Conditionally Verifiable Signatures

Aldar C-F. Chan, Ian F. Blake

University of Toronto*

May 24, 2005

Abstract

We introduce a new digital signature model, called conditionally verifiable signature (CVS), allowing a signer to specify and convince a recipient under what conditions his signature would become valid or verifiable; the resulting signature is not publicly verifiable immediately but can be converted back into an ordinary one (verifiable by anyone) after the recipient has obtained proofs, in the form of signatures/endorsements from a number of third party witnesses, that all the specified conditions have been fulfilled. A fairly wide set of conditions could be specified in CVS. Besides, the only job of the witnesses is to certify the fulfillment of a condition and none of them need to be actively involved in the actual signature conversion, thus protecting user privacy. We formalize the concept of CVS and define the related security notions. We also derive the relations between these notions. Besides, we give a generic CVS construction based on any identity based encryption (IBE) scheme show that the existence of IBE with semantic security against a chosen plaintext attack (a weaker notion than the standard one) is necessary and sufficient for secure CVS. Finally, we give a number of practical CVS constructions based on bilinear pairings for standard signature schemes like Elgamal and RSA.

Keywords: digital signatures, privacy, accountability, identity based encryption, bilinear pairings.

1 Introduction

Balancing between the accountability and the privacy of the signer is an important but largely unanswered issue of digital signatures. A digital signature scheme usually consists of two parties, a signer and a recipient, with the former giving his signature on a message/document to the latter as his commitment or endorsement on the message. To ensure that the signer is held accountable for his commitment, his signature needs to be publicly verifiable (by anyone) or, at least, verifiable by a mutually trusted third party; otherwise, the signer could deny having signed the document as nobody can prove he really did, and the non-repudiation property (which binds a signer, perhaps legally, to a statement he signs) cannot be achieved. However, public verifiability of a digital signature would put the signer's privacy at risk as a digital signature could be replicated and spread so easily, compared to its handwritten counterpart. More importantly, if the message presents valuable information about the signer, then the signed message itself is a certified piece of that information. Hence, the interests of the signer and the recipient are in conflict.

Of course, ensuring signer privacy and non-repudiation simultaneously seems to be impossible for any signature scheme. But, fortunately, in most real world scenarios, we usually wish to maintain privacy of a digital signature up to a certain instant after it is issued and restore non-repudiation afterwards. This

*Email: { aldar,ifblake } @comm.utoronto.ca

could be better illustrated by the example of future/option trading. In a future trade, the seller signs a contract with the buyer specifying the price and quantity he has agreed with the buyer but the contract is not effective before a future execution date. For reasons like preventing other sellers from manipulating the price or avoiding any adverse effects on further negotiation with other buyers, ideally, before the execution date, the seller does not want anyone able to associate him with the contract, at least ensuring that the buyer is unable to convince others of the validity of their agreement. That is, limited verifiability is desired before the execution date. Whereas, on or after the execution date, an honest seller usually does not worry about his signature being publicly verifiable. In fact, to protect the interest of the recipient, the seller's signature has to be verifiable by others. Hence, we could view reaching the execution date as a certain condition to be fulfilled before the signature of the signer (seller) could be revealed to the recipient (the buyer, who could convince others of the validity of the signer's signature afterwards), and before such fulfillment, we wish to achieve signer privacy. We notice that many business activities involving digital signatures have similar situations. The essence is how the signer could ensure non-verifiability of his signature before certain conditions are fulfilled (in the future trading case, the condition is the execution date has passed) but still can convince the recipient that he will be obligated to exercise his commitment; in other words, he needs to give the recipient some guarantee about his commitment or his signature will become effective or publicly verifiable once all the conditions are fulfilled.

On the other hand, the non-repudiation property of a digital signature could also have a serious repercussion to the signer if there is no way to allow him to control when and how a recipient could obtain his signature when sending it out. In the online world, the lack of physical proximity could render a careful signer hesitant in giving his signature (say for a payment authorization) to another party because he is not given any guarantee that he will obtain what he is supposed to as an exchange of his signature. From the recipient's perspective, if the signer does not send out his signature, the recipient will not give what the signer needs. For instance, if the signer makes an online purchase, he may not receive any guarantee that his order will be delivered but the seller (recipient) will not send it out unless the signer gives out his signature on a payment authorization. This kind of deadlock due to mistrusting parties is not easily solved. In the worst scenario, a careless signer may fall into a fraud trap to give out his signature unwisely. Nevertheless, the deadlock could be partially solved if the signer could ensure that the recipient can never obtain a valid signature of his unless some conditions (specified by the signer) are fulfilled, namely, the recipient sends out the signer's order in the online purchase example.

To provide a flexible solution to this problem of controllably passing signatures from one party to another without actively involving a trusted third party, we introduce a new signature concept called conditionally verifiable signatures (CVS). In a CVS scheme, the signer gives the recipient some seemingly random number, what we call a *partial signature*, and specifies a set of conditions the fulfillment of which will allow the recipient to extract the signer's signature from the partial signature. The partial signature is not immediately verifiable; fulfilling the specified conditions is necessary to retrieve a valid ordinary signature from it. To convince the recipient that his ordinary signature could be extracted form a partial signature, the signer runs a confirmation protocol with the recipient to prove that his signature could be retrieved once all the specified conditions are fulfilled. Before the ordinary signature becomes effective (that is, extracted), the partial signature is no more convincing than any random number, namely, nobody could link the partial signature to its alleged signer. We formulate this property by the notion of *simulatability* in this paper, that is, anyone could use just public information of the signer to simulate a given partial signature while others cannot judge whether it is genuine. In other words, nobody could distinguish between a genuine partial signature and a simulated one. In fact, in our model, even given the signer's private key, nobody could tell the validity of a given partial signature if the random coins used to generate it are not available. In order to enforce the verification of condition fulfillment, we need a

number of third party witnesses mutually trusted by both the signer and recipient. In our model, the only job of these witnesses is to verify whether the given conditions are fulfilled and they are unaware of the conversion or even the existence of the partial signature. That is, the witnesses do not participate in the actual signature conversion. Details of the model are given in the next section.

1.1 Conditionally Verifiable Signature

In the CVS model, a signer is allowed to embed a set of verifiability conditions C into his ordinary signature σ to create a partial signature δ that is solely verifiable by the recipient (possibly through the collaboration with the signer), who cannot immediately convince others of the validity of δ but can convert it back to the universally verifiable one σ (i.e. verifiable by everyone) after obtaining from a number of witnesses (appointed by the signer) the proofs that all the specified verifiability conditions have been fulfilled.[1] These proofs are in the form of signatures on condition statements, signed by the witnesses, about how the specified conditions are considered as fulfilled. In order to convince the recipient to accept a given partial signature δ on a message M (whose validity could not be verified), the signer runs a proof/confirmation protocol, which could be interactive or non-interactive, with the recipient to convince the latter that δ is indeed his partial signature on M, from which the corresponding ordinary signature could be recovered using the specified witnesses' signatures on the specified verifiability condition statements in C.

Given that $\mathcal{W}$ is the set of all possible witnesses, an instance set of verifiability conditions C is of the form $\{(c_i, W_i) : c_i \in \{0,1\}^*, W_i \in \mathcal{W}\}$ where each condition statement c_i is a string of alphabets of arbitrary length describing a condition to be fulfilled. Examples of c_i include "A reservation has been made for Alice on flight CX829, 14 Jul 2005.", "A parcel of XXX has been received for delivery to Bob.", "It is now 02:00AM 18 Jan 2003 GMT.", "An emergency has happened." and so on. The recipient needs to request each one of the specified witnesses, say W_i, to verify whether the condition stated in c_i is fulfilled and in case it is, to sign on c_i to give him a witness signature σ_i. These witness signatures σ_i's would allow the recipient to recover the publicly verifiable, ordinary signature σ from the partial signature δ.

Besides, it is not necessary for a recipient to present the partial signature or the message itself to the witnesses in order to get their endorsements on the statement about the fulfillment of a condition. Even so, the witness signatures could still recover the ordinary signature from the collected witness signatures. The only trust we place on the witnesses is that they only give out their signatures on a condition statement when the specified conditions are indeed fulfilled. In fact, it is not difficult to imagine that the existence of such witnesses is abundant in any business transaction; in most cases, any party involved in processing an order would inherently be trusted by both the signer and recipient, a good candidate as a witness. A typical example is the postal office which is involved in delivering the order the signer placed on the recipient of a signature for his payment authorization. In addition, we could achieve a fairly high level of privacy in that the witnesses are unaware of the message or the partial signature when verifying the fulfillment of a given condition, namely, he does not learn the deal between the signer and the recipient. But this would not hinder the recipient from obtaining a witness signature as it is so common in business processes to request a receipt.

We could view the partial signature as a blinded version of the ordinary signature, that is, nobody could verify its validity. In our CVS formulation, this non-verifiability property is expressed by the notion of *simulatability* — there exists a polynomial time simulator which is computable using only

[1] Throughout the rest of this paper, we will denote the ordinary (universally verifiable) signature and the CVS partial signature by σ and δ respectively, unless otherwise specified.

public information of the signer and outputs a fake signature computationally indistinguishable from the partial signature; that is, even given a genuine partial signature, nobody with bounded computation power could assure that it is not a fake one generated by the simulator. As a result, when the recipient presents a partial signature to others to convince them of its validity, nobody could tell whether the signer has really created it or the recipient has generated it himself using the simulator. Of course, it is natural to worry about whether the confirmation protocol would leak out useful information to help distinguishing between a genuine and a fake partial signature. We show in Section 3 that if the confirmation protocol is zero knowledge, then it would leak no useful information for such a purpose and the CVS scheme is said to be *non-transferable*.

Beside the notion of *simulatability*, there are other possible formulations of the non-verifiability property of a signature, namely, *anonymity* and *invisibility*.[2] Numerous similar notions have stemmed from these two notions in the literature on undeniable signatures [4, 8, 10, 11, 14, 23, 19, 31, 33] and designated confirmer signatures [9, 6, 18, 28, 34, 37]. Anonymity in essence means that given a message, two signers and a valid signature belonging to one of them, nobody could tell which one of the signers has created the given signature. Whereas, invisibility means that given a signer, two messages and a valid signature of the signer for one of messages, nobody could tell which message the given signature is for. These various nations represent different understanding about the security requirement of the non-verifiability of a signature, as well as different modeling. But we think that simulatability is a more natural and comprehensive notion to represent the non-verifiability property of a partial signature. In Section 3.3 we give a detailed treatment on deriving the relationships between these notions and simulatability. In particular, we show that anonymity and invisibility are indeed implicitly implied by simulatability if an appropriate simulator (with more restrictions on its requirement) exists despite that they are not completely compatible and covered by the notion of simulatability.

As usual, unforegeability is a basic requirement for a secure CVS scheme. More specifically, we require that even colluding with all the witnesses and allowed to query ordinary and partial signatures of his choice, nobody could present a message signature pair not previously queried such that the signature is valid for the message. This is often called existentially unforgeability against a chosen message attack.

As mentioned earlier, beside protecting the signer's privacy, CVS is also aimed to protect the signer from fraud trap. It offers the signer the guarantee that the recipient would not get his signature on a document if he could not get what the recipient are committed to. In other words, if the specified conditions are not fulfilled, that is, the corresponding witness signatures are not available, the ordinary signature could never be retrieved from a given partial signature. This is the *cheat-immunity* property of a CVS scheme. We could show that this property is implicitly achieved in an unforgeable and simulatable CVS scheme if its confirmation protocol is also zero knowledge.

1.2 Our Contributions

The main contribution of this paper is the new model of conditionally verifiable signatures through which the signer can incorporate a wide range of verifiability conditions into an ordinary signature scheme to control its verifiability and validity while minimizing the requirement or trust on third-parties. To the best of our knowledge, it is the first scheme of its kind in the literature. Before this work, it is fair to say that the problem of seamlessly incorporating verifiability conditions into a signature scheme to control its validity and allowing spontaneous signature recovery upon the fulfillment of the specified conditions remains largely open. Closely related work includes undeniable signatures [4, 8, 10, 11, 14, 23, 19, 31,

[2] When talking about signatures in this context, we are referring to some blinded version of an ordinary signature in undeniable signatures or designated confirmer signatures.

33], designated confirmer signatures [9, 6, 18, 28, 34, 37], fair exchange [1], and timed release of digital signatures [20, 21]. In fact, we could possibly view CVS as a more general, unified concept incorporating all these, but provides more effective and flexible solutions to the scenarios these existing schemes could not solve satisfactorily, particularly those in digital business or electronic commerce. A typical example of these would be the deadlock scenario mentioned earlier about the online purchase between mistrusting parties; using the post office as a witness, CVS would reasonably solve this problem.

Besides, we give a detailed treatment on modeling the security goals and the adversary capabilities of CVS. We show the relationship between these notions and distill them down into a much smaller set of core notions necessary for a CVS construction to fulfill all of them. In particular, we show that the notions of invisibility and anonymity, usually considered separately in undeniable signatures and designated confirmer signatures, are in essence directly implied in the notion of simulatability (a notion commonly found in commitment schemes and proof-of-knowledge protocols) if an appropriate simulator could be found. Moreover, we give the conditions under which the design of a CVS scheme and its confirmation protocol could be separated for consideration while preserving the needed security.

Furthermore, we demonstrate the feasibility of CVS by giving a generic construction based on any existentially unforgeable signature scheme and any semantically secure identity based encryption scheme. Based on this, we show that a secure CVS scheme is equivalent to an IBE scheme with indistinguishability security against a chosen plaintext attack (IND-ID-CPA) in terms of existence. As (IND-ID-CPA) security is a weaker notion than the commonly accepted security notion against an adaptive chosen ciphertext attack (IND-ID-CCA) in IBE, we believe that CVS could be constructed based on a weaker assumption than IBE.

Finally, we present a number of practical instantiations of CVS based on bilinear pairings. We give efficient CVS constructions for standard signature schemes like ElGamal [16] and RSA [38]. With slight modifications, these techniques could be applied to other signature schemes like Schnorr [40] and GHR [22] signatures.

1.3 Organization of the Paper

The rest of this paper is organized as follows. We discuss related work in the next section. Then, we give the definition of a conditionally verifiable signature scheme and its notions of security and derive relationships between these notions in Section 3. After that, we present the preliminary materials needed in our construction in Section 4. In Section 5, we give a generic CVS construction and show the equivalent between CVS and IBE. In Section 6, we give a number of efficient CVS constructions based on bilinear pairings. Finally, we conclude in Section 8 with a number of future problems.

2 Related Work

Related work on controlling the verifiability of a digital signature includes designated verifier signatures [30, 42], undeniable signatures [4, 8, 10, 11, 14, 23, 19, 31, 33], designated confirmer signatures [9, 6, 18, 28, 34, 37], fair exchanges [1], timed release of signatures [20, 21], and verifiable signature sharing [17]. Despite the considerable amount of work in limiting the verifiability of a digital signature, the conditions that could be incorporated into a digital signature scheme are still very restrictive; the existing protocols merely ensure that only a designated recipient can verify but cannot convince anybody else of the validity of a signature (in designated verifier signatures) and/or collaboration of the signer (in undeniable signatures) or a third party designated by the signer (in designated confirmer signatures, fair exchange) is needed in verifying the signature. Implementing more complex policies or specifying more

varied conditions in these schemes has to resort to appending the condition/policy description inside the message and rely on a third party to enforce them in signature verification and conversion. Hence, there is almost no protection of the privacy of the signer and the recipient with respect to any third party which, if present, is involved in the actual signature conversion and sees the message. In contrast, the only information a third party needs to know in CVS is the condition to be fulfilled.

In a designated verifier signature scheme [30, 42], the validity of a signature could only be verified by those specified by the signer and nobody else. However, there is no means to convert a signature back into an ordinary, publicly verifiable one, thus giving no guarantee to the recipient.

Undeniable signatures, introduced by Chaum [10, 8], are digital signatures which cannot be verified without interacting with the signer. Obviously, an undeniable signature offers almost no guarantee to the recipient as the signer could intentionally become unavailable. Chaum [9] also proposed designated confirmer signatures (as a remedy to undeniable signatures) which, in addition to the signer, can also be verified by interacting with a third party called confirmer who has been designated by the signer. This could in essence be viewed as a signature with limited verifiability. In the original versions of both undeniable and designated confirmer signatures, conversion into ordinary signatures is not possible but subsequent proposals [4, 33, 14, 37, 34] provide this capability. However, the only way to incorporate convertibility conditions is to embed them in the message itself, which is undesirable in the sense of recipient privacy. In many of these schemes, selective conversion is not even allowed; all the issued signatures are converted even though the signer just wants to convert one of them. Although CVS may not yield efficient schemes, roughly speaking, undeniable signatures and designated confirmer signatures could be considered as instantiations of CVS.[3]

Out of the existing schemes, fair exchange of digital signatures [1] has drawn much attention mainly due to its potential application in electronic commerce. In essence, it is an instantiation of a designated confirmer signature which uses the designated confirmer as an arbitrator. However, beside contract signing, the applications of fair exchange are still limited to trading regenerable (digital) goods. When asking the arbitrator to convert a signature, a party needs to show a considerable amount of evidence about the deal or give the digital goods under the custody of the arbitrator. In the latter case, such a requirement may not be achievable in trading non-regenerable items. In the former case, privacy breach (to the arbitrator) is inevitable. Unlike fair exchange, the witnesses in CVS do not act as arbitrators but to verify the fulfillment of a condition. They do not need to know what the deal is or what the signed message is in order to verify the fulfillment of a condition. In fact, the availability of such witnesses is so pervasive in any trading activity and requesting endorsements in the form of a receipt is so natural in the usual workflow.[4] Concurrent signatures [12] are another similar proposal for solving the contract signing problem but CVS cannot give a construction for concurrent signatures.

While covering an important type of verifiability conditions related to time, timed release of signatures are, however, usually implemented by the time-lock puzzle [20, 21] requiring the recipient to go through a series of computation tasks in order to control when he could recover the signature; the main advantage is no third party is needed but it requires intensive computation resources and the only condition specifiable is relative time. More importantly, resuming verifiability of a signature has a rough timing and may not be spontaneous; the guarantee that a signature becomes verifiable after the release time hinges on that the recipient starts the conversion immediately upon receipt of the signature. In fact, CVS could provide a seemingly better solution for this problem, consuming less computation resources and allowing a precise release time specification at the expense of using a passive time server which

[3] Depending on the assumptions on the adversary capabilities, modifications on the security definitions of CVS may be necessary in some cases to give a construction of undeniable/designated confirmer signatures fulfilling its own security definitions.

[4] For example, when sending a parcel, requesting a receipt from the post office is very natural and reasonable.

periodically broadcasts a single signature/endorsement (for all users) on the current time. Spontaneous signature conversion could hence be achieved.

In verifiable signature sharing [17], a signature is divided in such a way that a certain minimum number of parties, each holding a share of the signature, need to pool out their shares in order to recover the signature. When receiving a share, each party could verify its validity. However, it is not trivial to incorporate verifiability conditions in such a scheme and finding such a number of trusted parties in a trading activity is not easy either. Besides, the verifiability of a share also implies that one could link a signature share to its alleged signer even though it is not a complete signature with binding power. As a result, the privacy of the signer as required in scenarios like the future trading example could not be achieved. Although elegant, verifiable signature sharing may not be suitable for the scenarios considered in this paper.

3 Definitions and Security Notions

This section provides a formal definition of conditionally verifiable signatures. After defining the security notions, we discuss the relationships between them.

Notation Convention. For the sake of clarity, we use σ to denote an ordinary signature and δ to denote a CVS partial signature. For example, the ordinary signature of a signer S on a message m would be denoted by $\sigma_S(m)$ and its corresponding partial signature by $\delta_S(m)$. When there is no ambiguity, we might drop the parenthesis and its content. For instance, we may simply write the CVS partial signature as δ or δ_S instead of $\delta_S(m)$.

Suppose $A(PK_S, x)$ is an algorithm with the public key of S and x as input. Provided there is no ambiguity, we may denote it as $A_S(x)$ for short, similarly for the private key case. When comparing the output of two algorithms, we may drop the common input for simpler notations. For example, when comparing $A(w, x, y, z)$ and $B(w, x', y)$, we may simply write $A(x)$ and $B(x')$.

We denote the message space by $\mathcal{M}$, the condition statement space by $\mathcal{C}$, and the set of all possible witness by $\mathcal{W} = \{W_l\}$ and $|\mathcal{W}| = N$. Unless otherwise specified, we assume $\mathcal{M} = \mathcal{C} = \{0,1\}^*$. We further denote the partial signature and ordinary signature spaces by $\mathcal{S}_\delta$ and $\mathcal{S}_\sigma$ respectively.

An instance set of verifiability conditions is of the form $C = \{(c_i, W_i) : c_i \in \mathcal{C}, W_i \in \mathcal{W}\} \subseteq \mathcal{C} \times \mathcal{W}$. Given an instance set of verifiability conditions C, we usually denote the corresponding sets of witness public and private keys by PK_C and sk_C respectively. We also denote the set of witness signatures/endorsements specified in C by σ_C.

Usually, we use $\{W_i\}$ to denote the set containing all W_i's. But by $\{A(x)\}$ we also denote the set of all possible output values of a probabilistic algorithm A when input x, according to its probability distribution.

As usual, we use PPT to denote probabilistic polynomial time algorithm.

As usual, we have the following definition of negligible functions.

Definition 1 [Negligible Functions] *A function $\varepsilon : \mathbb{N} \to \mathbb{R}$ is negligible in λ if and only if $\varepsilon(\lambda) < \frac{1}{poly(\lambda)}$ for some polynomial $poly()$ in λ.*

The players in a conditionally verifiable signature scheme include a signer S, a recipient or verifier V, and a number of witnesses $\{W_i\} \subseteq \mathcal{W}$ (assuming $|\{W_i\}| = L$). A CVS scheme consists of the following algorithms and a confirmation protocol.

Key Generation (CVKGS, CVKGW). Given a security parameter λ, let $\mathsf{CVKGS}(1^\lambda) \rightarrow (PK_S, sk_S)$ and $\mathsf{CVKGW}(1^\lambda) \rightarrow (PK_W, sk_W)$ be two probabilistic algorithms. Then, (PK_S, sk_S) is the public/private key pair for a signer S and (PK_W, sk_W) is the public/private key pair for a witness W.[5]

Signing and Verification (Ordinary Signatures) ($\mathsf{SigS}, \mathsf{VerS}$)/($\mathsf{SigW}, \mathsf{VerW}$). $\mathsf{SigS}(m, sk_S) \rightarrow \sigma_S$ is a probabilistic algorithm generating an ordinary (universally verifiable) signature σ_S of the signer S for a message $m \in \mathcal{M}$. $\mathsf{VerS}(m, \sigma_S, PK_S) \rightarrow \{0, 1\}$ is the corresponding signature verification algorithm, which outputs 1 if σ_S is a true signature of S on message m and outputs 0 otherwise. As usual, for all $(PK_S, sk_S) \in \mathsf{CVKGS}(1^\lambda)$ and all $m \in \mathcal{M}$, we require the following:

$$\mathsf{VerS}(m, \mathsf{SigS}(m, sk_S), PK_S) = 1$$

Similarly, $\mathsf{SigW}(m, sk_W) \rightarrow \sigma_W$ and $\mathsf{VerW}(m, \sigma_W, PK_W) \rightarrow \{0, 1\}$ are the signature generation and verification algorithms of the witness W. Sometimes, we may write SigW as CVEndW to reflect it is actually an endorsement of W.

The signatures generated by these algorithms are publicly verifiable. Note that we use pairing based signatures [3] as witness signatures in our efficient CVS constructions.

Partial Signature Generation (CVSig). Given a set of verifiability conditions $C \subseteq \mathcal{C} \times \mathcal{W}$ and the corresponding set of witness public keys PK_C, $\mathsf{CVSig}(m, C, sk_S, PK_S, PK_C) \rightarrow \delta$ is a probabilistic algorithm for generating the partial signature δ on message $m \in \mathcal{M}$ under the set of verifiability conditions C.

Note that unlike σ, this partial signature δ is not universally verifiable.

