SDVS scheme ought to be unforgeable, non-transferable, and satisfy the privacy of the signer's identity. An SDVS is said to be non-delegatable if it satisfies non-delegatability. Formal definitions of these properties are expressed as follows.

1. Unforgeability: To have a formal definition for unforgeability, the following game between the simulator C and a probabilistic polynomial time (PPT) adversary A is considered to be played.
   (a) C prepares the key pairs (pk_S, sk_S) for S and (pk_V, sk_V) for V, and gives (pk_S, pk_V) to A.
   (b) A issues queries to the following oracles.
      - O_s: This oracle generates a signature σ on a given message M using sk_S such that this signature is valid w.r.t. pk_S and pk_V, then returns it to A.
      - O_{sim}: This oracle generates a simulated signature σ' on a given message M using sk_V such that this simulated signature is valid w.r.t. pk_S and pk_V, then returns it to A.
      - O_v: This oracle takes a query of the form (M, σ) as an input and returns a bit b which is 1 if σ is a valid signature on M w.r.t. pk_S and pk_V; otherwise, returns 0.

(c) A outputs a forgery (M^*, σ^*) and wins the game if the two following conditions hold
- $Ver(sk_V, pk_S, pk_V, M^*, \sigma^*) = 1$
- It did not query O_s and O_{sim} on input M^*.

The formal definition of unforgeability [13] is expressed in Definition 3.

Definition 3 (Unforgeability). An SDVS scheme is $(t, q_s, q_{sim}, q_v, \epsilon)$-unforgeable if no adversary A which runs in time at most t; issues at most q_s queries to O_s; issues at most q_{sim} queries to O_{sim}; and issues at most q_v queries to O_v can win the above game with probability at least ϵ.

Definition 4 (Non-transferability). An SDVS is non-transferable if there exists a PPT simulation algorithm Sim on sk_V, pk_S, pk_V, and a message M outputs a simulated signature which is indistinguishable from the real signatures generated by the signer on the same message. For any PPT distinguisher A, any (pk_S, sk_S), (pk_V, sk_V), and any message $M \in \{0,1\}^*$, Eq. (1) holds.

$$\left| pr \left[\begin{array}{l} \sigma_0 \longleftarrow Sign(sk_S, pk_S, pk_V, m), \\ \sigma_1 \longleftarrow Sim(sk_V, pk_S, pk_V, m), \\ b \longleftarrow \{0,1\}, \\ b' \longleftarrow A(pk_S, sk_S, pk_V, sk_V, \sigma_b) \\ : b' = b \end{array} \right] - \frac{1}{2} \right| < \epsilon(k) \tag{9}$$

Where $\epsilon(k)$ is a negligible function in the security parameter k, and the probability is taken over the randomness used in $Sign$ and Sim, and the random coins consumed by A. If the probability is equal to $\frac{1}{2}$, the SDVS scheme is perfectly non-transferable or source hiding [13].

2. Privacy of the Signer's Identity (PSI): To have a formal definition for PSI, the following game between the simulator C and the distinguisher D is considered.
   (a) C generates key pairs (pk_{S_0}, sk_{S_0}) for signer S_0, (pk_{S_1}, sk_{S_1}) for signer S_1, and (pk_V, sk_V) for designated verifier V, and invokes D on input pk_{S_0}, pk_{S_1}, and pk_V.
   (b) C issues queries (M, d) to the O_s and O_v which $d \in \{0,1\}$ indicating which signer responds to that query.
   (c) C tosses a coin $d \in \{0,1\}$ for the message M^* submitted by D, then computes the challenge signature $\sigma^* \longleftarrow Sign(sk_{S_d}, pk_{S_d}, pk_V, M^*)$ and returns σ^* to D
   (d) D outputs a bit d' and wins the game if the two following conditions hold.
   - $d' = d$
   - It did not query O_v on input (d, M^*, σ^*) for any $d \in \{0,1\}$

The formal definition of this property [16] is given in Definition 5.

Definition 5 (Privacy of the Signer's Identity). An SDVS scheme is (t, q_s, q_v, ϵ)-PSI-secure if no adversary A which runs in time at most t; issues at most q_s queries to O_s; and q_v queries to O_v can win the aforementioned game with probability that deviated from $\frac{1}{2}$ by more than ϵ.