Ordinary Signature Extraction (CVExtract). $\mathsf{CVExtract}(m, C, \delta, PK_S, \sigma_C) \rightarrow \sigma / \perp$ is an algorithm which extracts the corresponding ordinary signature σ from a partial signature δ for a message m under the verifiability condition specified by C and a signing public key PK_S when given the set of witness signatures or endorsements σ_C. The extracted signature σ is a universally verifiable one. In case the extraction fails, it outputs $\perp$. Extraction failure could happen when the witness endorsements/signatures used do not match what is required (i.e. a different witness or a different condition statement). *Note that* $\sigma_C = \{\mathsf{SigW}(sk_{W_i}, c_i) : (c_i, W_i) \in C\}$.

CVS Confirmation/Verification. $\mathsf{CVCon}_{(S,V)} = \langle \mathsf{CVConS}, \mathsf{CVConV} \rangle$ is the signature confirmation protocol between the signer and recipient, which could be interactive or non-interactive:

$$\mathsf{CVCon}_{(S,V)}(m, C, \delta) = \langle \mathsf{CVConS}(\sigma, sk_S, r), \mathsf{CVConV}() \rangle (m, C, \delta, PK_S, PK_C) \rightarrow v = \begin{cases} 0 \\ 1 \end{cases}$$

The common input consists of the message m, the set of verifiability conditions C, the partial signature δ, and the public keys of the signer PK_S and the involved witnesses public keys PK_C. The private input of the signer S is σ, sk_S, and r where σ is the corresponding ordinary signature (on the message m) embedded in δ, and r represents the random coins S used in generating δ. The output is either 1 ("true") or 0 ("false"). In essence, this protocol allows the signer S to prove to the recipient V that δ is indeed his partial signature on m, which can be converted back into a publicly verifiable signature σ (i.e. $\mathsf{VerS}(m, \sigma, PK_S) = 1$, once V has obtained all the witness signatures/endorsements on the condition statements as specified in C. Ideally, we want this protocol to be zero-knowledge. Besides, the interactive version is considered in this paper.

[5] In this paper, we may use (PK_i, sk_i) and (PK_{W_i}, sk_{W_i}) interchangeably to denote the public/private key pair of a witness W_i. Provided there is no ambiguity, we prefer to use the former for simpler notations.

The partial signature generation CVSig could be a 1-step or 2-step process. In the latter, an ordinary signature universally verifiable is first generated and a blinding process is then applied to create the CVS partial signature δ. In case CVSig is a 1-step process, the signer should be able to determine the ordinary signature embedded in δ based on his private key and the random coins he used in generating δ.

Ideally, a signature of the witness on the condition statement should be used to retrieve an ordinary signature from its partial signature. But such a requirement is not strict; it should be fine as long as a trapdoor for each condition statement known only to the witness is needed to recover an ordinary signature and finding such a trapdoor without the witness' private information is hard.

3.1 Security Properties of Conditionally Verifiable Signatures

In general, a CVS scheme should satisfy both completeness and perfect convertibility property described below. Completeness ensure that a valid ordinary signature can be retrieved from a valid partial signature. A CVS scheme is perfectly convertible if nobody could distinguish whether a given ordinary signature is extracted from a partial signature or generated directly.

Definition 2 *A CVS scheme is complete* [6] *if for all λ, all $(PK_S, sk_S) \in \{\mathsf{CVKGS}(1^\lambda)\}$, all $(PK_W, sk_W) \in \{\mathsf{CVKGW}(1^\lambda)\}$, all $m \in \mathcal{M}$, all $C \subseteq \mathcal{C} \times \mathcal{W}$, and for all $\delta \in \{\mathsf{CVSig}_S(m, C)\}$, the following holds:*

$$\mathsf{VerS}_S(\mathsf{CVExtract}_S(m, C, \delta)) = 1$$

Definition 3 *A CVS scheme is said to be perfectly convertible if the following ensembles of random variables are computationally indistinguishable (according to Definition 4 discussed later).*

$$\{(PK_S, sk_S) \leftarrow \{\mathsf{CVKGS}(1^\lambda)\}; m \leftarrow \mathcal{M} : \mathsf{SigS}_S(m)\},$$
$$\left\{ \begin{array}{l} (PK_S, sk_S) \leftarrow \{\mathsf{CVKGS}(1^\lambda)\}; \\ (PK_{W_l}, sk_{W_l}) \leftarrow \{\mathsf{CVKGW}(1^\lambda)\}, \forall W_l \in \mathcal{W}; \\ m \leftarrow \mathcal{M}; C \leftarrow 2^{\mathcal{C}\times\mathcal{W}}; \\ \sigma_C \leftarrow \{\{\mathsf{SigW}(c_i, W_i) : (c_i, W_i) \in C\}\} \end{array} : \mathsf{CVExtract}_S(m, C, \mathsf{CVSig}_S(m, C), \sigma_C) \right\}$$

Regarding security, a secure CVS scheme should also satisfy unforgeability, simulatability, cheat-immunity, and zero knowledge confirmation protocol. There are other notions analogous to simulatability, namely, anonymity and invisibility. Before formally defining these security notions, we need to give a basic definition about the indistinguishability between two probability distributions, describe the adversary capability allowed in our security model, and describe the signature and transcript simulators needed for the definitions related to the non-verifiability of a partial signature.

Definition of Indistinguishability

We need the following definition of computational indistinguishability for the discussions in this section.

Definition 4 [Indistinguishability between Random Distributions] *Let $\mathcal{X} = \{X_\lambda\}$ and $\mathcal{Y} = \{Y_\lambda\}$ be two ensembles of random variables over the same sample space for all λ. $\mathcal{X}$ and $\mathcal{Y}$ are computationally*

[6] Note the short form of notations we use here, for example, we use $\mathsf{CVSig}_S(m, C)$ to denote $\mathsf{CVSig}(m, C, sk_S, PK_S, PK_C)$ as PK_C could be uniquely determined by C. But keep in mind, the dropped parameters are still needed in running the algorithm.

indistinguishable (denoted by $\mathcal{X} \cong \mathcal{Y}$) if the following is negligible in λ for all probabilistic polynomial time (PPT) algorithm A:

$$|Pr[x \leftarrow X_\lambda : A(x) = 1] - Pr[y \leftarrow Y_\lambda : A(y) = 1]| \leq \epsilon_{XY}(\lambda)$$

We call ϵ_{XY} the indistinguishability coefficient between $\mathcal{X}$ and $\mathcal{Y}$. This definition in essence means we can transform X_λ into Y_λ and vice-versa by moving a negligible mass of probability distribution. The following lemma would often be useful in showing indistinguishability between distributions.

Lemma 1 *Given three ensembles of random variables, $\mathcal{X} = \{X_\lambda\}$, $\mathcal{Y} = \{Y_\lambda\}$ and $\mathcal{Z} = \{Z_\lambda\}$,*

$$\mathcal{X} \cong \mathcal{Y} \text{ and } \mathcal{Y} \cong \mathcal{Z} \Rightarrow \mathcal{X} \cong \mathcal{Z}$$

The indistinguishability coefficients are related as follows: $\epsilon_{XZ} \leq \epsilon_{XY} + \epsilon_{YZ}$

Proof For any given algorithm A, let $a = Pr[x \leftarrow X_\lambda : A(x) = 1]$, $b = Pr[y \leftarrow Y_\lambda : A(y) = 1]$ and $c = Pr[z \leftarrow Z_\lambda : A(z) = 1]$. Using the well-known triangular inequality, that is, $|a-c| \leq |a-b|+|b-c|$, the relationship between the indistinguishability coefficients follows directly. Given the fact that the sum of two negligible functions is still a negligible function, we can conclude that $\mathcal{X} \cong \mathcal{Z}$. ■

Corollary 2 *Given polynomially many ensembles, $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_N$,*

$$\mathcal{X}_1 \cong \mathcal{X}_2, \mathcal{X}_2 \cong \mathcal{X}_3, \ldots, \mathcal{X}_{N-1} \cong \mathcal{X}_N \Rightarrow \mathcal{X}_1 \cong \mathcal{X}_N$$

Oracle Queries — Allowed Adversary Interaction

In our security model, two types of adversary interaction are allowed:

1. **Signing Oracle $O_S(m, C)$.** For fixed keys PK_S, sk_S, $\{PK_{W_l}\}$, $\{sk_{W_l}\}$, on input a signing query $\langle m, C\rangle$ (where $m \in \mathcal{M}$ and $C = \{(c_i, W_i) : c_i \in \mathcal{C}, W_i \in \mathcal{W}\}$ is a set of verifiability conditions), O_S responds by running CVSig to generate the corresponding partial signature δ. After sending δ to the querying party, O_S runs the confirmation protocol $\mathsf{CVCon}_{(S,V)}$ with the querying party to confirm the validity of δ. *Note that a malicious querying party is allowed to put in any random number in place of δ when running the confirmation protocol.*

2. **Endorsement Oracle $O_E(c, W)$.** For fixed keys $\{PK_{W_l}\}$, $\{sk_{W_l}\}$, on input an endorsement query $\langle c, W\rangle$, Q_E responds by retrieving the needed witness private key sk_W and then running the witness endorsement/signing algorithm SigW (or CVEndW) to create a witness endorsement/signature $\sigma_W(c)$ on the condition statement c.

As we consider adaptive attacks in our model, these oracle queries may be asked adaptively, that is, each query may depend on the replies of the previous queries.

Partial Signature and Confirmation Transcript Simulators

As mentioned earlier, the simulatability property of a CVS scheme is formulated by means of the existence of a publicly known PPT partial signature simulator. Similarly, the zero knowledge property of the confirmation protocol is formulated with a transcript simulator. The simulators used in this paper are as follows.

1. **Partial Signature Simulator:** $\mathsf{Fake}(m, C, PK_S, PK_C) \rightarrow \delta'$

2. **Confirmation Protocol Transcript Simulator:** $\mathsf{FakeT}(m, C, \delta, PK_S, PK_C) \rightarrow \pi'$

On input a set of verifiability conditions $C = \{(c_i, W_i) : c_i \in \mathcal{C}, W_i \in \mathcal{W}\}$, $\mathsf{Fake}_S(m, C)$ outputs a "fake" partial signature of S for a message m under a set of verifiability conditions C. $\mathsf{Fake}_S(m, C)$ is to simulate the output of $\mathsf{CVSig}_S(m, C)$. Similarly, $\mathsf{FakeT}_S(m, C, \delta)$ is to simulate the communication transcript produced by the confirmation protocol $\mathsf{CVCon}_{(S,V)}(m, C, \delta)$ between S and V on input a message m, a set of verifiability conditions C, and a partial signature δ.

3.1.1 Unforgeability

Unforgeability ensures that even all the witnesses pooling out their private keys and given signatures of a signer on messages of their choice should not be able to forge a valid signature on a message not previously queried. The details of unforgeability would be better described by the following game between a challenger and an adversary.

Definition 5 *A CVS scheme is unforgeable against an adaptive chosen message attack if the probability of winning the following game is negligible in the security parameter λ for all PPT adversaries $\mathcal{A}$.*

Setup. The challenger takes a security parameter λ, runs the key generation algorithms for the signer and all witnesses, that is, $(PK_S, sk_S) \leftarrow \{\mathsf{CVKGS}(1^\lambda)\}$ and $(PK_{W_l}, sk_{W_l}) \leftarrow \{\mathsf{CVKGW}(1^\lambda)\}$. The challenger gives the adversary all the public keys, PK_S and $\{PK_{W_l}\}$ and all the witness private keys $\{sk_{W_l}\}$. The challenger keeps the signer's private key sk_S.

Query. The adversary is allowed to make queries to O_S to request a partial signature δ_j for $\langle m_j, C_j \rangle$. Note that the adversary has the witness private keys so no O_E query is necessary.

Guess. The adversary halts and outputs a message-signature pair (m, σ) where $m \neq m_j$ for all j.

Result. The adversary is said to win this game if $\mathsf{VerS}_S(m, \sigma) = 1$.

The winning probability $p_{\mathcal{A}}^{UF}$ is taken over the coin tosses of the key generators, the signer, and the adversary. *Note that the adversary can extract ordinary signatures from any partial signatures as it is given all the witness private keys.*

3.1.2 Simulatability

Simulatability guarantees that nobody, allowed to query other ordinary and partial signatures, can tell whether a given partial signature is genuine or fake. We formulate the simulatability property by means of the existence of a publicly known PPT partial signature simulator Fake which generates a fake partial signature δ_f such that nobody (with bounded computational power) could tell (better than a wild guess) whether $\delta \in \{\mathsf{CVSig}(m, C, sk_S, PK_S, PK_C)\}$ or $\delta \in \{\mathsf{Fake}(m, C, PK_S, PK_C)\}$ for a given δ. This signature is fake as there is negligibly small probability that one could extract a valid ordinary signature from it. The indistinguishability between these two distributions essentially implies that a valid partial signature alone is no more convincing than any random number, namely, nobody could infer who has signed it — the claimed signer (using CVSig) or a forger (using Fake). Detailed formulation of the simulatability property is described by the following game.

Definition 6 *A CVS scheme is simulatable if there exists a PPT simulator* $\mathsf{Fake}(m, C, PK_S, PK_C)$ *(with the same output space as that of* CVSig *for all* λ*) which uses only public information of the signer to simulate a partial signature on any arbitrary message and any set of verifiability conditions such that the advantage of winning the following game is negligible in the security parameter* λ *for all PPT distinguishers/adversaries* $\mathcal{D}$*.*

Setup. The challenger takes a security parameter λ, runs the key generation algorithms for the signer and all witnesses, that is, $(PK_S, sk_S) \leftarrow \{\mathsf{CVKGS}(1^\lambda)\}$ and $(PK_{W_l}, sk_{W_l}) \leftarrow \{\mathsf{CVKGW}(1^\lambda)\}$. The challenger gives the adversary all the public keys, PK_S and $\{PK_{W_l}\}$. The challenger keeps the witness private keys $\{sk_{W_l}\}$ but gives the adversary the signer private key sk_S.[7]

Query 1. The adversary makes queries to obtain the signer's partial signatures and witness endorsement signatures of messages of its choice until it is ready to receive a challenged partial signature. It can make two types of oracle queries:

- Signing Query $\langle m_j, C_j \rangle$ to O_S.
- Endorsement Query $\langle c_j, W_j \rangle$ to O_E.

As the simulator Fake is publicly known, the adversary could freely get a simulator output on any message and conditions of his choice. Since the adversary is given the signer's private key, in addition to O_S queries, it can also generate partial signatures of arbitrary messages and conditions on its own. But even on identical input, these signatures may not be the same as those from the challenger since the random coins used are likely to be different.

Challenge. Once the adversary decides that Query 1 is over, it outputs a message $m \in \mathcal{M}$ and a set of conditions $C \subset \mathcal{C} \times \mathcal{W}$ on which it wishes to be challenged. Let C_E^1 denote the set of all endorsement queries sent to O_E in Query 1. The only constraint is that $C \backslash C_E^1 \neq \phi$ (the empty set).

The challenger flips a coin $b \in \{0, 1\}$ and outputs the following challenge to the adversary:

$$\delta_b = \begin{cases} \mathsf{CVSig}(m, C, sk_S, PK_S, PK_C), & b = 0 \\ \mathsf{Fake}(m, C, PK_S, PK_C), & b = 1 \end{cases}$$

Query 2. The adversary is allowed to run until it outputs a guess. Let C_E^2 be the set of queries that have been made to O_E so far in Query 2. The adversary can issue more (but polynomially many) queries, both signing and endorsement queries, as in Query 1. But for endorsement queries, say with input (c_j, W_j) , the following must hold: $C \backslash (C_E^1 \cup C_E^2 \cup \{(c_j, W_j)\}) \neq \phi$.

Guess. The adversary halts and outputs a guess b' for the hidden coin b.

Result. The adversary is said to win this game if $b' = b$. The advantage of the adversary is defined as:

$$Adv_{\mathcal{D}}^{Sim}(\lambda) = \left| Pr[b' = b] - \frac{1}{2} \right|$$

[7] We actually consider the strongest notion of security

The probability is taken over all the random coins tossed by the key generators, the signer, the witnesses, and the adversary.

Note that after fixing a challenge, the adversary is still allowed to query partial signatures of the challenge in question because CVSig is a probabilistic procedure which would output different partial signatures even when input with the same signing keys, the same message, and the same set of verifiability conditions. Of course, we could only allow polynomially many of these queries. In fact, we are already adopting the strongest notion of security with respect to the privacy of the signer.

In this definition of simulatability, the communication transcript of the confirmation protocol is not given to the adversary. It is a natural question to ask whether the confirmation transcript would help in distinguishing a genuine partial signature from the simulator output. If the indistinguishability property still holds given with the transcript, the CVS scheme in question is said to be non-transferable. We will show in Section 3.1.6, when discussing the security property of the confirmation protocol, that if the confirmation protocol is zero knowledge, simulatability directly implies non-transferability.

The number of input arguments needed for the simulator Fake could lead to different tastes of simulatability, as summarized below. A CVS scheme is:

a. message-independent-simulatable, if Fake does not need the signed message m as input.

b. signer-independent-simulatable, if Fake does not need the identity of the signer PK_S as input.

c. condition-independent-simulatable, if Fake does not need the condition set C or PK_C as input.

d. independently simulatable if Fake just randomly picks an element from the CVS signature space without referencing to the message, the signer's identity or the condition set.

3.1.3 Cheat-immunity

Cheat-immunity guarantees that the recipient of a partial signature cannot retrieve the ordinary signature without collecting all the needed witness signatures. We show later that cheat-immunity could be achieved if a CVS scheme is simulatable and unforgeable and its confirmation protocol is zero knowledge.

Definition 7 *A CVS scheme is cheat-immune (against a chosen message and chosen verifiability condition attack) if the probability of winning the following game is negligible in the security parameter λ for all PPT adversary $\mathcal{A}$.*

Setup. The challenger takes a security parameter λ, runs the key generation algorithms for the signer and all witnesses, that is, $(PK_S, sk_S) \leftarrow \{\mathsf{CVKGS}(1^\lambda)\}$ and $(PK_{W_l}, sk_{W_l}) \leftarrow \{\mathsf{CVKGW}(1^\lambda)\}$. The challenger gives the adversary all the public keys, PK_S and $\{PK_{W_l}\}$. The challenger keeps all the private keys sk_S and $\{sk_{W_l}\}$.

Query 1. The adversary makes queries to obtain the signer's partial signatures and witness endorsement signatures on messages of its choice until it is ready to receive a challenged partial signature. It can make two types of queries:

- Signing Query $\langle m_j, C_j \rangle$ to O_S.
- Endorsement Query $\langle c_j, W_j \rangle$ to O_E.

With these two types of queries, the adversary can obtain any ordinary signatures of the signer on messages of his choice.

Challenge. Once the adversary decides that Query 1 is over, it outputs a message $m \in \mathcal{M}$ and a set of conditions $C \subset \mathcal{C} \times \mathcal{W}$ on which it wishes to be challenged. Let C_E^1 denote the set of all endorsement queries sent to O_E in Query 1. The only constraint is that $C \backslash C_E^1 \neq \phi$ (the empty set) and $m \neq m_j, \forall j$.

The challenger uses CVSig to generate a partial signature δ on message m under condition C. It sends δ as the challenge to the adversary and runs the confirmation protocol $\mathsf{CVCon}_{(S,V)}$ with it.

Query 2. The adversary is allowed to run until it outputs a guess. Let C_E^2 be the set of queries that have been made to O_E so far in Query 2. The adversary can issue more queries, both signing and endorsement queries, as in Query 1. But for signing queries, $m_j \neq m$, and for endorsement queries, say with input (c_j, W_j), the following must hold: $C \backslash (C_E^1 \cup C_E^2 \cup \{(c_j, W_j)\}) \neq \phi$.

Guess. The adversary halts and outputs an ordinary signature σ for message m.

Result. The adversary is said to win this game if $\mathsf{VerS}_S(m, \sigma) = 1$.

The winning probability $p_{\mathcal{A}}^{CI}$ is taken over the coin tosses of the key generators, the signer, the witnesses, and the adversary.

Note that a less restrictive adversary model could also be considered. In that model, the adversary is allowed to pose a challenge message which has been previously queried but no ordinary signature on it has been extracted so far using the results of all the endorsement queries made previously. However, a tight reduction between unforgeability and cheat-immunity in this model with relaxed adversary restriction would not be possible. Consequently, a slightly more restrictive model is considered in this paper for the sake of tight reduction. In the real case, this restriction simply means that a signer should not issue signatures to the same party on exactly the same message but with different verifiability conditions. This could easily be achieved by padding a message with data like serial number and, in fact, we believe this is a reasonable assumption in real practice. Besides, we also believe that a CVS scheme achieving the cheat-immunity property in the model considered in this paper would enjoy the same property in a slightly relaxed model in which the signer could issue polynomially many signatures on the same message but with different verifiability conditions to the same party.

3.1.4 Message Invisibility

As mentioned earlier, beside simulatability, there are two other possible formulations of the non-verifiability property of a partial signature — invisibility and anonymity. These definitions are variants of those in [34, 6, 18]. In general, the notions of invisibility and anonymity are not exactly equivalent to simulatability even though they are implied by simulatability in many cases.

Message invisibility ensures that given two messages and the partial signature of one of them, together with the associated verifiability conditions, nobody could tell which one of the messages the partial signature belongs to. The rationale behind this notion is that, in the worst case, even though the recipient can show to others who has signed a partial signature and under what conditions it would become verifiable, nevertheless, there is doubt about whether it is valid for the alleged message as the invisibility property guarantees that nobody (with bounded computational power) could link a message to its partial signature. In other words, although everyone knows the alleged signer has really signed a

given partial signature, nobody could assure that it is not an old one on a different message being abused by somebody. As a result, the signer is not bound to the alleged message and his privacy is protected. The details of the invisibility property are as follows.

Definition 8 *A CVS scheme is invisible (or message-invisible) if the advantage of winning the following game is negligible in the security parameter λ for all PPT distinguishers/adversaries $\mathcal{D}$.*

Setup. The challenger takes a security parameter λ, runs the key generation algorithms for the signer and all witnesses, that is, $(PK_S, sk_S) \leftarrow \{\mathsf{CVKGS}(1^\lambda)\}$ and $(PK_{W_l}, sk_{W_l}) \leftarrow \{\mathsf{CVKGW}(1^\lambda)\}$. The challenger gives the adversary all the public keys, PK_S and $\{PK_{W_l}\}$. The challenger keeps the witness private keys $\{sk_{W_l}\}$ but gives the adversary the signer private key sk_S.

Query 1. The adversary makes queries to obtain the signer's partial signatures and witness signatures of messages of his choice until it is ready to receive a challenged partial signature. It can make two types of queries:

- Signing Query $\langle m_j, C_j \rangle$ to O_S.
- Endorsement Query $\langle c_j, W_j \rangle$ to O_E.

Challenge. Once the adversary decides that Query 1 is over, it outputs two equal length messages $M_0, M_1 \in \mathcal{M}$ and a set of conditions $C \subset \mathcal{C} \times \mathcal{W}$ on which it wishes to be challenged. Let C_E^1 denote the set of all endorsement queries sent to O_E in Query 1. The only constraint is that $C \backslash C_E^1 \neq \phi$ (the empty set).

The challenger flips a coin $b \in \{0, 1\}$ and outputs the following challenge to the adversary:

$$\delta_b = \mathsf{CVSig}(M_b, C, sk_S, PK_S, PK_C)$$

Query 2. The adversary is allowed to run until it outputs a guess. Let C_E^2 be the set of queries that have been made to O_E so far in Query 2. The adversary can issue more (but polynomially many) queries, both signing and endorsement queries, as in Query 1. But for endorsement queries, say with input (c_j, W_j) , the following must hold: $C \backslash (C_E^1 \cup C_E^2 \cup \{(c_j, W_j)\}) \neq \phi$.

Guess. The adversary halts and outputs a guess b' for the hidden coin b.

Result. The adversary is said to win this game if $b' = b$. The advantage of the adversary is defined as:

$$Adv_{\mathcal{D}}^{Inv}(\lambda) = \left| Pr[b' = b] - \frac{1}{2} \right|$$

The probability is taken over all the random coins tossed by the key generators, the signer, the witnesses, and the adversary.

3.1.5 Signer Anonymity

Anonymity ensures that given two possible signers, a message, and a set of verifiability conditions, together with a partial signature on the message and conditions from one of the signers, nobody could tell which one of the signers has actually created the partial signature. That is, nobody could link a partial signature to its signer. Similar to invisibility, the rationale behind the notion of anonymity is that,

in the worst case, even the recipient can show to others which message a partial signature is signed for and under what conditions it would become verifiable, nobody could tell whether the partial signature in question was created by the alleged signer or the recipient himself. Hence, the signer's privacy is protected. In general, anonymity could provide seemingly better privacy protection than invisibility.

Definition 9 *A CVS scheme is anonymous (or signer-anonymous) if the advantage of winning the following game is negligible in the security parameter λ for all PPT distinguishers/adversaries $\mathcal{D}$.*

Setup. The challenger takes a security parameter λ, runs the key generation algorithms for two signers ($S0$ and $S1$) and all witnesses, that is, $(PK_{S0}, sk_{S0}) \leftarrow \{\mathsf{CVKGS}(1^\lambda)\}$, $(PK_{S1}, sk_{S1}) \leftarrow \{\mathsf{CVKGS}(1^\lambda)\}$, and $(PK_{W_l}, sk_{W_l}) \leftarrow \{\mathsf{CVKGW}(1^\lambda)\}$. The challenger gives the adversary all the public keys, PK_{S0}, PK_{S1}, and $\{PK_{W_l}\}$. The challenger keeps the witness private keys $\{sk_{W_l}\}$ but gives the adversary the two signer private keys sk_{S0} and sk_{S1}.

Query 1. The adversary makes queries to obtain the signer's partial signatures and witness signatures of messages of his choice until it is ready to receive a challenged partial signature. It can make two types of queries:

- Signing Query $\langle s_j, m_j, C_j \rangle$ (with $s_j \in \{0, 1\}$) to O_S where $s_j = 0/1$ represents requesting a partial signature from $S0/S1$.
- Endorsement Query $\langle c_j, W_j \rangle$ to O_E.

Challenge. Once the adversary decides that Query 1 is over it outputs a message $m \in \mathcal{M}$ and a set of conditions $C \subset \mathcal{C} \times \mathcal{W}$ on which it wishes to be challenged. Let C_E^1 denote the set of all endorsement queries sent to O_E in Query 1. The only constraint is that $C \backslash C_E^1 \neq \phi$ (the empty set).

The challenger flips a coin $b \in \{0, 1\}$ and outputs the following challenge to the adversary:

$$\delta_b = \mathsf{CVSig}(m, C, sk_{Sb}, PK_{Sb}, PK_C)$$

Query 2. The adversary is allowed to run until it outputs a guess. Let C_E^2 be the set of queries that have been made to O_E so far in Query 2. The adversary can issue more (but polynomially many) queries, both signing and endorsement queries, as in Query 1. But for endorsement queries, say with input (c_j, W_j) , the following must hold: $C \backslash (C_E^1 \cup C_E^2 \cup \{(c_j, W_j)\}) \neq \phi$.

Guess. The adversary halts and outputs a guess b' for the hidden coin b.

Result. The adversary is said to win this game if $b' = b$. The advantage of the adversary is defined as:

$$Adv_{\mathcal{D}}^{Ano}(\lambda) = \left| Pr[b' = b] - \frac{1}{2} \right|$$

The probability is taken over all the random coins tossed by the key generators, the signers, the witnesses, and the adversary.

As mentioned before, both invisibility and anonymity could limit the accountability of the signer of a partial signature to a more or less degree, thus protecting his privacy. Overall, we believe simulatability and anonymity provide seemingly better protection of signer privacy. The reason is if a CVS scheme

is invisible, one may be able to link a genuine partial signature to its signer (but could not assure if it is for the alleged message), whereas, if a CVS scheme is simulatable or anonymous, a genuine partial signature is indistinguishable from a fake one (possibly generated by the signature holder). Similar to simulatability, provided that the confirmation protocol is zero-knowledge, the transcript would not provide any additional information useful for breaking the invisibility and anonymity properties. This will be discussed in Section 3.1.6.

Note that we do not formulate any security notion related to protecting the indistinguishability of the verifiability conditions alone as there seems to have no point to make such a formulation if someone can determine who has signed on which message given a partial signature. Even though the partial signature looks different from an ordinary signature (and possibly have no legal binding power), if the identity of the signer and his signed message are revealed, it is meaningless to ensure the privacy of the verifiability conditions. We think this is hardly better than embedding in a signed message the verifiability conditions and signing it with an ordinary signature. Certainly, we need to distinguish between this case of just hiding the verifiability conditions (but leaving the signer identity and the message disclosed) and the case of hiding all information including the signer identity, the message and the verifiability conditions. The latter is the most desired property of a partial signature which provides the highest possible level of signer privacy but could be difficult to achieve.

3.1.6 Properties of the Confirmation Protocol

In the definition of the confirmation protocol, we do not impose any restriction on whether it should be interactive or non-interactive. But the interactive version is discussed in the paper. Like any protocol of zero-knowledge proof, completeness, soundness and zero knowledge are the required properties of the confirmation protocol $\mathsf{CVCon}_{(S,V)}$. Recall that if a given partial signature is valid for the given message and verifiability conditions, $\mathsf{CVCon}_{(S,V)}$ returns 1 and, otherwise, 0. The definitions are stated as follows.

Definition 10 Completeness. *For all $(PK_S, sk_S) \in \{\mathsf{CVKGS}(1^\lambda)\}$, all $(PK_W, sk_W) \in \{\mathsf{CVKGW}(1^\lambda)\}$, all $m \in \mathcal{M}$, all $C \subseteq \mathcal{C} \times \mathcal{W}$, all $\sigma_C \in \{\{\mathsf{CVEndW}(c_i, W_i) : (c_i, W_i) \in C\}\}$, and all $\delta \in \{0,1\}^*$, if $\mathsf{VerS}_S(m, \mathsf{CVExtract}_S(m, C, \delta, \sigma_C)) = 1$ (i.e. the extracted ordinary signature is valid), then*

$$Pr[\mathsf{CVCon}_{(S,V)}(m, C, \delta) = 0] < \varepsilon(\lambda)$$

where $\varepsilon(\lambda)$ is a negligible function in the security parameter λ.

Definition 11 Soundness. *Using the same random experiment as in the definition of completeness, if $\mathsf{VerS}_S(m, \mathsf{CVExtract}_S(m, C, \delta, \sigma_C)) = 0$ (i.e. the extracted ordinary signature is invalid), then*

$$Pr[\mathsf{CVCon}_{(S,V)}(m, C, \delta) = 1] < \varepsilon(\lambda)$$

where $\varepsilon(\lambda)$ is a negligible function in the security parameter λ.

Definition 12 Zero-knowledge. *Suppose the same random experiment as in the definition of completeness has been set up. For a given input instance $x = \langle m, C, \delta, PK_S, PK_C \rangle$ to the confirmation protocol $\mathsf{VCCon}_{(S,V)}(x) = \langle \mathsf{CVConS}(r_S), \mathsf{CVConV}^*() \rangle(x)$, let $\pi^{\mathsf{VCCon}}_{(S,V)}(x)$ denote the resulting communication transcript produced by the confirmation protocol run between the prover S (with private input r_S) and the verifier V (which may deviate from the specified protocol). $\mathsf{VCCon}_{(S,V)}$ is zero-knowledge if there exists a PPT simulator $\mathsf{SimT}(x)$ which could produce a simulated transcript $\pi^{\mathsf{SimT}}(x)$ without using the private input of the prover S in such a way that the distributions $\{\pi^{\mathsf{VCCon}}_{(S,V)}(x)\}$ and $\{\pi^{\mathsf{SimT}}(x)\}$ are computationally indistinguishable in terms of the security parameter λ.*

Note here we use the alternative definition of zero knowledge instead of the standard one which states that anything computable by a malicious verifier through interaction with the prover can be computed on his own without interacting with the prover. As mentioned earlier, the need of the zero knowledge property is to ensure the non-transferability of a partial signature along with the transcript of its confirmation protocol run, which is stated below.

Definition 13 Non-transferability. *Given a CVS scheme simulatable with respect to a PPT fake partial signature simulator* Fake *(according to Definition 6), let* $\delta_t = \mathsf{CVSig}_S(m, C)$ *and* $\delta_f = \mathsf{Fake}_S(m, C)$ *be the true and fake partial signatures of a signer* S *on a message* m *with a verifiability condition set* C*. Let* $\pi^{\mathsf{CVCon}}_{(S,V)}(m, C, \delta_t)$ *denote the communication transcript of the confirmation protocol run on* δ_t*. Suppose there exists a PPT transcript simulator* FakeT *taking* δ_f *as input to generate a simulated transcript* $\pi^{\mathsf{FakeT}}(m, C, \delta_f)$ *as if* δ_f *is a valid partial signature. The CVS scheme is non-transferable if the following distributions are computationally indistinguishable in terms of the security parameter* λ*:*

$$\{\mathsf{CVSig}_S(m, C),\ \pi^{\mathsf{CVCon}}_{S,V}(m, C, \mathsf{CVSig}_S(m, C))\},\ \{\mathsf{Fake}_S(m, C),\ \pi^{\mathsf{FakeT}}(m, C, \mathsf{Fake}_S(m, C))\}$$

As mentioned earlier, the non-transferability property is to ensure that the communication transcript of carrying the confirmation protocol on a genuine partial signature would not provide information non-negligibly help to determine the validity of the partial signature. The idea of formulating non-transferability using the transcript simulator is if anyone could use public information of the signer to generate a fake partial signature (from Fake) and a fake transcript for it (from FakeT) so that they are indistinguishable from a genuine partial signature and its genuine transcript, then a genuine partial signature along with its transcript is unlinkable to its signer. Such indistinguishability is possible because no interaction between the prover and the verifier is needed, and only a simulated transcript is produced.

We do not incorporate the adaptive attack model in the non-transferability definition as in the definition of simulatability, but the modification should be straightforward, which could be done by simply adding a genuine/simulated transcript to a genuine/fake partial signature in the challenge phase of the simulatability game in Definition 6. In fact, we could show that ensuring a CVS scheme satisfying simulatability in an adaptive attack model together with a zero-knowledge confirmation protocol for it would guarantee its non-transferability in the same adaptive attack model.

Just like the simulatability property whose fulfillment hinges on the existence of a PPT simulator Fake, the fulfillment of the non-transferability property depends on the existence of a PPT transcript simulator FakeT. If we recall that in the zero knowledge definition (Definition 12), a zero knowledge confirmation protocol implies the existence of a PPT transcript simulator SimT which, on input a partial signature δ_t, outputs a transcript indistinguishable from a true one recorded during a run of the confirmation protocol on δ_t, one may be tempted to use SimT as an implementation for FakeT. At first glance, it looks fine. However, the indistinguishability between the real transcript and the simulated transcript in any zero-knowledge proof is based on the assumption that they come up from the same input and the claim to prove is true. If we use SimT to implement FakeT, the input to the simulator is no longer a genuine partial signature, thus violating this basic assumption. A detailed explanation is as follows.

The transcript simulator SimT of any zero knowledge proof is usually implemented by emulating the conversation between a prover and a verifier. In each round of iteration, even though a claim to prove is false, a malicious prover (without any knowledge of the needed private information) could always answer a fraction of all possible challenge questions; the correctness of a claim in any (interactive) zero knowledge proof is assured through actual interaction between the prover and the signer. Hence,

the "rewinding" technique[8] is commonly adopted to simulate a transcript. By replacing the input to SimT with a fake signature, one could still produce a transcript that appears to be valid and passes the verification test of $\mathsf{CVCon}_{(S,V)}$ in all iterations. That looks fine at first glance, but the distributions of the two transcripts $\{\pi^{\mathsf{CVCon}}_{(S,V)}(m, C, \delta_t)\}$ and $\{\pi^{\mathsf{SimT}}(m, C, \delta_f)\}$ (where $\delta_t = \mathsf{CVSig}_S(m, C)$ and $\delta_f = \mathsf{Fake}_S(m, C)$) are not necessarily indistinguishable even though the confirmation protocol is perfectly zero knowledge with respect to SimT. In fact, to pass the verification test, the output space of SimT fed with an invalid partial signature is likely to be different than that fed with a valid partial signature, that is, the following two distributions could differ considerably: $\{\pi^{\mathsf{SimT}}(m, C, \delta_t)\}$ and $\{\pi^{\mathsf{SimT}}(m, C, \delta_f)\}$.

Although the zero knowledge property of the confirmation protocol $\mathsf{CVCon}_{(S,V)}$ with respect to SimT alone is not sufficient to ensure non-transferability, we can still show that, for any CVS scheme, if $\{\mathsf{CVSig}_S(m, C)\} \cong \{\mathsf{Fake}_S(m, C)\}$, then the zero-knowledge property of the confirmation protocol $\mathsf{CVCon}_{(S,V)}$ implies the non-transferability property (not in an adaptive attack model) and we could use SimT as FakeT. Before we prove the theorem, we need the following lemma.

Lemma 3 *Given two ensembles of distribution $\{X_\lambda\}$ and $\{Y_\lambda\}$, which have the same sample space for all λ, and a PPT algorithm T_λ (a transcript simulator) whose input space is the same as that of X_λ and Y_λ, let $\pi(x)$ denote the output of T_λ on input x.*[9] *If $\{X_\lambda\} \cong \{Y_\lambda\}$ in the security parameter λ, then*

$$\{x \leftarrow X_\lambda; \pi(x) \leftarrow \{T_\lambda(x)\} : (x, \pi(x))\} \cong \{y \leftarrow Y_\lambda; \pi(y) \leftarrow \{T_\lambda(y)\} : (y, \pi(y))\}$$

Proof The reduction is straightforward. For completeness, a proof is given in Appendix A. ■

Theorem 4 *For any CVS scheme, if there exists a fake partial signature simulator Fake such that $\{\mathsf{CVSig}_S(m, C)\} \cong \{\mathsf{Fake}_S(m, C)\}$ for all S, m, C and its confirmation protocol is zero knowledge with respect to a transcript simulator SimT, then SimT could be used as a transcript simulator FakeT for the fake partial signature Fake so that the following two distributions are indistinguishable for all S, m, C (i.e. non-transferable not in an adaptive attack model):*

$$\{\mathsf{CVSig}_S(m, C),\ \pi^{\mathsf{CVCon}}_{S,V}(m, C, \mathsf{CVSig}_S(m, C))\},\ \{\mathsf{Fake}_S(m, C),\ \pi^{\mathsf{FakeT}}(m, C, \mathsf{Fake}_S(m, C))\}$$

where $\pi^{\mathsf{CVCon}}_{S,V}(\cdot)$ and $\pi^{\mathsf{FakeT}}(\cdot)$ are the transcript outputs of a real confirmation protocol run and FakeT respectively.

Proof Let $\delta_t = \mathsf{CVSig}_S(m, C)$ and $\delta_f = \mathsf{Fake}_S(m, C)$,[10] then the following two distributions are indistinguishable: $\{\delta_t\}$ and $\{\delta_f\}$. Let $\pi^{\mathsf{CVCon}}_{S,V}$, π^{SimT}, and π^{FakeT} denote the transcript outputs of a real confirmation protocol run, SimT, and FakeT respectively. Then using Lemma 3,

$$\{(\delta_t, \pi^{\mathsf{SimT}}(\delta_t))\} \cong \{(\delta_f, \pi^{\mathsf{SimT}}(\delta_f))\} \Leftrightarrow \{(\delta_t, \pi^{\mathsf{SimT}}(\delta_t))\} \cong \{(\delta_f, \pi^{\mathsf{FakeT}}(\delta_f))\} \qquad \forall S, m, C$$

The zero-knowledge property of $\mathsf{CVCon}_{(S,V)}$ with respect to SimT ensures the following:

$$\{(\delta_t, \pi^{\mathsf{CVCon}}_{S,V}(\delta_t))\} \cong \{(\delta_t, \pi^{\mathsf{SimT}}(\delta_t))\}, \qquad \forall S, m, C$$

[8] In the rewinding technique, the simulator emulates a version of the zero-knowledge protocol between a prover and a verifier. In each round of iteration, it prepares the answer to a randomly picked challenge question beforehand, and runs the verifier algorithm to generate a challenge. When the challenge turns out to be what it has prepared, it just returns the prepared answer, whereas, if asked of a different challenge, it resets the verifier to go back to the start of the current iteration and prepares for another challenge.

[9] Note that T_λ is probabilistic, so even for the same input x, $T_\lambda(x)$ may be different between two evaluations.

[10] Since CVSig and Fake are probabilistic, both δ_t and δ_f are random variables.

By Lemma 1, the following two distributions are indistinguishable:

$$\{(\delta_t, \pi_{S,V}^{\mathsf{CVCon}}(\delta_t))\} \cong \{(\delta_f, \pi^{\mathsf{FakeT}}(\delta_f))\}, \qquad \forall S, m, C$$

■

Note the above theorem does not incorporate an adaptive attack model which is considered in the following theorem.

Theorem 5 *Given that a CVS scheme is simulatable with respect to a PPT fake partial signature simulator* Fake*, if its confirmation protocol* $\mathsf{CVCon}_{(S,V)}$ *is zero knowledge with respect to a PPT transcript simulator* SimT*, then it is non-transferable in the same adaptive attack model as in the simulatability definition and* SimT *could be used as the transcript simulator* FakeT *for the fake partial signature* Fake*. In other words, the following two distributions are indistinguishable for all* S, m, C *in an adaptive attack model:*

$$\{\mathsf{CVSig}_S(m, C),\ \pi_{S,V}^{\mathsf{CVCon}}(m, C, \mathsf{CVSig}_S(m, C))\},\ \{\mathsf{Fake}_S(m, C),\ \pi^{\mathsf{FakeT}}(m, C, \mathsf{Fake}_S(m, C))\}$$

where $\pi_{S,V}^{\mathsf{CVCon}}(\cdot)$ *and* $\pi^{\mathsf{FakeT}}(\cdot)$ *are transcript outputs of a real confirmation protocol run and* FakeT *respectively.*

Proof We prove by contradiction. We assume that the CVS scheme is simulatable with respect to a PPT simulator Fake, that is, the advantage $Adv_{\mathcal{D}}^{Sim}$ for breaking the simulatability with respect to Fake is negligible for all PPT adversaries $\mathcal{D}$. Assume we use the transcript simulator SimT of the zero knowledge proof for the confirmation protocol as the transcript simulator FakeT for the fake signature. Suppose there is a PPT distinguisher D which could break the non-transferability property with respect to Fake and FakeT with non-negligible advantage Adv_D^{NT}, then we can construct D' to break the simulatability property as follows:

$D'(\delta_b)$ where δ_b is a genuine/fake partial signature when $b = 0/1$
Setup.
Ask its challenger for the public keys of the signer and the witnesses
Run D on the same set of public keys.
Get the signer's private key from its challenger and pass it to D.
Query.
Answer all the signing queries itself.
Pass all the endorsement queries from D to its oracle and relay the results back to D.
Challenge.
D outputs (m, C) it wish to be challenged.
Output (m, C) as its challenge request and receive a challenge δ_b.
Compute $\pi^{\mathsf{SimT}}(\delta_b)$ and pass $(\delta_b, \pi^{\mathsf{SimT}}(\delta_b))$ as a challenge to D.
Guess.
D outputs b' as a guess for b. Output b'.

The query responses are perfectly simulated; the view of D in the simulated environment is identical to its view in a real attack. Let $\delta_t = \mathsf{CVSig}_S(m, C)$ and $\delta_f = \mathsf{Fake}_S(m, C)$. When $b = 1$, the challenge is a fake partial signature δ_f and $\pi^{\mathsf{SimT}}(\delta_b) = \pi^{\mathsf{FakeT}}(\delta_f)$, and the input to D is $(\delta_f, \pi^{\mathsf{FakeT}}(\delta_f))$. Whereas, when $b = 0$, the challenge is a true partial signature δ_t and $\pi^{\mathsf{SimT}}(\delta_b) = \pi^{\mathsf{SimT}}(\delta_t)$, and

the input to D is $(\delta_t, \pi^{\mathsf{SimT}}(\delta_t))$. Due to the zero knowledge property of the confirmation protocol, $(\delta_t, \pi^{\mathsf{SimT}}(\delta_t))$ could perfectly simulate $(\delta_t, \pi^{\mathsf{CVCon}}_{S,V}(\delta_t))$, a valid challenge to D. As a result, the challenge to D is perfectly simulated no matter $b = 0$ or $b = 1$. It could be seen that $Adv^{Sim}_{D'} = Adv^{NT}_{D}$ which is non-negligible if D can break the non-transferability property. This concludes the reduction.

Instead of stating the zero knowledge property informally as above, a more rigorous treatment is possible by evaluating the probability of success of D and D' respectively.

The probability of success of D' with respect to the simulatability game is given by:

$$\begin{aligned} Pr^{Sim}_{D'}[Success] &= \tfrac{1}{2}\left(Pr[D'(\delta_t) = 0] + Pr[D'(\delta_f) = 1]\right) \\ &= \tfrac{1}{2}(Pr[D(\delta_t, \pi^{\mathsf{SimT}}(\delta_t)) = 0] + Pr[D(\delta_f, \pi^{\mathsf{FakeT}}(\delta_f)) = 1]) \end{aligned}$$

The probability of success of D with respect to the non-transferability game is given by:

$$\begin{aligned} Pr^{NT}_{D}[Success] &= \tfrac{1}{2}(Pr[D(\delta_t, \pi^{\mathsf{CVCon}}_{S,V}(\delta_t)) = 0] + Pr[D(\delta_f, \pi^{\mathsf{FakeT}}(\delta_f)) = 1]) \\ &= \tfrac{1}{2}(Pr[D(\delta_t, \pi^{\mathsf{CVCon}}_{S,V}(\delta_t)) = 0] - Pr[D(\delta_t, \pi^{\mathsf{SimT}}(\delta_t)) = 0] \\ &\quad + Pr[D(\delta_t, \pi^{\mathsf{SimT}}(\delta_t)) = 0] + Pr[D(\delta_f, \pi^{\mathsf{FakeT}}(\delta_f)) = 1]) \\ &= Pr^{Sim}_{D'}[Success] + \tfrac{1}{2}(Pr[D(\delta_t, \pi^{\mathsf{CVCon}}_{S,V}(\delta_t)) = 0] - Pr[D(\delta_t, \pi^{\mathsf{SimT}}(\delta_t)) = 0]) \end{aligned}$$

Taking absolute values on both sides,

$$\begin{aligned} Adv^{NT}_{D} &\leq Adv^{Sim}_{D'} + \tfrac{1}{2}|Pr[D(\delta_t, \pi^{\mathsf{CVCon}}_{S,V}(\delta_t)) = 0] - Pr[D(\delta_t, \pi^{\mathsf{SimT}}(\delta_t)) = 0]| \\ &= Adv^{Sim}_{D'} + \tfrac{1}{2}|Pr[D(\pi^{\mathsf{CVCon}}_{S,V}(\delta_t)) = 1] - Pr[D(\pi^{\mathsf{SimT}}(\delta_t)) = 1]| \end{aligned}$$

Due to the zero knowledge property, that is, $\{\pi^{\mathsf{CVCon}}_{S,V}(\delta_t)\} \cong \{\pi^{\mathsf{SimT}}(\delta_t)\}$, which actually means that $|Pr[\mathcal{D}(\pi^{\mathsf{CVCon}}_{S,V}(\delta_t)) = 1] - Pr[\mathcal{D}(\pi^{\mathsf{SimT}}(\delta_t)) = 1]|$ is negligible in the security parameter λ for all PPT $\mathcal{D}$. As a result, $Adv^{NT}_{D} \leq Adv^{Sim}_{D'}$ up to a negligible term (in λ). If Adv^{NT}_{D} is non-negligible, then $Adv^{Sim}_{D'}$ must also be non-negligible, which is a contradiction as we assume $Adv^{Sim}_{\mathcal{D}}$ is negligible in λ for all PPT $\mathcal{D}$ (the simulatability property). In other words, simulatability implies non-transferability if the confirmation protocol is zero knowledge. ■

Using similar argument, we could arrive at the following two theorems about invisibility and anonymity.