Definition 6 (Non-delegatability). It is assumed that $\kappa \in [0, 1]$ be the knowledge error and F be a forger algorithm. Let F_M be F with M as its input, and oracle calls to F_M be counted as one step. An SDVS scheme is non-delegatable with knowledge error κ if there exists a positive polynomial poly() and a probabilistic oracle machine B which produces either the secret key of the signer or the secret key of the designated verifier with probability $\frac{\epsilon-\kappa}{poly(k)}$, where $\epsilon > \kappa$ in expected polynomial time with the help of the forger F that forges a valid signature on message M with probability ϵ.

7 Appendix B

Proof of theorem 3. It is assumed that D be a distinguisher against privacy of the signers' identity. Let G_i be i-th game and X_i be the event that D outputs the correct bit, correct signer's identity, in game G_i.

- G_0: This is the original game. The challenger C selects α, a_0, b_0, a_1, and $b_1 \in_R Z_p^*$; furthermore, C chooses two large numbers p' and q' such that $n = p'q'$, then, it sets $(pk_{1S_0}, pk_{2S_0}, pk_{1S_1}, pk_{2S_1}, pk_{1V}, pk_{2V}) = (g^{a_0}, g^{b_0}, g^{a_1}, g^{b_1}, n, c)$. A_1 invokes D on these inputs. The secret keys employed to generate a designated verifier signature are $K_{S_0} = g^{a_0 b_0}$ or $K_{S_1} = g^{a_1 b_1}$, where K_{S_d} is the secret key of signer S_d. For each signature query (M, d), where $d \in_R \{0, 1\}$ presenting the index of the signer, C simulates the corresponding answer as aforementioned in Eq.(1), while C uses K_{S_d} on behalf of $g^{x_{1S} x_{2S}}$. For each verification query (M, θ, d), C responses with the appropriate bit which O_v outputs such that O_v verifies the signature (M, θ, d) as aforementioned in Eq.(2) w.r.t. K_{S_d}. The adversary D finally asks for a designated verifier signature on a challenge message M^*; the challenger C randomly selects one of the two secret keys K_{S_d}, where $d \in_R \{0, 1\}$ to generate the signature of the challenge message, (M^*, θ^*). Then, D returns a bit d' as the signer's identity of the signature. The success probability of D in distinguishing signer's identity of the designated verifier by definition is $pr[X_0] = \epsilon$.

- G_1: This game is different from the game G_0 in which the secret key of the signer S_0, $K_{S_0} = K_0$ used to generate a designated verifier signature is chosen at random from Z_p, i.e. $K_0 \in_R Z_p$. Moreover, the validity of the adversary's verification queries are checked w.r.t. the random key, K_0 in case of $d = 0$. If successful probabilities of the adversaries in distinguishing the identity of the signer in games G_0 and G_1 differ non-negligibly, it leads to an algorithm A_1 which breaks the DDH assumption with probability ϵ_{ddh} in time $t_1 \simeq t$. As a consequence, we have $|pr[X_1] - pr[X_0]| \leq \epsilon_{ddh}$.

  To prove the above equation, we construct another algorithm A_1 to break DDH assumption. It is assumed a random instance of the DDH problem, i.e. g, g^{a_0}, g^{b_0}, and $K_0 \in_R Z_p$ is given;

A_1's goal is to decide whether $K_0 = g^{a_0 b_0}$ or K_0 is a random element of Z_p. A_1 chooses $b_1 \in_R Z_p$ and invokes D on inputs $(pk_{S0}, pk_{S1}, pk_{1V}, pk_{2V}) = ((g^{a_0}, g^{b_0}), (g^{a_1}, g^{b_1}), n, c)$. A_1 sets $K_{S_0} = K_0$ and $K_{S_1} = (g^{a_1})^{b_1}$. When D makes a signing query (M, d), which d indicates the index of the signer; A_1 uses K_{S_d} to generate the signature for V as presented in Eq. (1) such that $g^{x_{1S} x_{2S}}$ is replaced with K_{S_d}. When D makes a verification query (M, θ, d), A_1 returns the corresponding bit presenting the validity or invalidity of the signature using verification oracle O_v w.r.t. $K_{S_0} = K_0$ and $K_{S_1} = (g^{a_1})^{b_1}$. When D submits its challenge message M^*, A_1 randomly chooses one of the two secret keys and computes θ^* as presented in Eq.(2). The successive queries issued by D are handled as aforementioned. Finally, D outputs a bit d'. Then, A_1 outputs 1 if $d' = d$, meaning $K_0 = g^{a_0 b_0}$ and output 0 meaning K_0 is randomly chosen from Z_p. Let j be a bit output by A_1; therefore, we have $pr[j = 1 | K_0 = g^{a_0 b_0}] - pr[j = 1 | K_0 \in_R Z_p]$.