Theorem 6 *Given that a CVS scheme is invisible, if its confirmation protocol $\mathsf{CVCon}_{(S,V)}$ is zero knowledge with respect to a PPT transcript simulator SimT, then the confirmation transcript $\pi^{\mathsf{CVCon}}_{S,V}$ does not leak out information for breaking the invisibility property and the CVS scheme remains invisible in the following sense in the same adaptive attack model:*

$$\{\mathsf{CVSig}_S(m_0, C), \pi^{\mathsf{VCCon}}_{S,V}(\mathsf{CVSig}_S(m_0, C))\} \cong \{\mathsf{CVSig}_S(m_1, C), \pi^{\mathsf{VCCon}}_{S,V}(\mathsf{CVSig}_S(m_1, C))\}$$

for all signer S, messages m_0, m_1 and condition C.

Theorem 7 *Given that a CVS scheme is anonymous, if its confirmation protocol $\mathsf{CVCon}_{(S,V)}$ is zero knowledge with respect to a PPT transcript simulator SimT, then the confirmation transcript $\pi^{\mathsf{CVCon}}_{S,V}$ does not leak out information for breaking the anonymity property and the CVS scheme remains anonymoys in the following sense in the same adaptive attack model:*

$$\{\mathsf{CVSig}_{S0}(m, C), \pi^{\mathsf{VCCon}}_{(S,V)}(\mathsf{CVSig}_{S0}(m, C))\} \cong \{\mathsf{CVSig}_{S1}(m, C), \pi^{\mathsf{VCCon}}_{(S,V)}(\mathsf{CVSig}_{S1}(m, C))\}$$

for all signers $S0$, $S1$, message m and condition C.

The practical significance of Theorems 5 - 7 is that we could separate the designs of the CVS signing algorithm from that of the confirmation protocol, thus breaking down a more complex problem into two simpler ones. It also set out the sufficient conditions under which the transcript of a given confirmation protocol would not leak out knowledge to help in breaking the underlying security properties, be it simulatability, invisibility or anonymity. In fact, in our efficient constructions in Section 6, we separate the design into two parts — a blinding mechanism to hide an ordinary signature and a confirmation protocol.

3.2 Required Properties of a Secure CVS Scheme

As discussed before, a CVS scheme must balance the protection between the interests of the signer and the recipient. To protect the signer's privacy, the verifiability of his partial signatures must be limited before all his specified conditions are fulfilled. Hence, a secure CVS scheme should be unforgeable, non-transferable, and cheat-immune. On the other hand, to protect the interest of the recipient, a CVS scheme should provide an assurance that the signer's partial signature could be validated after all the conditions are fulfilled, which is guaranteed by the completeness of the CVS scheme and the soundness of its confirmation protocol. Besides, a good CVS scheme should also protect the privacy of the recipient in the sense that nobody would be able to see from a recovered ordinary signature what the recipient has done to validate the signature. This could be achieved by the perfect convertibility property. If a CVS scheme is perfectly convertible, an ordinary signature extracted from a partial signature is indistinguishable from a usual ordinary signature generated by the signer directly.

To summarize, a desired CVS scheme should be unforgeable, non-transferable, cheat-immune, complete, and perfectly convertible, and its confirmation protocol should be sound. As shown previously, the non-transferability property of a CVS scheme could be achieved if it is simulatable and its confirmation protocol is zero knowledge. Since working with the latter two properties has the advantage of design separability, we would prefer to use the following set of equivalent requirements on a CVS scheme: unforgeability, simulatability, cheat-immunity, completeness, and perfectly convertibility, and a zero knowledge confirmation protocol.

To further distill down this set of required properties into a smaller set, we will discuss the implications between some of these security notions in the next section. One of the main results is that cheat-immunity is implied by the unforgeability and simulatability properties. Regarding the simulatability property, there are two other similar but not exactly equivalent notions, namely, invisibility and anonymity. Although we believe simulatability is more pertinent in modeling the desired non-verifiability property, it is nice to see under what conditions simulatability implies invisibility and anonymity. We show in the next section that simulatability implies invisibility if the simulator is message-independent and anonymity if the simulator is signer-independent.

To prove that a CVS scheme satisfies all the desired properties, we only need to prove that it is unforgeable and simulatable with respect to a PPT simulator, and its confirmation protocol is zero-knowledge. This also leads to a natural paradigm for designing secure CVS schemes. More concretely, we could first choose an unforgeable ordinary signature scheme, and then construct a blinding mechanism which could make an ordinary signature covert in a partial signature in such a way that there exists a public, PPT fake signature simulator whose output is indistinguishable from the partial signature. (Of course, we may need to move back reconsidering the hiding mechanism while searching for an efficient simulator but this paradigm already provides a systematic way for designing CVS schemes). Finally, we only need to search for a zero-knowledge proof for the confirmation protocol, which could be trivial regardless of its efficiency. Additional property like invisibility (anonymity) could be achieved by

searching for a message (signer)-independent signature simulator.

3.3 Relations between Security Notions

In this section, we discuss the relations between the security notions of a CVS scheme, the purpose of which is to find out whether one notion is implicitly implied in the other or they are exclusive, and under what conditions such an implication exists. With this knowledge, one could simply focus on a smaller set of security properties when designing a CVS scheme or analyzing its security. Before doing so, we first consider the difference between the notions of indistinguishability with and without adaptive queries.

In general, we could possibly view the definitions of simulatability, invisibility and anonymity as a formulation of indistinguishability between two random distributions, that is, between a partial signature and the corresponding simulator output in simulatability, between signatures on the same message and condition set but using two different signing keys in anonymity, and between signatures of the same signer on the same condition set but for two different messages in invisibility. These notions of indistinguishability are computational in the sense that there is some kind of trapdoor, for each set C of verifiability conditions, without the knowledge of which the distributions in question are indistinguishable. However, if the trapdoor is known, then anyone could distinguish which distribution a give entity belongs to. In CVS, this trapdoor is the needed witness signatures or endorsements specified in C. If one knows all the witness signatures needed for a given partial signature, he could extract the ordinary signature from it to check whether the extracted signature is valid to distinguish between the two distributions: $\{\mathsf{CVSig}_S(m, C)\}$ and $\{\mathsf{Fake}_S(m, C)\}$.

In our definitions, we adopt a strong type of adversary which is allowed to query the trapdoors (witness signatures) for other verifiability conditions but not exactly the same set of conditions in question. It can be seen that two indistinguishable distributions may not remain indistinguishable to an adversary not given the needed trapdoor but allowed to query other trapdoors. Whereas, the indistinguishability between two distributions in the adaptive trapdoor query model guarantees indistinguishability to any adversary without the knowledge of the needed trapdoor.

3.3.1 Ensuring Cheat-immunity

The following theorem allows one to ignore the cheat-immunity requirement when designing a CVS scheme as long as he could ensure the scheme is unforgeable and simulatable and its confirmation protocol is zero-knowledge.

Theorem 8 *An unforgeable and simulatable CVS scheme is also cheat-immune given its confirmation protocol is zero knowledge.*

Proof See Appendix A. ■

3.3.2 Equivalence between the Notions of Simulatability, Invisibility and Anonymity

The definitions of invisibility and anonymity in this paper are adopted from the work on undeniable signatures and designated confirmer signatures in the literature including [4, 37, 23, 34, 6, 18, 1]. Learning from the experience of this line of work, whether to consider invisibility or anonymity as the design goal could be perplexing sometimes. As a result, we attempt to sort out whether one notion is implicitly implied by the other in the context of CVS and under what assumptions or conditions such an implication exists. The following theorems show the implication and separation between the notions of simulatability and anonymity and invisibility.

Theorem 9 *Given a simulatable CVS scheme in an adaptive query model with respect to a PPT fake signature simulator* $\mathsf{Fake}_S(m, C)$*, it is message-invisible in the same adaptive query model if and only if* $\{\mathsf{Fake}_S(m_0, C)\} \cong \{\mathsf{Fake}_S(m_1, C)\}$ *in the same adaptive query model for all* S*,* m_0*,* m_1*, and* C*.*

Proof See Appendix A. ■

Theorem 10 *Given a simulatable CVS scheme in an adaptive query model with respect to a PPT fake signature simulator* $\mathsf{Fake}_S(m, C)$*, it is signer-anonymous in the same adaptive query model if and only if* $\{\mathsf{Fake}_{S0}(m, C)\} \cong \{\mathsf{Fake}_{S1}(m, C)\}$ *in the same adaptive query model for all* $S0$*,* $S1$*,* m*, and* C*.*

Proof See Appendix A. ■

Corollary 11 *(1) A message-independent simulatable CVS scheme is also message-invisible. (2) A signer-independent simulatable CVS scheme is also signer-anonymous. (3) An independently simulatable CVS scheme is both message-invisible and signer-anonymous.*

Proof Proof follows directly from Theorems 9 and 10. ■

Theorem 12 *Message invisibility of a CVS scheme does not implies its simulatability.*

Proof We show that the condition necessary for a message-invisible CVS scheme to be simulatable is there exists a fake signature simulator $\mathsf{Fake}_S(m, C)$ for all S, m, C such that $\{\mathsf{Fake}_S(m, C)\} \cong \{\mathsf{CVSig}_S(m, C)\}$ in the same adaptive query model. This condition itself is already sufficient to guarantee the simulatability of the CVS scheme. Hence, we could conclude that invisibility does not imply simulatability in any sense.
For details, please see Appendix A. ■

Theorem 13 *Signer anonymity of a CVS scheme does not implies its simulatability.*

Proof We show that the condition necessary for a signer-anonymous CVS scheme to be simulatable is there exists a fake signature simulator $\mathsf{Fake}_S(m, C)$ for all S, m, C such that $\{\mathsf{Fake}_S(m, C)\} \cong \{\mathsf{CVSig}_S(m, C)\}$ in the same adaptive query model. This condition itself is already sufficient to guarantee the simulatability of the CVS scheme. Hence, we could conclude that anonymity does not imply simulatability in any sense.
For details, please see Appendix A. ■

Theorem 14 *Assuming the partial signatures of a CVS scheme generated from two distinct and independently picked public/private key pairs (i.e. from two different signers) are independent, an anonymous CVS scheme is also invisible.*

Proof See Appendix A. ■

We believe that the condition for Theorem 14 to hold is usually fulfilled in practice. Hence, anonymity should imply invisibility.

It should be noted that the reduction used in proving these theorems or showing the implications bases on no additional computational assumption, and effectively no extra computation is needed in achieving such reduction. Therefore, these results could be applied to a fairly board and general scenarios. Besides, we use the weakest possible assumptions or conditions sufficient for such implications to hold.

4 Preliminaries

In Section 4.1, we present a number of cryptographic primitives necessary for the generic CVS construction to be discussed in Section 5. Besides, we review bilinear pairings in Section 4.2 which is needed for the efficient CVS construction in Section 6.

4.1 Basic Primitives

We consider four types of cryptographic primitives mainly used for the generic construction of CVS in Section 5. They are: identity based encryption (IBE), signatures (SIG), multi-bit commitments (COM), and pseudorandom generators (PRG).

4.1.1 Identity Based Encryption

We use similar notations as in [2] for identity based encryption. A standard IBE scheme $IBE = \{Setup, Extract, Enc, Dec\}$ consists of a private key generator (PKG) and a number of users, and is made up of four algorithms:

$Setup(1^\lambda) \rightarrow (PK_G, sk_G)$**:** the key setup algorithm which outputs a public/private key pair (PK_G, sk_G) for the PKG.

$Extract(ID, sk_G) \rightarrow d_{ID}$**:** the private key extraction algorithm run by PKG which outputs a private key d_{ID} for the identity ID.

$Enc(PK_G, ID, M) \rightarrow C$**:** the encryption algorithm taking an identity ID and a message m to output the ciphertext C.

$Dec(PK_G, C, d_{ID}) \rightarrow M$**:** the decryption algorithm taking a ciphertext C and a private key d_{ID} to output the plaintext M.

Note that, unlike the description in [2], we incorporate all the public parameters in the PKG public key PK_G, and this public key is needed in all encryption and decryption.

Security of IBE.

In [2], Boneh and Franklin considered the strongest security notion for IBE, namely semantic security or indistinguishability against an adaptive chosen ciphertext attack (IND-ID-CCA). Although chosen-ciphertext security is the standard acceptable notion for encryption schemes, we only consider a weaker notion — semantic security against a chosen plaintext attack (IND-ID-CPA) or semantic security for short — which is sufficient for our generic construction of CVS. An IBE is semantically secure if no PPT adversary $\mathcal{A}$ could win the following game with a non-negligible advantage:

Setup. The challenger runs $Setup$ to generate a PKG public/private key pair (PK_G, sk_G), and gives the public key PK_G to the adversary but keeps the private/master key sk_G.

Query 1. The adversary could issue to the challenger one type of queries:

- Extraction Query $\langle ID_j \rangle$. The challenger responds by running $Extract$ on ID_j to generate the corresponding private key $d_j = d_{ID_j}$ and gives it to the adversary.

Challenge. Once the adversary decides that the first query phase is over it outputs two plaintexts M_0, M_1 and an identity ID to be challenged. The only constraint is that ID did not appear in any of the previous extraction queries, that is, $ID \neq ID_j, \forall j$. The challenger flips a coin $b \in \{0,1\}$, set $C = Enc(PK_G, ID, M_b)$ and sends C to the adversary.

Query 2. The adversary is allowed to make more queries as previously done but no query can be made on the challenged ID.

Guess. Finally, the adversary outputs a guess $b' \in \{0,1\}$ for b.

Result. The adversary wins the game if $b' = b$. The advantage of the adversary is defined as:

$$Adv_{\mathcal{A}}^{IBE} = \left| Pr[b' = b] - \frac{1}{2} \right|$$

4.1.2 Signatures

A signature scheme $SIG = \{SKG, Sig, Ver\}$ consists of three algorithms:

$SKG(1^\lambda) \rightarrow (PK_S, sk_S)$**:** the key generator which generates the public/private key pair (PK_S, sk_S) for a signer S.

$Sig(m, sk_S) \rightarrow \sigma$**:** the signing algorithm taking a message m and a private key sk_S to output a signature σ on m.

$Ver(m, \sigma, PK_S) \rightarrow v \in \{0,1\}$**:** the signature verification algorithm taking a message m a signature σ and a public key PK_S to check whether σ is a valid signature of S on m. If it is, Ver outputs 1, otherwise, 0.

Security of SIG.

A signature scheme is considered secure if the probability of successful existential forgery is negligible even under chosen message attacks. In details, this means the following: An adversary $\mathcal{A}$ is allowed to make oracle access adaptively to obtain signatures of a targeted signer S on any message m_j of his choice; he could make a query based on the results of the previous queries. Finally, $\mathcal{A}$ has to output a message-signature pair (m, σ). The probability that the signature is a valid one for the message (i.e. $Ver(m, \sigma, PK_S) = 1$) and the message has not be queried before (i.e $m \neq m_j, \forall j$) should be negligible for all PPT $\mathcal{A}$.

4.1.3 Pseudorandom Generators

Assume $l(n) > n$. Let $x \leftarrow X$ denote that x is uniformly sampled from X. $h : \{0,1\}^n \rightarrow \{0,1\}^{l(n)}$ is a pseudorandom generator [24] if the following is negligible in n for all PPT distinguisher D:

$$\left| Pr[y \leftarrow \{0,1\}^{l(n)} : D(y) = 1] - Pr[s \leftarrow \{0,1\}^n : D(h(s)) = 1] \right|$$

This in essence means that h take a seed s to generate a string $h(s)$ of longer length $l(n)$ and nobody could distinguish $h(s)$ from a uniformly sampled string from $\{0,1\}^{l(n)}$.

4.1.4 Commitments

We adopt the multi-bit commitment definitions [35, 15] instead of the common single-bit commitment [35]. The core of a cryptographic commitment scheme is the committing algorithm $Com(s, m) \rightarrow c$ on input a message m and a randomly chosen salt s outputting a commitment c. By revealing s and m, one can check whether a commitment c is properly formed. A commitment scheme should satisfy the following properties:

Binding. Let λ be the security parameter, then the following is negligible (computationally binding) or zero (perfectly binding) for all PPT algorithm A:

$$Pr[(s, m, s', m') \leftarrow \{A(1^{\lambda})\} : Com(s, m) = Com(s', m')]$$

Hiding. For all $m, m' \in \{0,1\}^*, m \neq m'$, the following is negligible (computationally hiding) or zero (perfectly hiding) for all PPT distinguisher D:

$$\left|Pr[s \leftarrow \{0,1\}^*; c \leftarrow Com(s, m) : D(c) = 1] - Pr[s' \leftarrow \{0,1\}^*; c \leftarrow Com(s', m') : D(c') = 1]\right|$$

The binding property essentially means that, once a message m is committed in c, nobody could change its value without being detected. In perfectly hiding schemes, the distribution of the commitments for different messages should be identical. Note that we use a different definition for the hiding property than that of the multi-bit scheme in [35] which states that, for a message $m = b_1 b_2 \dots b_n$ ($b_i \in \{0,1\}, 1 \leq i \leq n$), given a commitment on m, nobody could guess any bit b_i correctly with a probability greater than $\frac{1}{2} + \varepsilon(\lambda)$ (where $\epsilon(\lambda)$ is a negligible function in λ), even when told $b_1, b_2, \dots, b_{i-1}, b_{i+1}, \dots, b_n$. However, it could be shown that the two definitions are equivalent.

4.2 Bilinear Pairings

In this section, we briefly review the basic concepts of bilinear pairings and the related computational problems. Let $\mathbb{G}_1$ be a cyclic additive group generated by G, whose order is a prime q, and $\mathbb{G}_2$ be a cyclic multiplicative group with the same order q. A bilinear pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties:

1. **Bilinearity:** $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ where $P, Q \in \mathbb{G}_1$, $a, b \in \mathbb{Z}_q^*$.
2. **Non-degeneracy:** $\hat{e}(P, P) \neq 1$. Therefore, it is a generator of $\mathbb{G}_2$
3. **Computability:** There is an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

In this paper, we will write $\mathbb{G}_1$ with an additive notation and $\mathbb{G}_2$ with a multiplicative notation as implementations of $\mathbb{G}_1$ are usually groups of points on an elliptic curve. The discussion of this paper is based on choosing groups in which the following computational problems are assumed to be hard or any PPT solution to them is negligibly better than a wide guess.

Definition 14 Computational Bilinear Diffie-Hellman (CBDH) Problem*: Given $P \in \mathbb{G}_1$, aP, bP and cP for some unknowns a, b, $c \in \mathbb{Z}_q^*$, find $\hat{e}(P, P)^{abc}$.*

Definition 15 Decisional Bilinear Diffie-Hellman (DBDH) Problem*: Given $P \in \mathbb{G}_1$, aP, bP and cP for some unknowns a, b, $c \in \mathbb{Z}_q^*$, decide whether a given $y \in \mathbb{G}_2$ satisfies that $y \stackrel{?}{=} \hat{e}(P, P)^{abc}$.*

5 The Existence of a Secure CVS Scheme

In this section, we give a generic CVS construction from IBE and show the equivalence between CVS and IBE in terms of existence.

5.1 A Generic Construction of CVS from IBE

We show how to construct a secure CVS scheme based on the following components:

- A secure signature $SIG = (SKG, Sig, Ver)$ which is existentially unforgeable under an adaptive chosen message attack [27].
- An identity base encryption scheme $IBE = (Setup, Extract, Enc, Dec)$ with semantic security, that is, IND-ID-CPA [2].
- A computationally hiding commitment scheme $COM = (Com)$ [35, 15].
- A pseudorandom generator [24, 29].

Let the plaintext and ciphertext spaces of IBE be $\mathcal{P}_{IBE}$ and $\mathcal{C}_{IBE}$ respectively.
Let the message and signature spaces of SIG be $\mathcal{M}$ (same as the message space of CVS) and $\mathcal{S}_\sigma$ (same as the ordinary signature space of CVS) respectively.
Let $h : \{0,1\}^{l_p} \rightarrow \{0,1\}^{l_s}$ be a pseudonrandom generator where l_p and l_s are the length of an IBE plaintext and a SIG signature respectively.
Let $\mathcal{C}_{COM}$ be the output space of the commitment scheme COM and $Com : \mathcal{P}_{IBE} \times \mathcal{S}_\sigma \rightarrow \mathcal{C}_{COM}$ be its committing function.

Denote the signer by S, and the witnesses by W_i. Depending on the number of witnesses, the IBE scheme is used multiple times with each witness W_i being a private key generator for its IBE scheme (IBE_i). Assume there are N witnesses, then the partial signature $\delta \in \mathcal{S}_\sigma \times \mathcal{C}_{IBE}^N \times \mathcal{C}_{COM}$. The generic CVS construction is as follows.

Key Generation. CVKGS $\stackrel{\text{def}}{=} SKG$ for generating (Pk_S, sk_S) for the signer S.
CVKGW $\stackrel{\text{def}}{=} Setup$ for generating (PK_{W_i}, sk_{W_i}) for the witnesses W_i.

Partial Signature Generation. Given an input message $m \in \mathcal{M}$, a condition set $C = \{(c_i, W_i) : 1 \leq i \leq N\}$, a signing key sk_S, a signer's public key PK_S and the set of witness public keys $PK_C = \{PK_{W_i} : 1 \leq i \leq N\}$, do the following:

1. Generate an ordinary signature using the signing algorithm of SIG:
$$\sigma = Sig(m, sk_S)$$
2. For each $(c_i, W_i) \in C$, pick a random $a_i \in \mathcal{P}_{IBE}$, $1 \leq i \leq N$.
3. The CVS signature is as follows:
$$\delta = \left\langle \sigma \oplus h\left(\bigoplus_i^N a_i\right),\ \{Enc(PK_{W_i}, c_i, a_i) : 1 \leq i \leq N\},\ Com\left(\sigma, h\left(\bigoplus_i^N a_i\right)\right)\right\rangle$$
where $Enc(PK_{W_i}, c_i, a_i)$ is the IBE ciphertext on message a_i using W_i (witness) as the PKG and c_i (condition statement) as the identity.

Note: for short, we may denote $Enc(PK_{W_i}, c_i, a_i)$ as $Enc_{W_i}(c_i, a_i)$ in the following discussion.

Witness Signature Generation. $\mathsf{SigW}(c, sk_W) \stackrel{\text{def}}{=} Extract(c, sk_W)$.

Taking the condition statement c as an identity, the witness W (run as an PKG of the IBE scheme) could extract the private key d_c^W corresponding to c. The private key d_c^W could be considered as a kind of trapdoor on the condition statement c which could be generated by the witness W only. Roughly, it could also be considered as a kind of signature as in [3].

Signature Extraction. Given a partial signature $\delta = \langle \alpha, \{\beta_i : 1 \leq i \leq N\}, \gamma \rangle$ and all the witness signatures $\{\sigma_i\} = \{d_{c_i}^{W_i}\}$ (with each σ_i being a signature or endorsement on the condition (c_i, W_i)), do the following:

1. For $1 \leq i \leq N$, get $a'_i = Dec(PK_{W_i}, \beta_i, \sigma_i)$.
2. Recover $\sigma' = \alpha \oplus h(\bigoplus_i^N a'_i)$.
3. Check if $Com(\sigma', h(\bigoplus_i^N a'_i)) \stackrel{?}{=} \gamma$. If not, output "extraction-fail", otherwise, σ' is the ordinary signature.

Note that the partial signature is slightly over-designed as it uses a commitment scheme to guard against adversaries tampering with a partial signature.

Signature Verification. $\mathsf{VerS} \stackrel{\text{def}}{=} Ver$.

Confirmation Protocol. Using general interactive zero-knowledge proofs [25] or concurrent zero-knowledge proofs [13], the signer with private input $a_1, a_2, \ldots, a_i, \ldots, a_N$ and σ and all the random coins used to generate β_i could convince the verifier that there exists $(\sigma, a_1, a_2, \ldots, a_i, \ldots a_N)$ satisfying the following equations:

$$\begin{aligned}
&\delta = \langle \alpha, \{\beta_1, \beta_2, \ldots, \beta_i, \ldots, \beta_N\}, \gamma \rangle \\
&\alpha = \sigma \oplus h\left(\bigoplus_i^N a_i\right) \\
&\beta_i = Enc(PK_{W_i}, c_i, a_i), 1 \leq i \leq N \\
&\gamma = Com\left(\sigma, h\left(\bigoplus_i^N a_i\right)\right) \\
&Ver(m, \sigma, PK_S) = 1
\end{aligned}$$

The common input to the confirmation protocol is PK_S, PK_{W_i} $(1 \leq i \leq N)$, m, $C = \{(c_i, W_i) : 1 \leq i \leq N\}$ and δ. Since verifying whether a given tuple $(\sigma, a_1, a_2, \ldots, a_i, \ldots a_N)$ satisfies the above equations is a poly-time predicate, a general zero-knowledge proof for it should exist. The construction is straightforward but inefficient and varying depending on the signature and IBE schemes in use. The simulated transcript generator for this zero-knowledge proof is used as the transcript simulator FakeT for the following fake partial signature simulator Fake.

Fake Signature Simulator

The fake signature simulator for this CVS construction is the following:

$\mathsf{Fake}(C)$: $\quad C = \{(c_i, W_i) : 1 \leq i \leq N\}$

1. Randomly (uniformly) pick $\sigma_f \in \mathcal{S}_\sigma$.

2. Randomly pick $b_i \in \mathcal{P}_{IBE}$, for $1 \leq i \leq N$.

3. Output the fake partial signature:

$$\delta_f = \left\langle \sigma_f \oplus h\left(\bigoplus_i^N b_i\right),\ \{Enc(PK_{W_i}, c_i, b_i) : 1 \leq i \leq N\},\ Com\left(\sigma_f, h\left(\bigoplus_i^N b_i\right)\right)\right\rangle$$

Obviously, this simulator is PPT.

5.1.1 Security of the Generic CVS Construction

The completeness of the above CVS construction is guaranteed by the correctness of the underlying IBE scheme. Besides, it is also perfectly convertible. The security of this CVS construction is best summarized with the following lemmas and theorem.

Lemma 15 *If SIG is existentially unforgeable under an adaptive chosen message attack, then the generic CVS construction is unforgeable.*

Proof See Appendix B. ■

Lemma 16 *If IBE is IND-ID-CPA secure, COM is a computationally hiding commitment scheme, and h is a pseudorandom generator, then the generic CVS construction is simulatable with respect to the simulator Fake.*

Proof See Appendix B. ■

Theorem 17 *Given any semantically secure IBE scheme (under a chosen plaintext attack) and any existentially unforgeable signature scheme, together with a pseudorandom generator and a computationally hiding commitment scheme, a secure CVS scheme can be constructed.*

Proof We could use the generic paradigm described in this section to construct a CVS scheme and the corresponding fake partial signature simulator Fake satisfying the following properties: unforgeability (according to Lemma 15), simulatability with respect to Fake (according to Lemma 16). As mentioned before, a zero knowledge proof exists for the given construction and could be used as the confirmation protocol. Together with the simulatability property, the construction is non-transferable. The completeness and soundness of the confirmation protocol is guaranteed by the zero knowledge proof. Besides, it could be seen that this construction is perfectly convertible. In conclusion, this generic CVS construction is secure. In addition to its security, this construction enjoys additional properties of message-invisibility and signer-anonymity since the partial signature simulator Fake does not take the message or the signer's identity as input. ■

5.2 A Generic Construction of IBE from CVS

We show how to construct a 1-bit IBE scheme with semantic security (i.e. IND-ID-CPA) using a CVS scheme. We assume the CVS scheme is simulatable with respect to a fake partial signature simulator Fake. Our construction is similar to that in the seminal work of probabilistic encryption by Goldwasser and Micali [26]. While they used the indistinguishability between the quadratic residues and non-residues in $\mathbb{Z}_n*$ for some composite n (Quadratic Residuosity Problem) to encrypt a single bit, we leverage the indistinguishability between a true and a simulated (fake) partial signature of CVS to create a ciphertext for a 1-bit plaintext.

By repeating the operation of the 1-bit scheme k times, we could construct an IBE scheme for k-bit long messages. This repetition technique is the same as in [26] and, using the same hybrid argument, we could prove that the security property of the underlying 1-bit scheme is preserved in the k-bit one.

Now, we just need to focus on a 1-bit IBE scheme. We consider a CVS scheme with just a single witness $G \in \mathcal{W}$ which is used as the PKG for the IBE scheme. Suppose Fake is a PPT simulator for the CVS scheme. The IBE scheme works as follows.

Key Setup. The public and private keys of the witness G in the CVS scheme are used as the public and private keys of the PRG in the IBE scheme. We set $Setup \stackrel{\text{def}}{=} \mathsf{CVKGW}$ to generate the public/private key pair of the PRG: $\mathsf{CVKGW}(1^\lambda) \rightarrow (PK_G, sk_G)$.

Private Key Extraction. The identity ID_i of any user could be treated as a condition statement in the CVS scheme as they are both a bit string of arbitrary length. We set $Extract \stackrel{\text{def}}{=} \mathsf{SigW}/\mathsf{CVEndW}$, then extracting the private key d_i for ID_i is the same as requesting an endorsement or signature on the statement ID_i: $\mathsf{SigW}(ID_i, sk_G) \rightarrow d_i$.

Encryption. The identity of a user i is the bit string ID_i (treated as a condition statement in the underlying CVS scheme) and its private key is the witness endorsement d_i obtained from G.

We consider a 1-bit plaintext $b \in \{0, 1\}$. To encrypt,

- randomly pick a message $m \in \mathcal{M}$
- run $\mathsf{CVKGS}(1^\lambda)$ to generate the public/private key pair (PK_S, sk_S) of the signer
- the encryption function is then: $Enc(PK_G, ID_i, b) \rightarrow (m, \delta_b, PK_S)$, where

$$\delta_b = \begin{cases} \mathsf{CVSig}(m, ID_i, sk_S, PK_S, PK_G), & b = 0 \\ \mathsf{Fake}(m, ID_i, PK_S, PK_G), & b = 1 \end{cases}$$

That is, when $b = 0$, δ_b is a valid partial signature on m, whereas, when $b = 1$, δ_b is a fake one.

Decryption. Given an identity ID_i, a PKG public key PK_G and the user private key d_i, to decrypt a given ciphertext $C = (m', \delta', PK'_S)$, the decryption function $Dec(PK_G, C, d_i) \rightarrow b$ is implemented as follows:

- extract the ordinary signature from δ': $\mathsf{CVExtract}(m', ID_i, \delta', PK'_S, d_i) \rightarrow \sigma'$
- check if $\mathsf{VerS}(m', \sigma', PK'_S) \stackrel{?}{=} 1$, the plaintext b' is given by the following[11]:

$$b' = \begin{cases} 0, & if\ \mathsf{VerS}(m', \sigma', PK'_S) = 1 \\ 1, & otherwise \end{cases}$$

[11]The case in which CVExtract returns $\perp$ is covered by the "otherwise" part.

5.2.1 Correctness of the CVS-based IBE

If all the algorithms used in the CVS scheme are polynomial time, then so are those used in the above IBE construction. The completeness of the CVS scheme guarantees the correctness of decryption in the above IBE scheme. The completeness property of the CVS scheme ensures that,
if $\delta = \mathsf{CVSig}(m, ID_i, sk_S, PK_S, PK_G)$ and $d_i = \mathsf{CVEndW}(ID_i, sk_G)$, then the verification must return 1, that is, $\mathsf{VerS}(m, \mathsf{CVExtract}(m, ID_i, \delta, PK_S, d_i), PK_S) = 1$. Besides, the CVS scheme also guarantees that with negligible probability a valid ordinary signature on message m could be extracted form $\mathsf{Fake}(m, ID_i, PK_S, PK_G)$, otherwise, the CVS scheme would be forgeable. These together ensure that $Dec(PK_G, Enc(PK_G, ID_i, b), d_i) = b$ with probability almost 1 up to a negligible deviation.

5.2.2 Security of the CVS-based IBE

The security of above IBE construction is stated by the following theorem.

Theorem 18 *The above IBE construction from CVS is semantic secure against a chosen plaintext attack (IND-ID-CPA).*

Proof See Appendix C. ■

5.3 The Equivalence between CVS and IBE

A secure CVS scheme is equivalent to a secure IBE scheme, in terms of existence, which is summarized by the following theorem.

Theorem 19 *A secure conditionally verifiable signature (CVS) scheme (unforgeable, simulatable, with zero knowledge confirmation protocol) exists if and only if an IND-ID-CPA secure identity based encryption (IBE) scheme exists.*

Proof

Only if Part

Follow directly from the CVS-based IBE construction in the last section.

If Part

We assume the existence of a secure identity based encryption scheme with security in the IND-ID-CPA sense. Then it is straightforward to see why a one-way function exists (We could use $Setup$ of the IBE scheme to construct a one-way function.).

First, an ordinary signature scheme which is not existentially forgeable under an adaptive chosen message attack [27] exists since such a secure signature scheme exists if and only if one-way function exists [39, 36].

Second, a pseudorandom generator exists as Impagliazzo et. al. [29] showed that given any one-way function, a pseudorandom generator can be constructed.

Third, a computationally hiding bit commitment function exists if a pseudorandom generator exists [35]. In the same work, Naor show how to construct a multi-bit commitment scheme from any pseudorandom generator. That is, along the chain from IBE to one-way functions to pseudorandom generators and finally to commitment schemes, the existence of IBE implies the existence of a computationally hiding commitment scheme.

Finally, the existence of a one-way function also implies the existence of zero-knowledge proofs.

Based on a secure IBE scheme, an existentially unforgeable signature scheme, a pseudorandom generator, and a computationally hiding commitment scheme, from Theorem 17, we could use the generic construction in this section to build a secure CVS scheme which is unforgeable and simulatable and a zero knowledge for its confirmation protocol exists. Hence, the existence of a secure IBE scheme implies the existence of a secure CVS scheme. ■

We should mention that we show in Theorem 19 that a weaker notion of IBE, namely, one with IND-ID-CPA security, is necessary and sufficient for the construction of a secure CVS scheme. It is thus fair to say that CVS could be constructed based on weaker assumptions than IBE with the standard IND-ID-CCA security [2].

6 Efficient CVS Constructions from Bilinear Pairings

Although we do not make the restriction that the CVS partial signature generation has to start with an ordinary signature (i.e. a 2-step generation), it is more convenient and efficient to proceed in this way in practice. A typical design of the CVS partial signature generation would then consists of three components:

1. An ordinary signature scheme is chosen. The only criteria for such a choice is that the signature scheme is existentially unforgeable under a chosen message attack.

2. A blinding mechanism is designed to transform an ordinary signature into a partial signature, making it covert inside the partial signature. The main design criteria of the blinding mechanism is the simulatability property; that means we also need to find a simulator for the chosen blinding mechanism. We show in Section 5 that identity based encryption (IBE) could in general be used as a blinding mechanism.

3. A zero knowledge confirmation protocol is designed to provide the signer a means to give the recipient some guarantee that a real signature could be retrieved from a given partial signature.

The following theorem would be useful for designs based on this paradigm.

Theorem 20 *Given an ordinary signature scheme SIG existentially unforgeable under a chosen message attack*[12]*, a CVS scheme constructed from SIG using any PPT blinding mechanism B is unforgeable.*

Proof See Appendix F. ■

We could use the general verifiable encryption (VE) approach [5, 1] to construct a CVS scheme (Details could be found in the Appendix D), but the resulting construction would have a large partial signature (which is inconvenient for storage), for instance, an online trader may receive a huge number of partial signatures daily for payment authorization. In some cases, a zero knowledge proof is not achievable from the VE approach [28]. A well adjusted balance between the performance of the blinding mechanism and the confirmation protocol is necessary to achieve efficient schemes. In practice, to construct efficient schemes with a practical confirmation protocol, we need to fix the underlying signature scheme and blinding mechanism and pose restrictions on their parameter dependency. In this section, we show how to use bilinear pairings to construct efficient CVS schemes for Elgamal and RSA signatures.

[12] In a chosen message attack, the adversary is given the signatures of messages of his choice.

Suppose $\mathbb{G}_1$ and $\mathbb{G}_2$ are additive and multiplicative cyclic groups of order q (prime) respectively and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a computable, non-degenerate bilinear map. Assume that $\mathbb{G}_2$ is a subgroup of some extension field of $\mathbb{F}_p$, say $\mathbb{F}_{p^l}$, similar to those in Weil or Tate pairings. Depending on the underlying ordinary signature scheme used, we have different restrictions on p. A sufficiently large p matched with the parameters in the signature scheme should be enough to cover most restrictions. We also need the following cryptographic hash function in our constructions which is modeled as a random oracle:

$$H : \{0,1\}^* \rightarrow \mathbb{G}_1.$$

Witness Key and Signature Generation

With possibly slight variations, these algorithms are in essence the same for different ordinary signature schemes, both Elgamal and RSA.

Witness Key Generation (CVKGW). Each witness W_i picks a generator of $\mathbb{G}_1$, say P_i, and its private key $x_i \in \mathbb{Z}_q^*$ and publishes the corresponding public key (P_i, Y_i) where $Y_i = x_i P_i$.

Witness Signature Generation (SigW/CVEndW). The witnesses use a pairing based signature scheme [3] to generate its signature σ_{W_i} or endorsement on a condition statement c_i as follows: $\sigma_{W_i} = x_i H(c_i)$

Witness Signature Verification. Given a witness public key (P_i, Y_i), a condition statement c_i and a witness signature σ_{W_i}, the verification is done by checking whether $\hat{e}(Y_i, H(c_i)) \stackrel{?}{=} \hat{e}(P_i, \sigma_{W_i})$. The correctness is ensured as $\hat{e}(Y_i, H(c_i)) = \hat{e}(x_i P_i, H(c_i)) = \hat{e}(P_i, x_i H(c_i)) = \hat{e}(P_i, \sigma_{W_i})$.

Important Notation Conventions. As we would frequently use the pairing values in the following discussion, we should clarify some notations before we move on. Given a witness public key (P_i, Y_i) and a condition statement c_i, and a random coin $r \in \mathbb{Z}_q^*$ with $U_i = rP_i$, we often have the following notations for the following pairing values:

- $e_i = \hat{e}(P_i, H(c_i)) \in \mathbb{G}_2$
- $y_i = \hat{e}(Y_i, H(c_i)) = \hat{e}(P_i, H(c_i))^{x_i} = e_i^{x_i} \in \mathbb{G}_2$
- $w_i = \hat{e}(U_i, H(c_i)) = \hat{e}(P_i, H(c_i))^r = e_i^r \in \mathbb{G}_2$

6.1 A Pairing-based CVS Construction for Generalized ElGamal Signatures

We describe the construction for the ElGamal signature scheme but the techniques should apply to other DL based schemes like DSA and Schnorr signatures [40]. In fact, we use a general cyclic group G of order q' (where q' is a large safe prime[13]) for the sake of generality. In order to give an efficient confirmation protocol, we require $q' = p$ where p is the characteristic of the extension field of which $\mathbb{G}_2$ of the bilinear pairing is a subgroup (i.e. $\mathbb{G}_2$ is a multiplicative subgroup of $\mathbb{F}_{p^l}$ for some integer l).

[13] That is, $q' - 1$ is a multiple of another large prime.

6.1.1 Ordinary Signature

Let $h : \{0,1\}^* \rightarrow \mathbb{Z}_{q'}$ (which is modeled as a random oracle). The message space is $\{0,1\}^*$ and the signature space is $G \times \mathbb{Z}_{q'}$.

Key Generation (CVKGS). The signer picks a generator of G, say g, picks its private key $x_s \in \mathbb{Z}_{q'}^*$ and publishes the corresponding public key (g, y_s) where $y_s = g^{x_s}$.

Signing SigS. For a message $m \in \{0,1\}^*$, the signature generation is as follows:

1. Pick a random $k \in \mathbb{Z}_{q'}^*$, and compute $\gamma = g^k$.
2. Compute $a = k^{-1}[h(m) + x_s\gamma] \bmod q'$.
3. The ordinary signature is: $\sigma = (\gamma, a)$.

Verification (VerS). Given a signature $\sigma' = (\gamma', a')$, to verify whether it is a valid signature of message m, check the following: $\gamma'^{a'} \stackrel{?}{=} g^{h(m)} y_s^{\gamma'}$

Note that we do not take the repaired version of the ElGamal signature scheme, but the discussions in this paper apply directly to the repaired version which replaces $h(m)$ by $h(m, \gamma)$.

6.1.2 Partial Signature

Given an ElGamal signature $\sigma = (\gamma, a)$, we could use the generic IBE-based approach described in Section 5 to simply multiply a with a number of pairing values to make the signature convert and non-verifiable, and use the technique in [41] to run the confirmation protocol. However, a does not fit into $\mathbb{G}_2$ which the pairing values belong to. Hence, the double decker techniques in [41] is not applicable. To glue the pairing-based blinding mechanism with the proof technique in [41], we need to introduce an invertible group homomorphism, a mapping $f : \mathbb{Z}_{q'} \rightarrow \mathbb{F}_{p^l}$ with an inverse mapping f^{-1}. For such a mapping to exist, we set $q' = p$, that is, the mapping becomes $f : \mathbb{Z}_p \rightarrow \mathbb{F}_{p^l}$. We require that

- $f^{-1}(f(a)) = a, \forall a \in \mathbb{Z}_p$,
- $f(a_1 a_2) = f(a_1) f(a_2), \forall a_1, a_2 \in \mathbb{Z}_p$, and
- $f^{-1}(e_1 e_2) = f^{-1}(e_1) f^{-1}(e_2), \forall e_1, e_2 \in \mathbb{F}_{p^l}$.

Such a mapping could be constructed using the norm of $\mathbb{F}_{p^l}$ (See Appendix E).

If there are N ($< L$) witnesses specified in a partial signature δ, then $\delta \in G \times \mathbb{F}_{p^l} \times \mathbb{G}_1^N$. Now we can describe the blinding mechanism and the signature retrieval process.

Blinding (CVSig). Given a signer private key x_s, a message m, a verifiability condition set $C = \{(c_i, W_i) : 1 \leq i \leq N\}$ and the witness public keys $\{(P_i, Y_i) : 1 \leq i \leq N\}$, we create a partial signature δ as follows:

1. Run SigS on m to generate an ordinary signature $\sigma = (\gamma, a)$.
2. Randomly pick $r \in \mathbb{Z}_q^*$, and compute $U_i = rG_i,\ 1 \leq i \leq N$.
3. Compute $z = f(a) \prod_{i=1}^N \hat{e}(Y_i, H(c_i))^r = f(a) \prod_{i=1}^N y_i^r = f(a) \prod_{i=1}^N e_i^{x_i r}$.
4. The partial signature is then given by: $\delta = (\gamma, z, U_1, U_2, \ldots, U_N)$.

Signature Retrieval (Extract). Given a partial signature $\delta' = (\gamma', z', U'_1, U'_2, \ldots, U'_N)$, and a set of witness signatures $\sigma_i = x_i H(c_i)$, $1 \le i \le N$, with each being a short signature of W_i on a condition statement c_i, the signature retrieval process is as follows.

1. Compute the following: $a' = f^{-1}\left(\frac{z'}{\prod_{i=1}^{N} \hat{e}(U'_i, \sigma_i)}\right)$
2. The recovered ordinary signature is then given by: $\sigma' = (\gamma', a')$.

Correctness of the Extraction. If the partial signature is properly formed, that is, the following set of equations holds for unknown $r \in \mathbb{Z}_q^*$ and $k \in \mathbb{Z}_k^*$:

$$\begin{array}{l} z' = f(a)\prod_{i=1}^{N} y_i^r = f(a)\prod_{i=1}^{N} e_i^{x_i r}; \\ U'_i = rG_i, \ \forall i; \\ \gamma' = g^k; \\ a = k^{-1}[h(m) + x_s \gamma']. \end{array} \tag{1}$$

Then the correctness of the ordinary signature retrieval is guaranteed as:

$$\begin{array}{l} \hat{e}(U'_i, \sigma_i) = \hat{e}(rG_i, x_i H(c_i)) = \hat{e}(G_i, H(c_i))^{x_i r} = e_i^{x_i r}, \ \forall i \\ a' = f^{-1}\left(\frac{f(a)\prod_{i=1}^{N} e_i^{x_i r}}{\prod_{i=1}^{N} \hat{e}(U'_i, \sigma_i)}\right) = f^{-1}\left(\frac{f(a)\prod_{i=1}^{N} e_i^{x_i r}}{\prod_{i=1}^{N} e_i^{x_i r}}\right) = f^{-1}(f(a)) = a \\ \gamma'^{a'} = g^{ka} = g^{kk^{-1}[h(m)+x_s\gamma']} = g^{h(m)} g^{x_s \gamma'} = g^{h(m)} y_s^{\gamma'}. \end{array}$$

Hence, $\mathsf{VerS}(m, (\gamma', a'), (g, y_s)) = 1$.

6.1.3 Partial Signature Simulator

A possible partial signature simulator Fake is as follows:

Fake(C)

Input: $C = \{(c_i, W_i) : 1 \le i \le N\}, \{(P_i, Y_i) : 1 \le i \le N\}, g$
Output: $\delta_f = (\gamma_f, z_f, V_1, V_2, \ldots, V_N)$

1. Randomly pick $k_f \in \mathbb{Z}_p^*$ and compute $\gamma_f = g^{k_f}$.
2. Randomly pick $r_f \in \mathbb{Z}_q^*$ and compute $V_i = r_f G_i$, $1 \le i \le N$.
3. Randomly pick $d \in \mathbb{Z}_p$ and compute $z_f = f(d)\prod_{i=1}^{N} \hat{e}(Y_i, H(c_i))^{r_f} = f(d)\prod_{i=1}^{N} y_i^{r_f} = f(d)\prod_{i=1}^{N} e_i^{x_i r_f}$.
4. Output $(\gamma_f, z_f, V_1, V_2, \ldots, V_N)$.

This simulator only uses the verifiability condition set C as input and neither the message nor the signer's information is needed. Hence, the ElGamal CVS construction enjoys the message-invisibility and signer-anonymity properties.

Claim 21 *The ElGamal based CVS construction given above is simulatable with respect to the simulator* Fake *if decisional bilinear Diffie-Hellmen problem is hard assuming H is a random oracle.*

Proof See Appendix F. ■

6.1.4 Confirmation Protocol

Let m be a message, $C = \{(c_i, W_i) : 1 \leq i \leq N\}$ be a verifiability condition set, $PK_S = (g, y_s)$ be a signer public key, and $PK_i = (G_i, Y_i)$, for $1 \leq i \leq N$, be the witness public key of W_i. Let also the partial signature be $\delta = (\gamma, z, U_1, U_2, \ldots, U_N)$. As can be seen above, if the set of equations in Equation 1 holds, anyone could be sure that a proper and correct ordinary signature can be retrieved from the given partial signature. Hence, in order to convince a recipient that a given partial signature is properly formed and a valid ordinary signature on the message m could be extracted if he obtains all the needed witness signatures specified in C, the signer just needs to prove that Equation 1 holds for the tuple $(m, C, PK_S, \{PK_i : 1 \leq i \leq N\}, \delta)$ using his private input (r, a). The confirmation protocol $\mathsf{CVCon}_{(S,V)}$ is then as follows.

$\langle \mathsf{CVConS}(r, a), \mathsf{CVConV}() \rangle (m, C, PK_S, \{PK_i : 1 \leq i \leq N\}, \delta)$

Common Input: $m; C = \{(c_i, W_i) : 1 \leq i \leq N\}; PK_S = (g, y_s); PK_i = (G_i, Y_i), 1 \leq i \leq N; \delta = (\gamma, z, U_1, U_2, \ldots, U_N)$

Signer Private Input: r, a

Protocol: Shown below is just one round of iteration, which should be run multiple rounds say k in the actual protocol. Recall that $e_i = \hat{e}(P_i, H(c_i))$, $y_i = \hat{e}(Y_i, H(c_i)) = e_i^{x_i}$, and $w_i = \hat{e}(U_i, H(c_i)) = e_i^r$. Let $\psi = g^{h(m)} y_s^{\gamma} = \gamma^a$ which is used in the ElGamal signature verification.