If $K_0 = g^{a_0 b_0}$, the game simulated by A_1 is game G_0; then, success probability of D is $pr[X_0]$. If K_0 is a random element of Z_p, the game simulated by A_1 is game G_1; hence, the success probability of D is $pr[X_1]$. Therefore, we have $pr[j = 1 | K_0 = g^{a_0 b_0}] - pr[j = 1 | K_0 \in_R Z_p] = |pr[X_1] - pr[X_0]|$ and by the DDH assumption we conclude that $|pr[X_1] - pr[X_0]| \leq \epsilon_{ddh}$.

– G_2: This game differs from the game G_1 in which the secret key, K_{S_1} is chosen at random from Z_p, i.e. $K_1 \in_R Z_p$. Moreover, the validity of the adversary's forgery is checked w.r.t. the random key, K_1. Similar to the game G_1, we have $|pr[X_2] - pr[X_1]| \leq \epsilon_{ddh}$.

– G_3: This game is different from the game G_2 in which the secret key of the designated verifier inside c, α', is chosen at random by A_2. Besides, the validity of the verification queries is checked w.r.t. the random keys (K_0, K_1) and α'. A_2 invokes D on inputs $(pk_{S0}, pk_{S1}, pk_{1V}, pk_{2V}) = (g^{a_0}, g^{b_0}, g^{a_1}, g^{b_1}, n, c)$. If success probabilities of the adversaries in game G_3 and G_2 differ non-negligibly, it leads to an algorithm for breaking semantic security of $H_{Paillier}$. As a result, we have $|pr[X_3] - pr[X_2]| \leq \epsilon_{ind}$.

To prove the above equation, it is assumed a random instance of the ciphertext created by the cryptosystem $H_{Paillier}$, i.e. c^* is given; A_2's goal is to decide if the plaintext inside c^* is α or α'. A_2 sets $(pk_{S0}, pk_{S1}, pk_{1V}, pk_{2V}) = ((g^{a_0}, g^{b_0}), (g^{a_1}, g^{b_1}), n, c)$ and invokes D on these inputs. To answer a signing query (M, d), A_2 chooses σ and $z_1 \in_R Z_p^*$ and computes Eq.(3). A_2 responses to a verification query (M, θ, d) with the help of O_v, while verification is performed w.r.t. (K_0, K_1) and α'. When D submits a challenge message M^*, A_2 chooses σ and $z_1 \in_R Z_p^*$ and computes Eq.(3) and returns (M^*, θ^*) to D. Finally, D outputs a bit d'. If $d' = d$ means that $\alpha' = D_{sk_{1V}}(c^*)$; otherwise, $\alpha' = D_{sk_{1V}}(c^*)$. Therefore, we have $pr[\alpha' \leftarrow A_2(c^*)] - pr[\alpha \leftarrow A_2(c^*)]$.

If $\alpha' = D_{sk_{1V}}(c^*)$, the game simulated by A_2 is game G_3; then, success probability of D is $pr[X_3]$. If $\alpha = D_{sk_{1V}}(c^*)$, the game simulated by A_2 is game G_2; hence, the success probability of D is $pr[X_2]$. We have $pr[\alpha' \leftarrow A_2(c^* = c)] - pr[\alpha \leftarrow A_2(c^* = c)] = |pr[X_3] - pr[X_2]|$. As a result, we have $|pr[X_3] - pr[X_2]| \leq \epsilon_{ind}$ by assumption 3.

In game G_3, the signatures including challenge signature on a message M^* is chosen at random from Z_p^* because of knowing α'. As a result, the success probability of the adversary in distinguishing the index of the signer is $pr[X_3] = \frac{1}{2}$. As a consequence, we have $\epsilon = pr[X_0] \leq |pr[X_0] - pr[X_1]| + |pr[X_1] - pr[X_2]| + |pr[X_2] - pr[X_3]| + pr[X_3] \leq 2\epsilon_{ddh} + \epsilon ind + \frac{1}{2}$.