1. **Commit.** The signer randomly picks $u \in \mathbb{Z}_q^*$, computes and sends the following to the verifier:
$$t_i = e_i^u,\ 1 \leq i \leq N,\ ; \qquad t = \gamma^{f^{-1}(z \prod_{i=1}^{N} y_i^u)}$$
2. **Challenge.** The verifier uniformly picks $b \in_R \{0, 1\}$ and sends it to the signer.
3. **Response.** The signer sends back $\theta = u + br$.
4. **Verify.** The verifier then checks the validity of the following and accepts only if:
$$t_i \stackrel{?}{=} e_i^{\theta} w_i^{-b}; \qquad t \stackrel{?}{=} \gamma^{(1-b) f^{-1}(z \prod_{i=1}^{N} y_i^{\theta})} \psi^{b f^{-1}(\prod_{i=1}^{N} y_i^{\theta})}.$$

Claim 22 *The above confirmation protocol for the ElGamal based CVS construction is a zero-knowledge proof.*

Proof This protocol satisfies the completeness, soundness, and zero-knowledge properties. See Appendix F. ■

6.1.5 Security of the ElGamal based CVS Construction

Putting the pieces together, we could conclude that the ElGamal based CVS scheme is secure satisfying the properties of unforgeability (Theorem 20) and simulatability (Claim 21), which in turn imply the cheat-immunity property. Besides, its confirmation protocol is zero knowledge (Claim 22) which, together with the simulatability property, further implies the ElGamal based CVS construction is non-transferable.

6.2 A Pairing-based CVS Construction for RSA-based Signatures

We describe the construction for the basic hash-and-sign RSA signature scheme but the techniques should apply to other RSA variants like the GHR [22] signature scheme.

Suppose $n = p'q'$ where p', q' are large primes. We require that $n^2 < p$ where p is the characteristic of the extension field of which $\mathbb{G}_2$ is a subgroup (that is, $\mathbb{G}_2$ is a multiplicative subgroup of $\mathbb{F}_{p^l}$).

6.2.1 Ordinary Signature

Let $h : \{0,1\}^* \rightarrow \mathbb{Z}_n^*$ (which is modeled as a random oracle). The message space is $\{0,1\}^*$ and the signature space is $\mathbb{Z}_n^*$.

Key Generation (CVKGS). The signer picks a random n and keeps the factorization secret. Then the signer picks a random public exponent $e \in \phi(n)$ such that $gcd(e, \phi(n)) = 1$ (where $\phi(n)$ is the Euler totient function), and computes the secret exponent d such that $ed \equiv 1 \bmod \phi(n)$. The public key is then (n, e) and the private key is d.

Signing (SigS). For a message $m \in \{0,1\}^*$, the signature is: $\sigma = h(m)^d \bmod n$.

Verification (VerS). Given a signature σ', to verify whether it is a valid signature of message m, check the following: $\sigma'^e \bmod n \stackrel{?}{=} h(m)$.

6.2.2 Partial Signature

Given an RSA signature σ, we blind it by multiplying it with a random $a \in \mathbb{Z}_n^*$ and then use bilinear pairings to hide a. Again, a does not fit into $\mathbb{G}_2$. We need to use the same homomorphic mapping f as in the construction for ElGamal. Here we need an additional invertible mapping $f_1 : \mathbb{Z}_n^* \rightarrow \mathbb{Z}_p$. The implementation of f_1 is simple, which is: $f_1(x) = x \bmod p, \forall x \in \mathbb{Z}_n^*$ and the inverse is: $f_1^{-1}(y) = y \bmod n, \forall y \in \mathbb{Z}_p$. These mappings satisfy the following properties we need: $\forall x_1, x_2 \in \mathbb{Z}_n^*$, $f_1(x_1 x_2) = f_1(x_1) f_1(x_2)$ and $f_1^{-1}(f_1(x_1) f_1(x_2)) = x_1 x_2$. These hold because $n^2 < p$.

If there are N witnesses specified in a partial signature δ, then $\delta \in \{0,1\}^k \times \mathbb{F}_{p^l} \times \mathbb{G}_1^N$. To avoid possible distinction due to different modulus size, padding is used to extend each computed signature to some arbitrary length k by adding a random multiple of n and padding zero's to the left.

Blinding (CVSig). Given a signing exponent d, a message m, a verifiability condition set $C = \{(c_i, W_i) : 1 \leq i \leq N\}$ and the witness public keys $\{(P_i, Y_i) : 1 \leq i \leq N\}$, we create a partial signature δ as follows:

1. Run SigS on m to generate an ordinary signature $\sigma \in \mathbb{Z}_n^*$.
2. Flip a coin $b \in \{0,1\}$. Randomly pick $a \in \mathbb{Z}_n^*$ such that: if $b = 0$, the Jacobi symbol $\left(\frac{a}{n}\right)$ must be 1, otherwise $\left(\frac{a}{n}\right) = -1$.[14]
3. Compute $\lambda = a\sigma \bmod n$. The resulting λ could have a Jacobi symbol of 1 or -1, thus avoiding the distinction based on the Jacobi symbol. Extend λ to length k by adding a random multiple of n, that is, $\gamma = \lambda + xn$ where $x \in_R \left[0, \lfloor \frac{2^k - \lambda}{n} \rfloor\right]$.
4. Randomly pick $r \in \mathbb{Z}_q^*$, and compute $U_i = rG_i,\ 1 \leq i \leq N$.

[14] This could be easy as half of the elements in $\mathbb{Z}_n^*$ have Jacobi symbol 1 and the other half have Jacobi symbol -1.

5. Compute $z = f(a \bmod p) \prod_{i=1}^{N} \hat{e}(Y_i, H(c_i))^r = f(a \bmod p) \prod_{i=1}^{N} y_i^r = f(a \bmod p) \prod_{i=1}^{N} e_i^{x_i r}$.
6. The partial signature is then given by: $\delta = (\gamma, z, U_1, U_2, \ldots, U_N)$.

Signature Retrieval (Extract) . Given a partial signature $\delta' = (\gamma', z', U_1', U_2', \ldots, U_N')$, and a set of witness signatures $\sigma_i = x_i H(c_i)$, $1 \leq i \leq N$, with each being a short signature of W_i on a condition statement c_i, the signature retrieval process is as follows:

1. Compute the following: $a' = f^{-1}\left(\frac{z'}{\prod_{i=1}^{N} \hat{e}(U_i', \sigma_i)}\right) \bmod n$
2. Use the extended Euclidean algorithm to find $a'^{-1} \bmod n$.
3. The recovered ordinary signature is then given by: $\sigma' = \gamma' a'^{-1} \bmod n$.

Correctness of the Extraction. If the partial signature is properly formed, that is, the following set of equations holds for some unknown $a \in \mathbb{Z}_n^*$ and $r \in \mathbb{Z}_q^*$:

$$\begin{array}{l} z' = f(a \bmod p) \prod_{i=1}^{N} y_i^r = f(a \bmod p) \prod_{i=1}^{N} e_i^{x_i r}; \\ U_i' = rG_i, \ \forall i; \\ \gamma' = a\sigma + xn; \\ \sigma = h(m)^d \bmod n. \end{array} \tag{2}$$

Then the correctness of the ordinary signature retrieval is guaranteed as:

$\hat{e}(U_i', \sigma_i) = \hat{e}(rG_i, x_i H(c_i)) = \hat{e}(G_i, H(c_i))^{x_i r} = e_i^{x_i r}, \ \forall i$
$a' = f^{-1}\left(\frac{f(a \bmod p) \prod_{i=1}^{N} e_i^{x_i r}}{\prod_{i=1}^{N} \hat{e}(U_i', \sigma_i)}\right) \bmod n = f^{-1}\left(\frac{f(a \bmod p) \prod_{i=1}^{N} e_i^{x_i r}}{\prod_{i=1}^{N} e_i^{x_i r}}\right) \bmod n = f^{-1}(f(a \bmod p)) \bmod n$
$= a \bmod p \bmod n = a$
$\sigma' = \gamma' a'^{-1} \bmod n = (a\sigma + xn)a^{-1} \bmod n = \sigma \in \mathbb{Z}_n$
$\sigma'^e \bmod n = \sigma^e \bmod n = h(m)$.

Hence, $\mathsf{VerS}(m, \sigma', (n, e)) = 1$.

6.2.3 Partial Signature Simulator

A possible partial signature simulator Fake is as follows:

Fake(C)
Input: $C = \{(c_i, W_i) : 1 \leq i \leq N\}, \{(P_i, Y_i) : 1 \leq i \leq N\}, n$
Output: $\delta_f = (\gamma_f, z_f, V_1, V_2, \ldots, V_N)$

1. Randomly pick $\lambda_f \in \mathbb{Z}_n^*$ and compute $\gamma_f = \lambda_f + xn$ where $x \in_R \left[0, \lfloor \frac{2^k - \lambda_f}{n} \rfloor\right]$.
2. Randomly pick $r_f \in \mathbb{Z}_q$ and compute $V_i = r_f G_i$, $1 \leq i \leq N$.
3. Randomly pick $d \in \mathbb{Z}_n^*$ and compute $z_f = f(d \bmod p) \prod_{i=1}^{N} \hat{e}(Y_i, H(c_i))^{r_f} = f(d \bmod p) \prod_{i=1}^{N} y_i^{r_f} = f(d \bmod p) \prod_{i=1}^{N} e_i^{x_i r_f}$.
4. Output $(\gamma_f, z_f, V_1, V_2, \ldots, V_N)$.

This simulator only uses the verifiability condition set as input and the message. As the modulus of the signer is needed, it is difficult to tell if it is signer-anonymous.

Claim 23 *The RSA based CVS construction given above is simulatable with respect to the simulator* Fake *if the decisional bilinear Diffe-Hellmen problem is hard assuming H is a random oracle.*

Proof See Appendix F. ■

6.2.4 Confirmation Protocol

Let m be a message, $C = \{(c_i, W_i) : 1 \leq i \leq N\}$ be a verifiability condition set, $PK_S = (n, e)$ be a signer public key, and $PK_i = (G_i, Y_i)$, for $1 \leq i \leq N$, be the witness public key of W_i. Let also the partial signature be $\delta = (\gamma, z, U_1, U_2, \ldots, U_N)$.

Similar to the case for the ElGamal signature scheme, in the confirmation protocol, the signer just needs to convince the recipient that Equation 2 holds for a given tuple $(m, C, PK_S, PK_i, \delta)$ using his private input (r, a). The confirmation protocol $\mathsf{CVCon}_{(S,V)}$ is then as follows.

$\langle \mathsf{CVConS}(r, a), \mathsf{CVConV}() \rangle (m, C, PK_S, PK_i, \delta)$

Common Input: $m; C = \{(c_i, W_i) : 1 \leq i \leq N\}; PK_S = (n, e); PK_i = (G_i, Y_i), 1 \leq i \leq N; \delta = (\gamma, z, U_1, U_2, \ldots, U_N)$

Signer Private Input: r, a

Protocol: The recipient first checks whether $\left(\frac{\gamma}{n}\right) \stackrel{?}{=} 0$ and proceeds if and only if it is not zero. This is necessary to ensure that a^{-1} exists to recover σ. Shown below is just one round of iteration, which should be run multiple rounds say k in the actual protocol. Recall that $e_i = \hat{e}(P_i, H(c_i))$, $y_i = \hat{e}(Y_i, H(c_i)) = e_i^{x_i}$, $w_i = \hat{e}(U_i, H(c_i)) = e_i^r$

1. **Commit.** The signer randomly picks $u \in \mathbb{Z}_q^*$ and $v \in \mathbb{Z}_n^*$, computes and sends the following to the verifier:

$$s = v^e \bmod n; \qquad t_i = e_i^u,\ 1 \leq i \leq N; \qquad t = f(v \bmod p) \prod_{i=1}^{N} y_i^u$$

2. **Challenge.** The verifier uniformly picks $b \in_R \{0, 1\}$ and sends it to the signer.
3. **Response.** The signer sends back $\theta = u + br$, $\psi = (a \bmod n)^b (v \bmod n) \bmod p$. Note that $\psi \bmod n = a^b v \bmod n$.
4. **Verify.** The verifier then checks the validity of the following and accepts only if:

$$s(\gamma^e)^b \stackrel{?}{\equiv} h(m)^b (\psi \bmod n)^e \pmod n; \qquad t_i \stackrel{?}{=} e_i^\theta w_i^{-b}; \qquad f(\psi) \prod_{i=1}^{N} y_i^\theta \stackrel{?}{=} z^b t.$$

Claim 24 *The above confirmation protocol for the RSA based CVS construction is a zero-knowledge proof.*

Proof The above protocol satisfies the completeness, soundness and zero-knowledge properties. See Appendix F. ■

The above construction for RSA applies to GHR [22] signatures.

6.2.5 Security of the RSA based CVS Construction

Putting the pieces together, we could conclude that the RSA based CVS scheme is secure satisfying the properties of unforgeability (Theorem 20) and simulatability (Claim 23), which in turn imply the cheat-immunity property. Besides, its confirmation protocol is zero knowledge (Claim 24) which, together with the simulatability property, further implies that the RSA based CVS constrcution is non-transferable.

7 Real World Applications of CVS

In this section, we give details about some possible application scenarios of CVS, including post-dated cheques, electronic commerce and policy-based access control.

7.1 Post-dated Cheques

Based on the CVS model, it is fairly straightforward to give an implementation like post-dated cheques, which incorporate time into a digital signature to control when its validity could be verified and when a document becomes effective. A distinctive difference between a real world post-dated cheque and the CVS implementation is that anyone could see the instructions put down by the signer and verify the validity of the cheque in the real world whereas nobody could assure or convince others that a given CVS-based post-dated cheque is valid or that the instructions shown are really what the signer endorsed. Hence, the CVS post-dated cheque has the additional advantage of protecting the privacy of the signer and his anonymity in some cases (depending on the CVS construction). This is one of the most desired properties in the commercial world, bespoken by the future/option trading scenario mentioned earlier.

In the post-dated cheque application, the partial signature generation is no different from that in a usual CVS scheme, except there is only one verifiability condition of the form (T, W_{time}) where T is a string specifying the release time and W_{time} is the trusted time server (a witness of time) specified by the signer. T could simply be the statement "It is now 2:00PM GMT Dec 23, 2000". All other processes are the same as in CVS but the delivery of the witness signatures is different. Instead of requiring the recipient of a partial signature to request the time server for its witness signature, the time server is set up to periodically broadcast its signature on a condition statement about the current time, and this statement could be "It is now 2:00PM GMT Dec 23, 2000", etc.. The broadcast period could be tuned down to whatever granularity appropriate for the desired applications. The advantages of this model include scalability, anonymity of both the signer and recipient, and the privacy of the message with respect to the time server. It is highly scalable because no matter how many users are supported, a single broadcast at each time instant is sufficient.

7.2 Electronic Commerce

It is a natural problem in electronic commerce to ask how a customer can ensure that an online trader can get his payment, possibly a signature for payment authorization, only when the trader has delivered his order or completed the services in the deal. Looking from the trader's perspective, he also wants to have some guarantee that he can receive the customer's payment before delivering the order. As mentioned before, CVS could partially solve the deadlock by using witnesses which are the parties involved in the workflow of processing the order. We could view CVS as filling the trust gap between traders and customers without physical proximity. In details, the customer could just pick a number of third parties that he trusts and will be involved in processing his order as witnesses to create a CVS partial signature

with conditions specifying that the parts of order processing involving these witnesses are completed by the trader. Due to the non-verifiability of the partial signature, nobody could verify the validity of payment authorization, thus preventing the trader from getting any payment unless he has obtained all the endorsements from the specified witnesses, which in turn requires him to somehow complete processing the order.

A typical example about how CVS helps in trading between mistrusting parties is as follows: A customer wishes to buy a durable item from an online trader but the price is so low that he is concerned about possible fraud. As usual, the trader needs the customer to pay before delivering the order. Some cautious customers may just walk away, unnecessarily ruining a deal he wants. In this scenario, the CVS could possibly help to narrow this trust gap. The customer could give the trader a CVS partial signature on his payment authorization and require him to get a signed (digital) receipt from the post office or courier company detailing about what they receive from the trader for delivering to the customer in order to retrieve his ordinary signature. Note that the trader would only be able to obtain the receipt if he has sent out the order; of course, the post office or the courier company needs to be trusted in checking the order, which we believe should be a reasonable assumption. If the trader has not sent out the order, the partial signature will not grant him any payment.

Although fair exchange can be useful in exchanging digital goods, it could not solve the above scenario satisfactorily. In the fair exchange solution, the customer signs his payment authorization to create some number which cannot be verified but can be converted into an ordinary signature by a designated trusted third party, acting as an arbitrator. The customer then convinces the trader that this partial signature can allow the trader to obtain his signature from the arbitrator even though he does not collaborate later on. If the trader has delivered the order, the customer will give him his signature. In case the customer does not give out his signature after the order has been delivered, the trader could access the arbitrator with all the evidences of order delivery, asking him to convert/retrieve the customer's signature. This approach still has the drawback of compromising the trader's privacy when a dispute arises in trading non-regenerable goods. In order to make a fair arbitration when the customer repudiates, the arbitrator usually requests the trader to submit evidences revealing much more than information about the deal; in some cases, this may hinder the trader to initiate the arbitration process and the fairness may not be achieved as stated.

When used for trading regenerable goods, CVS still has the advantage that an ordinary signature could be retrieved from a partial signature spontaneously without the help of the signer (customer) once all the specified conditions are fulfilled. In some cases, when there is a time lag between when an order is placed and when it is completed, this advantage of CVS would manifest itself. Airline or hotel reservations are just some examples.

Despite the need of trusted third parties in the CVS model, they are not special for arbitration but inevitable in processing the order. Although they already know some information about the order, they learn no information about the deal. The customer could assign witnesses without needing to notify them. Besides, the trader does not need to leak out any information about the payment in order to get these witnesses to help him retrieve the signature, and requesting a witness to sign on a condition, in the form of a receipt, seems to be natural in the business world. However, there are still a number of problems that cannot be solved using the CVS model such as fairly exchanging signed contracts.

7.3 Policy-Based Access Control

The CVS model could also be used for monitored controls of accessing resources. Suppose the president of a certain nation wishes to grant one of his aides access to a certain highly confidential resources or files

for which that aide is not entitled unless it is an emergency as certified by a certain number of cabinet members. Note that there is an implicit implication that the president would be absent for some reasons when such a certificate becomes effective. Obviously, this can be done using the CVS model with the president creating a partial signature on the access control certificate. Very complex access policies could also be implemented using the CVS model.

The advantage of using the CVS-based approach for access control is two-fold. First, it could avoid the abuse of the signed access control certificate by the holder. Second, the non-verifiability of the certificate could minimize the potential risk of coercion on its holder. For example, in the scenario mentioned above, a curious aide would not be able to abuse the certificate to access the resource in question unless he colludes with all the cabinet members specified as witnesses. On the other hand, if the aide is kidnapped, the enemies would still not be able to determine whether the aide holds a valid certificate.

8 Conclusions

In this paper, we introduce a new signature concept called CVS which could provide effective solutions in many digital business scenarios, in particular, those involving mutually distrusting parties. Through CVS, one could limit and control the verifiability of his digital signatures subject to the fulfilled of a number of conditions he specifies. We also give two efficient CVS constructions based on bilinear pairings for the standard signature schemes of ElGamal and RSA.

In future work, we plan to add the function of traceability to the CVS scheme. In details, in the current schemes, once the ordinary signature is extracted, nobody could tell whether it is generated directly or extracted from a partial signature. This is the perfect convertibility property, which is basically good. But in some scenarios in which the recipient may be able to corrupt all the witnesses, the signer may want to have a certain trapdoor to allow him to prove to others, say a court judge, whether a given signature is signed directly or recovered from partial signature using the signatures of the witnesses on a number of condition statements. That is, a recovered signature is normally indistinguishable from an ordinary signature signed directly, but when the signer release a trapdoor, everyone would be convinced that a recovered signature is one extracted from a partial signature by the witness endorsements. Consequently the witnesses are held accountable.

References

[1] N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communication*, 18(4), April 2000.

[2] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *Advances in Cryptology — Crypto'2001, Springer-Verlag LNCS vol. 2139*, pages 213–229, 2001.

[3] D. Boneh, B. Lynn, and H. Shacham. Short signatures from Weil pairing. In *Advances in Cryptology — Asiacrypt'2001, Springer-Verlag LNCS vol. 2248*, pages 514–532, 2001.

[4] J. Boyar, D. Chaum, I. Damgård, and T. Pedersen. Convertible undeniable signatures. In *Advances in Cryptology — Crypto'90, Springer-Verlag LNCS vol. 537*, pages 189–205, 1991.

[5] J. Camenisch and I. Damgård. Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. In *Advances in Cryptology — Asiacrypt'2000, Springer-Verlag LNCS vol. 1976*, pages 331–345, 2000.

[6] J. Camenisch and M. Michels. Confirmer signature schemes secure against adaptive adversaries. In *Advances in Cryptology — Eurocrypt'00, Springer-Verlag LNCS vol. 1870*, pages 243–258, 2000.

[7] D. Chaum. Zero-knowledge undeniable signatures. In *Advances in Cryptology — Eurocrypt'90, Springer-Verlag LNCS vol. 473*, pages 458–464, 1990.

[8] D. Chaum. Designated confirmer signatures. In *Advances in Cryptology — Eurocrypt'94, Springer-Verlag LNCS vol. 950*, pages 86–91, 1995.

[9] D. Chaum and H. van Antwerpen. Undeniable signatures. In *Advances in Cryptology — Crypto'89, Springer-Verlag LNCS vol. 435*, pages 212–216, 1989.

[10] D. Chaum, H. van Antwerpen, and B. Pfitzmann. Cryptographically strong undeniable signatures, unconditionally secure for the signer. In *Advances in Cryptology — Crypto'91, Springer-Verlag LNCS vol. 576*, pages 470–484, 1992.

[11] L. Chen, C Kudla, and K. G. Paterson. Concurrent signatures. In *Advances in Cryptology — Eurocrypt'2004, Springer-Verlag LNCS vol. 3027*, pages 287–305, 2004.

[12] I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *Advances in Cryptology — Eurocrypt'00, Springer-Verlag LNCS vol. 1807*, pages 418–430, 2000.

[13] I. Damgård and T. Pedersen. New convertible undeniable signature schemes. In *Advances in Cryptology — Eurocrypt'96, Springer-Verlag LNCS vol. 1070*, pages 372–386, 1996.

[14] I. Damgård, B. Piftzmann, and T. Pedersen. Statistical secrecy and multi-bit commitments. *IEEE Transaction on Information Theory*, 44:1143–1151, 1998.

[15] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transaction on Information Theory*, IT-30(4):469–472, July 1985.

[16] M. Franklin and M. Reiter. Verifiable signature sharing. In *Advances in Cryptology — Eurocrypt'95, Springer-Verlag LNCS vol. 921*, pages 50–63, 1995.

[17] S. Galbraith and W. Mao. Invisibility and anonymity of undeniable and confirmer signatures. In *Cryptographers' Track RSA Conference (CT-RSA 2003), Springer-Verlag LNCS vol. 2612*, pages 80–97, 2003.

[18] S. D. Galbraith, W. Mao, and K. G. Paterson. RSA-based undeniable signatures for general moduli. In *Cryptographers' Track RSA Conference (CT-RSA 2002), Springer-Verlag LNCS vol. 2271*, pages 200–217, 2002.

[19] J. Garay and M. Jakobsson. Timed release of standard digital signatures. In *Financial Crypto'2002, Springer-Verlag LNCS vol.*, 2002.

[20] J. Garay and C. Pomerance. Timed fair exchange of standard signatures. In *Financial Crypto'2003, Springer-Verlag LNCS vol.*, 2003.

[21] R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signatures without the random oracle. In *Advances in Cryptology — Eurocrypt'99, Springer-Verlag LNCS vol.*, pages 123–139, 1999.

[22] R. Gennaro, H. Krawczyk, and T. Rabin. RSA-based undeniable signatures. In *Advances in Cryptology — Crypto'97, Springer-Verlag LNCS vol. 1294*, pages 397–416, 1997.

[23] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of ACM*, 33(4):792–807, 1986.

[24] O. Goldreich, S. Micali, and A. Wigderson. How to prove all NP-statements in zero-knowledge, and a methodolgy of cryptographic protocol design. In *Advances in Cryptology — Crypto'86, Springer-Verlag LNCS vol.*, pages 171–185, 1986.

[25] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[26] S. Goldwasser, S. Micali, and R. Rivest. A secure signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[27] S. Goldwasser and E. Waisbard. Transformation of digital signature schemes into designated confirmer signature schemes. In *TCC2004, Springer-Verlag LNCS vol. 2951*, pages 77–100, 2004.

[28] I. Impagliazzo, L. Levin, and M. Luby. Pseudo-random generation from one-way functions. In *ACM Symposium on Theory of Computing (STOC 1989*, 1989.

[29] M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In *Advances in Cryptology — Eurocrypt'96, Springer-Verlag LNCS vol. 1070*, pages 143–154, 1996.

[30] B. Libert and J. J. Quisquater. Identity based undeniable signatures. In *Cryptographers' Track RSA Conference (CT-RSA 2004), Springer-Verlag LNCS vol. 2964*, pages 112–125, 2004.

[31] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *CRC Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1996.

[32] M. Michels and M. Stadler. Efficient convertible undeniable signature schemes. In *SAC97, Springer-Verlag LNCS vol.*, pages 231–244, 1997.

[33] M. Michels and M. Stadler. Generic constructions for secure and efficient confirmer signature schemes. In *Advances in Cryptology — Eurocrypt'98, Springer-Verlag LNCS vol. 1403*, pages 402–421, 1998.

[34] M. Naor. Bit commitment using pseudo-randomness. In *Advances in Cryptology — Crypto'89, Springer-Verlag LNCS vol. 435*, pages 128–136, 1990.

[35] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *ACM Symposium on Theory of Computing (STOC 89*, pages 33–43, 1989.

[36] T. Okamoto. Designated confirmer signatures and public-key encryption are equivalent. In *Advances in Cryptology — Crypto'94, Springer-Verlag LNCS vol. 839*, pages 61–74, 1994.

[37] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of ACM*, 21(2):120–126, February 1978.

[38] J. Rompel. One-way functions are necessary and sufficient for secure signature. In *Proceedings 22ndACM Symposium on Theory of Computing (STOC 90)*, pages 387–394, 1990.

[39] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[40] M. Stadler. Publicly verifiable secret sharing. In *Advances in Cryptology — Eurocrypt'96, Springer-Verlag LNCS vol.*, pages 190–199, 1996.

[41] W. Susilo, F. Zhang, and Y. Mu. Identity-based strong designated verifier signature schemes. In *ACISP2004, Springer-Verlag LNCS vol. 3108*, pages 313–324, 2004.

Appendix A: Proofs — Relations between Security Notions

Proof related to the Confirmation Protocol

Lemma 3. Given two ensembles of distribution $\{X_\lambda\}$ and $\{Y_\lambda\}$, which have the same sample space for all λ, and a PPT algorithm T_λ (a transcript simulator) whose input space is the same as that of X_λ and Y_λ, let $\pi(x)$ denote the output of T_λ on input x.[15] If $\{X_\lambda\} \cong \{Y_\lambda\}$ in the security parameter λ, then

$$\{x \leftarrow X_\lambda; \pi(x) \leftarrow \{T_\lambda(x)\} : (x, \pi(x))\} \cong \{y \leftarrow Y_\lambda; \pi(y) \leftarrow \{T_\lambda(y)\} : (y, \pi(y))\}$$

Proof of Lemma 3.

We prove this lemma by contradiction. Suppose $\{X_\lambda\}$ and $\{Y_\lambda\}$ are indistinguishable with negligible indistinguishability coefficient $\epsilon_{X,Y}$, that is, for all PPT A,

$$|Pr[x \leftarrow X_\lambda : A(x) = 1] - Pr[y \leftarrow Y_\lambda : A(y) = 1]| \leq \epsilon_{X,Y}(\lambda) < \frac{1}{poly(\lambda)}.$$

Assume there is a PPT distinguisher $\mathcal{D}$ which can tell apart the two distributions: $\{x \leftarrow X_\lambda : (x, \pi(x))\}$ and $\{y \leftarrow Y_\lambda : (y, \pi(y))\}$. That is, the following is non-negligible.

$$\epsilon_{\mathcal{D}}(\lambda) = \left| \begin{array}{l} Pr[x \leftarrow X_\lambda; \pi(x) \leftarrow \{T_\lambda(x)\} : \mathcal{D}(x, \pi(x)) = 1] \\ -Pr[y \leftarrow Y_\lambda; \pi(y) \leftarrow \{T_\lambda(y)\} : \mathcal{D}(y, \pi(y)) = 1] \end{array} \right|$$

We show how to use $\mathcal{D}$ to construct $\mathcal{D}'$ to tell whether a given δ belongs to X_λ or Y_λ. The construction is as follows:

$\mathcal{D}'(\delta)$ where $\delta \leftarrow X_\lambda$ when $b = 0$ and $\delta \leftarrow Y_\lambda$ when $b = 1$
Run T_λ to generate the transcript $\pi(\delta)$ for δ.
Run $\mathcal{D}$ on $(\delta, \pi(\delta))$.
Output $\mathcal{D}$'s guess b' for b.

If $\mathcal{D}$ and T_λ are PPT, then so is $\mathcal{D}'$. Obviously,

$$Pr[x \leftarrow X_\lambda : \mathcal{D}'(x) = 1] = Pr[x \leftarrow X_\lambda; \pi(x) \leftarrow \{T_\lambda(x)\} : \mathcal{D}(x, \pi(x)) = 1], \text{ and}$$
$$Pr[y \leftarrow Y_\lambda : \mathcal{D}'(y) = 1] = Pr[y \leftarrow Y_\lambda; \pi(y) \leftarrow \{T_\lambda(y)\} : \mathcal{D}(y, \pi(y)) = 1].$$

Substituting these two equations into the expression of $\epsilon_{\mathcal{D}}(\lambda)$, then,

$$\epsilon_{\mathcal{D}}(\lambda) = |Pr[x \leftarrow X_\lambda : \mathcal{D}'(x) = 1] - Pr[y \leftarrow Y_\lambda : \mathcal{D}'(y) = 1]| \leq \epsilon_{X,Y}(\lambda)$$

This concludes the reduction: $\epsilon_{\mathcal{D}}(\lambda)$ must be negligible, otherwise $\epsilon_{X,Y}(\lambda)$ is non-negligible (a contradiction). That is,

$$\{X_\lambda\} \cong \{Y_\lambda\} \Rightarrow \{x \leftarrow X_\lambda; \pi(x) \leftarrow \{T_\lambda(x)\} : (x, \pi(x))\} \cong \{y \leftarrow Y_\lambda; \pi(y) \leftarrow \{T_\lambda(y)\} : (y, \pi(y))\}.$$

■

[15] Note that T_λ is probabilistic, so even for the same input x, $T_\lambda(x)$ may be different between two evaluations.

Proof: Simulatability and Unforgeability imply Cheat-immunity

Theorem 8. An unforgeable and simulatable CVS scheme is also cheat-immune given its confirmation protocol is zero knowledge.

Proof of Theorem 8.

Assume the given CVS scheme is unforgeable and simulatable with respect to a PPT simulator $\mathsf{Fake}_S(m, C)$. Let $\mathsf{SimT}(\delta)$ be the transcript simulator of the zero knowledge proof used for the confirmation protocol where δ is a partial signature.

In the cheat-immunity game defined in the paper, an adversary is always given a valid partial signature as a challenge. In the following proof, we force an adversary, capable to win the cheat-immunity game with non-negligible probability, to run on a challenge which is not a valid partial signature but a fake one from the simulator Fake. Since the adversary is just an algorithm, it is thus definitely possible to run it on a deviated input. Of course, it is likely that the adversary would not output the desired result on the deviated input, but this is what we want to show.

In order to run the adversary on a deviated input, we modify the definition of the cheat-immunity game slightly, namely, in the challenge phase, no confirmation protocol would be run between the challenger and the adversary, but instead the adversary is given a challenged partial signature and a transcript of a confirmation protocol run on that partial signature. Note that a run of the interactive confirmation protocol is replaced by a transcript without any interaction. We argue that the proof obtained in this amended model also applies to the original model of cheat-immunity if the confirmation protocol is zero knowledge. The justification is as follows:

> If the confirmation protocol of a CVS scheme is zero-knowledge, the only information obtainable from running the confirmation protocol is whether a given partial signature is true/valid. Hence, the only difference between the information obtainable from a given partial signature and the transcript recorded during the confirmation protocol run on it and the information obtainable from a given partial signature and a simulated transcript of the confirmation protocol is the validity of the given partial signature and nothing else. In other words, if an adversary can extract the ordinary signature from a valid partial signature after running the confirmation protocol on it, it should also be able to do so with almost the same computational effort even without running the confirmation protocol. Consequently, we would neglect running the confirmation protocol in the challenge phase to force as adversary to run on an invalid partial signature. In fact, if we insist on running the confirmation protocol between the adversary and the challenger in the challenge phase, it is still possible (even though inefficient) using the rewinding technique commonly found in the transcript simulator of any zero knowledge proof, as it is used in [28]. In order to make an adversary accept a partial signature input and run on it, in each round of iteration of the confirmation protocol, we prepare the answer of some of all the possible challenged questions. If the challenge question comes out to be what has been prepared, then this round is successful; otherwise, we reset the adversary to the start of the current iteration round and restart this round again. As mentioned before, this rewinding is possible because the adversary is just another algorithm or Turing machine we use as a subroutine. Of course, we have to take more computations to complete an iteration round now but in most zero knowledge proofs, the overall computation would still remain polynomial time.

Now we can describe the proof. Suppose there exists a PPT adversary A which can win the cheat-immunity game with non-negligible probability p_A^{CI}. We show how to construct a distinguisher D from

A for the simulatability game, which can distinguish a true partial signature (CVSig) from a fake one generated by Fake.

$D(\delta_b)$: δ_b is a true/fake partial signature when $b = 0/1$

Setup.
Get from its challenger the public keys of the signer and witnesses, and pass them to A.
Run A on the same set of public keys.
Keep the signer private key if given one.
Query.
Pass all signing and endorsement queries from A to its oracles and return the results to A.
For signing queries, run the confirmation protocol as an agent in between A and the challenger.
Challenge.
A outputs (m, C), $m \in \mathcal{M}$, $C \subset \mathcal{C} \times \mathcal{W}$, to be challenged.
Pass (m, C) to its challenger and receive the challenge δ_b.
Compute the confirmation transcript $\pi(\delta_b) = \mathsf{SimT}(\delta_b)$ for δ_b.
Pass $(\delta_b, \pi(\delta_b))$ as a challenge to A.
Guess.
A outputs σ. Output guess b' where:

$$b' = \begin{cases} 0, & \mathsf{VerS}(m, \sigma) = 1 \\ 1, & otherwise \end{cases}$$

First, it can be seen that D is PPT if A and VerS are both PPT.

The probability of success of D with respect to the simulatability game is:[16]

$$\begin{aligned} Pr_D^{Sim}[success] &= Pr[b' = b|\delta_b] \\ &= \tfrac{1}{2}Pr[b' = 0|\delta_0] + \tfrac{1}{2}Pr[b' = 1|\delta_1] \\ &= \tfrac{1}{2}Pr[\delta_0 \leftarrow \{\mathsf{CVSig}_S(m, C)\}; \sigma \leftarrow \{A(\delta_0)\} : \mathsf{VerS}(m, \sigma) = 1] \\ &\quad + \tfrac{1}{2}Pr[\delta_1 \leftarrow \{\mathsf{Fake}_S(m, C)\}; \sigma \leftarrow \{A(\delta_1)\} : \mathsf{VerS}(m, \sigma) = 0] \\ &= \tfrac{1}{2}p_A^{CI} + \tfrac{1}{2} - \tfrac{1}{2}Pr[\delta_1 \leftarrow \{\mathsf{Fake}_S(m, C)\}; \sigma \leftarrow \{A(\delta_1)\} : \mathsf{VerS}(m, \sigma) = 1]. \end{aligned}$$

Note we use the fact: $p_A^{CI} = Pr[\delta_0 \leftarrow \{\mathsf{CVSig}_S(m, C)\}; \sigma \leftarrow \{A(\delta_0)\} : \mathsf{VerS}(m, \sigma) = 1]$. Rearranging terms, we have:

$$\tfrac{1}{2}p_A^{CI} = \left(Pr_D^{Sim}[success] - \tfrac{1}{2}\right) + \tfrac{1}{2}Pr[\delta_1 \leftarrow \{\mathsf{Fake}_S(m, C)\}; \sigma \leftarrow \{A(\delta_1)\} : \mathsf{VerS}(m, \sigma) = 1]$$

[16] For the sake of simple notations, we tend to use short notations for the probability in question. For example, we just write $Pr[b' = b|\delta_b]$ to denote the probability that the guess of D, that is, b' is the same as the challenged bit b given δ_b which could be generated from CVSig (if $b = 0$) or Fake (if $b = 1$). We also neglect the preamble like public key generation. Formally, this probability should be written as:

$$Pr\left[\begin{array}{l} (PK_S, sk_S) \leftarrow \{CVKGS(1^\lambda)\}; (PK_W, sk_W) \leftarrow \{CVKGW(1^\lambda), \forall W; \\ m \leftarrow \mathcal{M}; C \leftarrow 2^{\mathcal{C}\times\mathcal{W}}; b \leftarrow \{0,1\}; \delta_b \leftarrow \left\{\begin{array}{ll} \{\mathsf{CVSig}_S(m, C)\}, & b = 0 \\ \{\mathsf{Fake}_S(m, C)\}, & b = 1 \end{array}\right. ; \quad : \quad b' = b \\ \sigma \leftarrow \{A(\delta_b)\}; b' = \neg(\mathsf{VerS}(m, \sigma) = 1)\}) \end{array}\right]$$

Taking absolute values on both sides and denoting $Pr[\delta_1 \leftarrow \{\mathsf{Fake}_S(m,C)\}; \sigma \leftarrow \{A(\delta_1)\} : \mathsf{VerS}(m,\sigma) = 1]$ by ε_f, we have:

$$\begin{aligned} \tfrac{1}{2}p_A^{CI} &\leq \left|Pr_D^{Sim}[success] - \tfrac{1}{2}\right| + \left|\tfrac{1}{2}Pr[\delta_1 \leftarrow \{\mathsf{Fake}_S(m,C)\}; \sigma \leftarrow \{A(\delta_1)\} : \mathsf{VerS}(m,\sigma) = 1]\right| \\ &= Adv_D^{Sim} + \tfrac{1}{2}Pr[\delta_1 \leftarrow \{\mathsf{Fake}_S(m,C)\}; \sigma \leftarrow \{A(\delta_1)\} : \mathsf{VerS}(m,\sigma) = 1] \end{aligned}$$

$$p_A^{CI} \leq 2Adv_D^{Sim} + \varepsilon_f. \tag{3}$$

If p_A^{CI} is non-negligible, then either Adv_D^{Sim} or ε_f is non-negligible. We consider the following two cases:

Case 1 — Adv_D^{Sim} is non-negligible. Obviously, the existence of such a PPT algorithm D would break the simulatability property, which is a contradiction as we assume the CVS scheme is simulatable.

Case 2 — ε_f is non-negligible. We argue that if ε_f is non-negligible, then we could use A to create an existential forgery as follows.

F
Setup.
Get all the public keys of the signer and witnesses.
Run A on the same set of public keys.
Keep the witness private keys.[17]
Query.
Pass all signing queries to its oracle and relay the results back to A.
Run the confirmation protocol as an agent in between A and the challenger.
Answer all endorsement queries itself using the witness private keys.
Challenge.
A outputs (m, C), $m \in \mathcal{M}$, $C \subset \mathcal{C} \times \mathcal{W}$, to be challenged.
Create $\delta = \mathsf{Fake}_S(m, C)$, and compute the confirmation transcript $\pi(\delta) = \mathsf{SimT}(\delta)$ for δ.
Pass $(\delta_b, \pi(\delta_b))$ as a challenge to A.
Guess.
Output the final output σ of A as a forgery output.

Obviously, if A is PPT, then F is also PPT as Fake is PPT. As m is chosen to be not queried before, the probability of successful existential forgery by F is then given by:

$$p_F^{UF} = Pr[\delta \leftarrow \{\mathsf{Fake}_S(m,C)\}; \sigma \leftarrow \{A(\delta)\} : \mathsf{VerS}(m,\sigma) = 1]$$

Note that p_F^{UF} should be equal to ε_f which is non-negligible. This concludes that the given CVS scheme is existentially forgeable if ε_f is non-negligible, which is a contradiction as we assume the CVS scheme is unforgeable.

In conclusion, if the given CVS scheme is simulatable (i.e. $Adv_{\mathcal{D}}^{Sim}$ is negligible for all PPT $\mathcal{D}$) and unforgeable (i.e. $p_{\mathcal{F}}^{UF}$ is negligible for all PPT $\mathcal{F}$), then it is also cheat-immune with negligible $p_{\mathcal{A}}^{CI}$ for all PPT $\mathcal{A}$. ■

Equivalence between Simulatability, Invisibility and Anonymity

Theorem 9 (Simulatability implies Invisibility). Given a simulatable CVS scheme in an adaptive query model with respect to a PPT fake signature simulator $\mathsf{Fake}_S(m, C)$, it is message-invisible in the same adaptive query model if and only if $\{\mathsf{Fake}_S(m_0, C)\} \cong \{\mathsf{Fake}_S(m_1, C)\}$ in the same adaptive query model for all S, m_0, m_1, and C.

Proof of Theorem 9.

Assume the given CVS scheme is simulatable with respect to a PPT simulator Fake, that is, the corresponding Adv_D^{Sim} is negligible for all PPT D.

If Part

Suppose there exists a PPT distinguisher $\mathcal{D}$ which can break the invisibility property, that is, able to distinguish which one of the two given messages m_0 and m_1 a given partial signature δ is for. We show how to construct another distinguisher $\mathcal{D}'$ to tell whether a given δ_b is genuine from CVSig ($b = 0$) or fake from Fake ($b = 1$). In the following discussion, we denote the negation of b by $\overline{b}$.

$\mathcal{D}'(\delta_b)$: δ_b is a true/fake partial signature when $b = 0/1$

Setup.
Ask its challenger for the public keys of the signer and the witnesses
Run $\mathcal{D}$ on the same set of public keys.
Get the signer's private key from its challenger and pass it to $\mathcal{D}$.
Query.
Pass all signing and endorsement queries from $\mathcal{D}$ to its oracle.
Relay the results back to $\mathcal{D}$.
Run the confirmation protocol as an agent between $\mathcal{D}$ and the challenger.
Challenge.
$\mathcal{D}$ outputs (m_0, m_1, C) to be challenged.
Flip a coin $c \leftarrow \{0, 1\}$. Output (m_c, C) to its challenger.
Pass the challenge δ_b to $\mathcal{D}$.
Guess.
$\mathcal{D}$ outputs a guess b'. Output the final guess b'' for b:

$$b'' = \begin{cases} b', & c = 0 \\ \overline{b'}, & c = 1 \end{cases}$$

Obviously, if $\mathcal{D}$ is PPT, so is $\mathcal{D}'$.

Then the probability of success of $\mathcal{D}'$ with respect to simulatability is given by:

$$\begin{aligned}
Pr_{\mathcal{D}'}^{Sim}[Success] &= Pr[b'' = b|\delta_b] \\
&= \tfrac{1}{2}Pr[b' = b|\delta_b, c = 0] + \tfrac{1}{2}Pr[b' = \overline{b}|\delta_b, c = 1] \\
&= \tfrac{1}{4}Pr[b' = 0|\delta_0, c = 0] + \tfrac{1}{4}Pr[b' = 1|\delta_1, c = 0] \\
&\quad + \tfrac{1}{4}Pr[b' = 1|\delta_0, c = 1] + \tfrac{1}{4}Pr[b' = 0|\delta_1, c = 1] \\
&= \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_0, C)\} : \mathcal{D}(\delta) = 0] + \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_0, C)\} : \mathcal{D}(\delta) = 1] \\
&\quad + \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_1, C)\} : \mathcal{D}(\delta) = 1] + \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_1, C)\} : \mathcal{D}(\delta) = 0] \\
&= \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_0, C)\} : \mathcal{D}(\delta) = 0] + \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_1, C)\} : \mathcal{D}(\delta) = 1] \\
&\quad + \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_0, C)\} : \mathcal{D}(\delta) = 1] - \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_1, C)\} : \mathcal{D}(\delta) = 1] \\
&\quad + \tfrac{1}{4}.
\end{aligned}$$

Note we use in the above expression the fact:

$$Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_1, C)\} : \mathcal{D}(\delta) = 1] + Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_1, C)\} : \mathcal{D}(\delta) = 0] = 1$$

Note also that the probability of success of $\mathcal{D}$ with respect to invisibility is given by:

$$Pr^{Inv}_{\mathcal{D}}[Success] = \frac{1}{2}Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_0, C)\} : \mathcal{D}(\delta) = 0] + \frac{1}{2}Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_1, C)\} : \mathcal{D}(\delta) = 1].$$

Hence,

$$\begin{aligned}
\tfrac{1}{2}Pr^{Inv}_{\mathcal{D}}[Success] + \tfrac{1}{4} &= Pr^{Sim}_{\mathcal{D}'}[Success] + \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_1, C)\} : \mathcal{D}(\delta) = 1] \\
&\quad -\tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_0, C)\} : \mathcal{D}(\delta) = 1] \\
\tfrac{1}{2}\left(Pr^{Inv}_{\mathcal{D}}[Success] - \tfrac{1}{2}\right) &= \left(Pr^{Sim}_{\mathcal{D}'}[Success] - \tfrac{1}{2}\right) + \tfrac{1}{4}(Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_1, C)\} : \mathcal{D}(\delta) = 1] \\
&\quad -Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_0, C)\} : \mathcal{D}(\delta) = 1] \\
Adv^{Inv}_{\mathcal{D}} &\leq 2Adv^{Sim}_{\mathcal{D}'} + \tfrac{1}{2}\epsilon^{\mathsf{Fake}}_{m_0,m_1}
\end{aligned}$$

where $\epsilon^{\mathsf{Fake}}_{m_0,m_1} = |Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_1, C)\} : \mathcal{D}(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_0, C)\} : \mathcal{D}(\delta) = 1]|$. Therefore, if $Adv^{Inv}_{\mathcal{D}}$ is non-negligible, then either $Adv^{Sim}_{\mathcal{D}'}$ or $\epsilon^{\mathsf{Fake}}_{m_0,m_1}$ is non-negligible. The former condition implies the given CVS scheme is not simulatable (a contradiction) whereas the latter implies $\{\mathsf{Fake}_S(m_0, C)\} \ncong \{\mathsf{Fake}_S(m_1, C)\}$ (again a contradiction). Hence, $Adv^{Inv}_{\mathcal{D}}$ must be negligible if the given CVS scheme is simulatable with respect to Fake and $\{\mathsf{Fake}_S(m_0, C)\} \cong \{\mathsf{Fake}_S(m_1, C)\}$ in the same attack model.

Only if Part

The given CVS scheme is simulatable with respect to Fake implies indistinguishability between the following: $\{\mathsf{CVSig}_S(m, C)\} \cong \{\mathsf{Fake}_S(m, C)\}$, $\forall S, m, C$, that is, the following is negligible for all PPT D.

$$\epsilon^{Sim}_{D:\, m} = |Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m, C)\} : D(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{Fake}_S(m, C)\} : D(\delta) = 1]|$$

If the CVS scheme is also invisible, then for any two messages m_0 and m_1, $\{\mathsf{CVSig}_S(m_0, C)\} \cong \{\mathsf{CVSig}_S(m_1, C)\}$, $\forall S, C$, that is, the following is negligible for all PPT D.

$$\epsilon^{Inv}_{D:\,(m_0,m_1)} = |Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_0, C)\} : D(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_1, C)\} : D(\delta) = 1]|$$

For all S, C and any two messages m_0 and m_1, and for any PPT distinguisher $\mathcal{D}$,

$$\begin{aligned}
&Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_0, C)\} : \mathcal{D}(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_1, C)\} : \mathcal{D}(\delta) = 1] \\
&= Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_0, C)\} : \mathcal{D}(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_0, C)\} : \mathcal{D}(\delta) = 1] \\
&\quad +Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_1, C)\} : \mathcal{D}(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_1, C)\} : \mathcal{D}(\delta) = 1] \\
&\quad +Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_0, C)\} : \mathcal{D}(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_1, C)\} : \mathcal{D}(\delta) = 1]
\end{aligned}$$

If we denote $|Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_0, C)\} : \mathcal{D}(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_1, C)\} : \mathcal{D}(\delta) = 1]|$ by $\epsilon^{\mathsf{Fake}}_{\mathcal{D}:\,(m_0,m_1)}$ and take absolute values on both sides, then we have:

$$\begin{aligned}
\epsilon^{\mathsf{Fake}}_{\mathcal{D}:\,(m_0,m_1)} &\leq |Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_0, C)\} : \mathcal{D}(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_0, C)\} : \mathcal{D}(\delta) = 1]| \\
&\quad + |Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_1, C)\} : \mathcal{D}(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_1, C)\} : \mathcal{D}(\delta) = 1]| \\
&\quad + |Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_0, C)\} : \mathcal{D}(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m_1, C)\} : \mathcal{D}(\delta) = 1]| \\
&= \epsilon^{Sim}_{\mathcal{D}:\, m_0} + \epsilon^{Sim}_{\mathcal{D}:\, m_1} + \epsilon^{Inv}_{\mathcal{D}:\,(m_0,m_1)}
\end{aligned}$$

If $\epsilon^{\mathsf{Fake}}_{\mathcal{D}:\,(m_0,m_1)}$ is non-negligible, either one of the following is non-negligible: $\epsilon^{Sim}_{\mathcal{D}:\,m_0}$, $\epsilon^{Sim}_{\mathcal{D}:\,m_1}$, $\epsilon^{Inv}_{\mathcal{D}:\,(m_0,m_1)}$, which is a contradiction to either the simulatability or invisibility assumption. As a result, given a simulatable CVS scheme (with respect to Fake), if it is also invisible, then the following must be true for all S, m_0, m_1, C:

$$\{\mathsf{Fake}_S(m_0, C)\} \cong \{\mathsf{Fake}_S(m_1, C)\}$$

■

Theorem 10 (Simulatability implies Anonymity). Given a simulatable CVS scheme in an adaptive query model with respect to a PPT fake signature simulator $\mathsf{Fake}_S(m, C)$, it is signer-anonymous in the same adaptive query model if and only if $\{\mathsf{Fake}_{S0}(m, C)\} \cong \{\mathsf{Fake}_{S1}(m, C)\}$ in the same adaptive query model for all $S0$, $S1$, m, and C.

Proof of Theorem 10.

Assume the given CVS scheme is simulatable with respect to a PPT simulator Fake, that is, the corresponding Adv^{Sim}_D is negligible for all PPT D.

<u>If Part</u>

Suppose there exists a PPT distinguisher $\mathcal{D}$ which can break the anonymity property, that is, able to distinguish which one of the two given signers $S0$ and $S1$ has signed a given partial signature δ. We show how to construct another distinguisher $\mathcal{D}_c$ ($c \in \{0, 1\}$) to tell whether a given δ_b is genuine from CVSig (that is, $b = 0$) or fake from Fake (that is, $b = 1$). We give two constructions; in the following discussion, we use $c = 0$ and $c = 1$ to denote the difference between the two implementations of $\mathcal{D}_c$. In the following, we use $\overline{b'}$ and $\overline{c}$ to denote the negations of b' and c respectively (where $b', c \in \{0, 1\}$).

$\mathcal{D}_c(\delta_b)$: δ_b is a true/fake partial signature when $b = 0/1$

Setup.
Ask its challenger for the public keys of the witnesses.
Ask its challenger for the public and private keys of one signer, say Sc.
Run $\mathsf{CVKGS}(1^\lambda)$ to generate the public and private keys of the other signer $S\overline{c}$.
Run $\mathcal{D}$ on the public keys of Sc and $S\overline{c}$.
Pass all the witness public keys and the two signer private keys to $\mathcal{D}$.
Query.
Pass all Sc signing queries from $\mathcal{D}$ to its oracle. Relay the results back to $\mathcal{D}$.
Answer all $S\overline{c}$ signing queries from $\mathcal{D}$ by running $\mathsf{CVSig}_{S\overline{c}}$.
Pass all endorsement queries from $\mathcal{D}$ to its oracle and relay the results back to $\mathcal{D}$.
Challenge.
$\mathcal{D}$ outputs (m, C) to be challenged.
Output (m, C) to its challenger.
Pass the challenge δ_b to $\mathcal{D}$.
Guess.
$\mathcal{D}$ outputs a guess b'. Output the final guess b'' for b:

$$b'' = \begin{cases} b', & c = 0 \\ \overline{b'}, & c = 1 \end{cases}$$

Obviously, if $\mathcal{D}$ is PPT, so is $\mathcal{D}_c$ for both $c = 0$ and $c = 1$.

The probability of success of $\mathcal{D}_0$ with respect to simulatability is given by:

$$\begin{aligned} Pr^{Sim}_{\mathcal{D}_0}[Success] &= Pr[b'' = b|\delta_b] \\ &= \tfrac{1}{2}Pr[b' = 0|\delta_0] + \tfrac{1}{2}Pr[b' = 1|\delta_1] \\ &= \tfrac{1}{2}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m, C)\} : \mathcal{D}(\delta) = 0] + \tfrac{1}{2}Pr[\delta \leftarrow \{\mathsf{Fake}_{S0}(m, C)\} : \mathcal{D}(\delta) = 1]. \end{aligned}$$

The probability of success of $\mathcal{D}_1$ with respect to simulatability is given by:

$$\begin{aligned} Pr^{Sim}_{\mathcal{D}_1}[Success] &= Pr[b'' = b|\delta_b] \\ &= \tfrac{1}{2}Pr[b' = 1|\delta_0] + \tfrac{1}{2}Pr[b' = 0|\delta_1] \\ &= \tfrac{1}{2}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m, C)\} : \mathcal{D}(\delta) = 1] + \tfrac{1}{2}Pr[\delta \leftarrow \{\mathsf{Fake}_{S1}(m, C)\} : \mathcal{D}(\delta) = 0]. \end{aligned}$$

Note the probability of success of $\mathcal{D}$ with respect to anonymity is given by:

$$\begin{aligned} Pr^{Ano}_{\mathcal{D}}[Success] &= \tfrac{1}{2}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m, C)\} : \mathcal{D}(\delta) = 0] + \tfrac{1}{2}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m, C)\} : \mathcal{D}(\delta) = 1] \\ &= \tfrac{1}{2}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m, C)\} : \mathcal{D}(\delta) = 0] + \tfrac{1}{2}Pr[\delta \leftarrow \{\mathsf{Fake}_{S0}(m, C)\} : \mathcal{D}(\delta) = 1] \\ &\quad \tfrac{1}{2}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m, C)\} : \mathcal{D}(\delta) = 1] + \tfrac{1}{2}Pr[\delta \leftarrow \{\mathsf{Fake}_{S1}(m, C)\} : \mathcal{D}(\delta) = 0] \\ &\quad -\tfrac{1}{2}Pr[\delta \leftarrow \{\mathsf{Fake}_{S0}(m, C)\} : \mathcal{D}(\delta) = 1] - \tfrac{1}{2}Pr[\delta \leftarrow \{\mathsf{Fake}_{S1}(m, C)\} : \mathcal{D}(\delta) = 0] \\ &= Pr^{Sim}_{\mathcal{D}_0}[Success] + Pr^{Sim}_{\mathcal{D}_1}[Success] - \tfrac{1}{2} \\ &\quad -\tfrac{1}{2}Pr[\delta \leftarrow \{\mathsf{Fake}_{S0}(m, C)\} : \mathcal{D}(\delta) = 1] + \tfrac{1}{2}Pr[\delta \leftarrow \{\mathsf{Fake}_{S1}(m, C)\} : \mathcal{D}(\delta) = 1]. \end{aligned}$$

Note we substitute the values of $Pr^{Sim}_{\mathcal{D}_0}[Success]$ and $Pr^{Sim}_{\mathcal{D}_1}[Success]$ into the above equation and use the fact: $Pr[\delta \leftarrow \{\mathsf{Fake}_{S1}(m, C)\} : \mathcal{D}(\delta) = 1] + Pr[\delta \leftarrow \{\mathsf{Fake}_{S1}(m, C)\} : \mathcal{D}(\delta) = 0] = 1$

Denote $|Pr[\delta \leftarrow \{\mathsf{Fake}_{S1}(m, C)\} : \mathcal{D}(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{Fake}_{S0}(m, C)\} : \mathcal{D}(\delta) = 1]|$ by $\epsilon^{\mathsf{Fake}}_{\mathcal{D}:(S0,S1)}$. Note that $\mathcal{D}$ is set to distinguish between the genuine signatures of $S0$ and $S1$; nevertheless, if input with Fake, $\mathcal{D}$ must give an output. Subtracting $\frac{1}{2}$ from both sides and taking absolute values, we have:

$$Adv^{Ano}_{\mathcal{D}} \leq Adv^{Sim}_{\mathcal{D}_0} + Adv^{Sim}_{\mathcal{D}_1} + \tfrac{1}{2}\epsilon^{\mathsf{Fake}}_{\mathcal{D}:(S0,S1)}$$

Therefore, if $Adv^{Ano}_{\mathcal{D}}$ is non-negligible, then either $Adv^{Sim}_{\mathcal{D}_0}$, $Adv^{Sim}_{\mathcal{D}_1}$ or $\epsilon^{\mathsf{Fake}}_{m_0,m_1}$ is non-negligible. Either $Adv^{Sim}_{\mathcal{D}_0}$ or $Adv^{Sim}_{\mathcal{D}_1}$ is non-negligible implies the given CVS scheme is not simulatable (a contradiction). On the other hand, $\epsilon^{\mathsf{Fake}}_{m_0,m_1}$ is non-negligible implies $\{\mathsf{Fake}_{S0}(m, C)\} \ncong \{\mathsf{Fake}_{S1}(m, C)\}$ (again a contradiction with the given condition). Hence, $Adv^{Ano}_{\mathcal{D}}$ must be negligible if the given CVS scheme is simulatable with respect to Fake and $\{\mathsf{Fake}_{S0}(m, C)\} \cong \{\mathsf{Fake}_{S1}(m, C)\}$.

Only if Part

The given CVS scheme is simulatable with respect to Fake implies indistinguishability between the following: $\{\mathsf{CVSig}_S(m, C)\} \cong \{\mathsf{Fake}_S(m, C)\}$, $\forall S, m, C$, that is, the following is negligible for all PPT D.

$$\epsilon^{Sim}_{D:\,S} = |Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m, C)\} : D(\delta) = 1\}] - Pr[\delta \leftarrow \{\mathsf{Fake}_S(m, C)\} : D(\delta) = 1]|$$

If the CVS scheme is also anonymous, then for any two signers $S0$ and $S1$, $\{\mathsf{CVSig}_{S0}(m, C)\} \cong \{\mathsf{CVSig}_{S1}(m, C)\}$, $\forall m, C$, that is, the following is negligible for all PPT D.

$$\epsilon^{Ano}_{D:\,(S0,S1)} = |Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m, C)\} : D(\delta) = 1\}] - Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m, C)\} : D(\delta) = 1]|$$

For all m, C and any two signers $S0$ and $S1$, and for any PPT distinguisher $\mathcal{D}$,

$$\begin{aligned}
&Pr[\delta \leftarrow \{\mathsf{Fake}_{S0}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{Fake}_{S1}(m,C)\} : \mathcal{D}(\delta)=1] \\
=\ & Pr[\delta \leftarrow \{\mathsf{Fake}_{S0}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m,C)\} : \mathcal{D}(\delta)=1] \\
& + Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{Fake}_{S1}(m,C)\} : \mathcal{D}(\delta)=1] \\
& + Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m,C)\} : \mathcal{D}(\delta)=1]
\end{aligned}$$

If we denote $|Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_0,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{Fake}_S(m_1,C)\} : \mathcal{D}(\delta)=1]|$ by $\epsilon^{\mathsf{Fake}}_{\mathcal{D}:\,(m_0,m_1)}$ and take absolute values on both sides, then we have:

$$\begin{aligned}
\epsilon^{\mathsf{Fake}}_{\mathcal{D}:\,(S0,S1)} \leq\ & |Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{Fake}_{S0}(m,C)\} : \mathcal{D}(\delta)=1]| \\
& + |Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{Fake}_{S1}(m,C)\} : \mathcal{D}(\delta)=1]| \\
& + |Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m,C)\} : \mathcal{D}(\delta)=1]| \\
=\ & \epsilon^{Sim}_{\mathcal{D}:\,S0} + \epsilon^{Sim}_{\mathcal{D}:\,S1} + \epsilon^{Ano}_{\mathcal{D}:\,(S0,S1)}
\end{aligned}$$

If $\epsilon^{\mathsf{Fake}}_{\mathcal{D}:\,(S0,S1)}$ is non-negligible, either one of the following is non-negligible: $\epsilon^{Sim}_{\mathcal{D}:\,S0}$, $\epsilon^{Sim}_{\mathcal{D}:\,S1}$, $\epsilon^{Ano}_{\mathcal{D}:\,(S0,S1)}$, which is a contradiction to either the simulatability or anonymity assumption. As a result, given a simulatable CVS scheme (with respect to Fake), if it is also anonymous, then the following must be true for all S_0, S_1, m, C:

$$\{\mathsf{Fake}_{S0}(m,C)\} \cong \{\mathsf{Fake}_{S1}(m,C)\}$$

■

Theorem 12. Invisibility does not imply Simulatability.

Proof of Theorem 12.

In the following, we will show the necessary requirement for a given invisible CVS scheme to be simulatable.

If the given CVS scheme is invisible, then the following must hold: $\{\mathsf{CVSig}_S(m,C)\} \cong \{\mathsf{CVSig}_S(m',C)\}$ for all $m \neq m' S, C$. That is, the following is negligible for all PPT D.

$$\epsilon^{Inv}_{D:\,(m,m')} = \left|Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m,C)\} : D(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m',C)\} : D(\delta)=1]\right|$$

For all possible simulators Fake, the following must hold for all PPT distinguishers $\mathcal{D}$ for S, C, $m \neq m'$:

$$\begin{aligned}
\epsilon^{\mathsf{Fake}}_{\mathcal{D}} =\ & Pr[m,m' \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{CVSig}_S(m',C)\} : \mathcal{D}(\delta)=1] \\
& - Pr[m,m' \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{Fake}_S(m,C)\} : \mathcal{D}(\delta)=1] \\
=\ & Pr[m,m' \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{CVSig}_S(m',C)\} : \mathcal{D}(\delta)=1] \\
& - Pr[m,m' \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{CVSig}_S(m,C)\} : \mathcal{D}(\delta)=1] \\
& + Pr[m,m' \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{CVSig}_S(m,C)\} : \mathcal{D}(\delta)=1] \\
& - Pr[m,m' \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{Fake}_S(m,C)\} : \mathcal{D}(\delta)=1]
\end{aligned}$$

Taking absolute values on both sides, we get:

$$
\begin{aligned}
&|Pr[m, m' \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{CVSig}_S(m', C)\} : \mathcal{D}(\delta) = 1] - Pr[m, m' \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{Fake}_S(m, C)\} : \mathcal{D}(\delta) = 1]| \\
\leq\ & |Pr[m, m' \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{CVSig}_S(m', C)\} : \mathcal{D}(\delta) = 1] \\
& -Pr[m, m' \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{CVSig}_S(m, C)\} : \mathcal{D}(\delta) = 1]| \\
& +|Pr[m, m' \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{CVSig}_S(m, C)\} : \mathcal{D}(\delta) = 1] \\
& -Pr[m, m' \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{Fake}_S(m, C)\} : \mathcal{D}(\delta) = 1]| \\
=\ & |Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m', C)\} : \mathcal{D}(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m, C)\} : \mathcal{D}(\delta) = 1]| \\
& +|Pr[m \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{CVSig}_S(m, C)\} : \mathcal{D}(\delta) = 1] - Pr[m \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{Fake}_S(m, C)\} : \mathcal{D}(\delta) = 1]| \\
=\ & \epsilon^{Inv}_{\mathcal{D}:\,(m,m')} + \epsilon^{Sim-\mathsf{Fake}}_{\mathcal{D}:\,m}
\end{aligned}
$$

Note that $\epsilon^{Inv}_{\mathcal{D}:\,(m,m')}$ is the advantage of $\mathcal{D}$ to break the invisibility property when the challenge messages are m and m', and $\epsilon^{Sim-\mathsf{Fake}}_{\mathcal{D}:\,m}$ is the advantage of $\mathcal{D}$ to break the simulatability property with respect to the simulator Fake when the challenge message is m. Since the given scheme is invisible, then $\epsilon^{Inv}_{\mathcal{D}:\,(m,m')}$ is negligible for all PPT $\mathcal{D}$.

If the given CVS scheme is simulatable, then there has to exist a PPT simulator Fake' so that the following distributions are indistinguishable: $\{\mathsf{CVSig}_S(m, C)\}$ and $\{\mathsf{Fake}_S(m, C)\}$ for all S, m, C. That is, the following is negligible for all PPT D.

$$\epsilon^{Sim-\mathsf{Fake}'}_{D:\,m} = \big|Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m, C)\} : D(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{Fake}'_S(m, C)\} : D(\delta) = 1]\big|$$

As the condition that $\epsilon^{\mathsf{Fake}'}_D \leq \epsilon^{Inv}_{D:\,(m,m')} + \epsilon^{Sim-\mathsf{Fake}'}_{D:\,m}$ applies for all Fake including Fake' and $\epsilon^{Inv}_{D:\,(m,m')}$ is negligible in the security parameter since the scheme is invisible. For the scheme to be simulatable with respect to Fake', $\epsilon^{Sim-\mathsf{Fake}'}_{D:\,m}$ is negligible. These together imply $\epsilon^{\mathsf{Fake}'}_D$ must be negligible. In other words, then the following value must be negligible:

$$\big|Pr[m, m' \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{CVSig}_S(m', C)\} : \mathcal{D}(\delta) = 1] - Pr[m, m' \leftarrow \mathcal{M}; \delta \leftarrow \{\mathsf{Fake}'_S(m, C)\} : \mathcal{D}(\delta) = 1]\big|$$

which in essence implies $\{\mathsf{CVSig}_S(m', C)\} \cong \{\mathsf{Fake}'_S(m, C)\}$ for all S, m and C, and all $m' \neq m$. In fact, this necessary condition implies that there exists another PPT simulator Fake'' such that $\{\mathsf{CVSig}_S(m, C)\} \cong \{\mathsf{Fake}''_S(m, C)\}$ for all S, m, C, which is the sufficient condition for the scheme to be simulatable.

In conclusion, invisibility does not imply simulatability.

■

Theorem 13. Anonymity does not imply Simulatability.

Proof of Theorem 13.

In the following, we show the necessary requirement for an anonymous CVS scheme to be simulatable.

If the given CVS scheme is anonymous, then for any two signers $S0$ and $S1$, $\{\mathsf{CVSig}_{S0}(m, C)\} \cong \{\mathsf{CVSig}_{S1}(m, C)\}$, $\forall m, C$, that is, the following is negligible for all PPT D.

$$\epsilon^{Ano}_{D:\,(S0,S1)} = |Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m, C)\} : D(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m, C)\} : D(\delta) = 1]|$$

For all possible simulators Fake the following must hold for all PPT distinguisher $\mathcal{D}$ for all m, C and signers $S0 \neq S1$:

$$\begin{aligned}\epsilon_D^{\mathsf{Fake}} &= Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{Fake}_{S0}(m,C)\} : \mathcal{D}(\delta)=1] \\ &= Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m,C)\} : \mathcal{D}(\delta)=1] \\ &\quad + Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{Fake}_{S0}(m,C)\} : \mathcal{D}(\delta)=1]\end{aligned}$$

Taking absolute values on both sides, we get:

$$\begin{aligned}&|Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{Fake}_{S0}(m,C)\} : \mathcal{D}(\delta)=1]| \\ \leq\ & |Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m,C)\} : \mathcal{D}(\delta)=1]| \\ & + |Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{Fake}_{S0}(m,C)\} : \mathcal{D}(\delta)=1]| \\ =\ & |Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m,C)\} : \mathcal{D}(\delta)=1]| \\ & + |Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m,C)\} : \mathcal{D}(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{Fake}_{S0}(m,C)\} : \mathcal{D}(\delta)=1]| \\ =\ & \epsilon^{Ano}_{D:\,(S0,S1)} + \epsilon^{Sim-\mathsf{Fake}}_{D:\,S0}\end{aligned}$$

Note that $\epsilon^{Ano}_{\mathcal{D}:\,(S0,S1)}$ is the advantage of $\mathcal{D}$ to break the anonymity property for signers $S0$ and $S1$, and $\epsilon^{Sim-\mathsf{Fake}}_{\mathcal{D}:\,m}$ is the advantage of $\mathcal{D}$ to break the simulatability property with respect to the simulator Fake when the challenge message is m. Since the given scheme is anonymous, then $\epsilon^{Ano}_{\mathcal{D}:\,(S0,S1)}$ is negligible for all PPT $\mathcal{D}$.

If the given CVS scheme is simulatable, then there has to exist a PPT simulator Fake' so that the following distributions are indistinguishable: $\{\mathsf{CVSig}_S(m,C)\}$ and $\{\mathsf{Fake}_S(m,C)\}$ for all S, m, C. That is, the following is negligible for all PPT D.

$$\epsilon^{Sim-\mathsf{Fake}'}_{D:\,m} = \left|Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m,C)\} : D(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{Fake}'_S(m,C)\} : D(\delta)=1]\right|$$

As the condition that $\epsilon_D^{\mathsf{Fake}'} \leq \epsilon^{Ano}_{D:\,(S0,S1)} + \epsilon^{Sim-\mathsf{Fake}'}_{D:\,m}$ applies for all Fake including Fake' and $\epsilon^{Ano}_{D:\,(S0,S1)}$ is negligible in the security parameter since the scheme is anonymous. For the scheme to be simulatable with respect to Fake', $\epsilon^{Sim-\mathsf{Fake}'}_{D:\,m}$ is negligible. These together imply $\epsilon_D^{\mathsf{Fake}'}$ must be negligible. In other words, then the following value must be negligible:

$$\epsilon^{Sim}_{D:\,m} = |Pr[\delta \leftarrow \{\mathsf{CVSig}_S(m,C)\} : D(\delta)=1] - Pr[\delta \leftarrow \{\mathsf{Fake}_S(m,C)\} : D(\delta)=1]|$$

which in essence implies $\{\mathsf{CVSig}_{S1}(m,C)\} \cong \{\mathsf{Fake}_{S0}(m,C)\}$ for all $S0$, m and C, and all $S1 \neq S0$. In fact, this necessary condition implies that there exists another PPT simulator Fake'' such that $\{\mathsf{CVSig}_S(m,C)\} \cong \{\mathsf{Fake}''_S(m,C)\}$ for all S, m, C, which is the sufficient condition for the scheme to be simulatable.

In conclusion, invisibility does not imply simulatability. ■

Theorem 14 (Anonymity implies Invisibility). Assuming the partial signatures of a CVS scheme generated from two distinct and independently picked public/private key pairs (i.e. from two different signers) are independent, an anonymous CVS scheme is also invisible.

Proof of Theorem 14.

Assume the given CVS scheme is anonymous with negligible Adv^{Ano}_D for all PPT D.

Suppose there exists a PPT distinguisher $\mathcal{D}$ which can break the invisibility property, that is, able to distinguish which one of the two given messages m_0 and m_1 a given partial signature δ is for. We show how to construct another distinguisher $\mathcal{D}'$ to tell whether a given δ_b is signed by signer $S0$ or $S1$.

For the sake of clarity, we add an index to the distinguisher in such a way that $\mathcal{D}_{S0}$ denote an invisibility distinguisher which tells whether a given signature of S_0 is on message m_0 or m_1.

$\mathcal{D}'(\delta_b)$: $b = S0/S1$

Setup.
Get the witness public keys and the two signer public keys (for $S0$ and $S1$) from its challenger.
Run $\mathcal{D}$ on the public key of $S0$ and all witness public keys.
Get the private key of the signers from its challenger and pass the one for $S0$ to $\mathcal{D}$.
Query.
Pass all signing and endorsement queries from $\mathcal{D}$ to its oracle.
Relay the results back to $\mathcal{D}$.
Challenge.
$\mathcal{D}$ outputs (m_0, m_1, C) to be challenged.
Flip a coin $c \leftarrow \{0,1\}$. Output (m_c, C) to its challenger.
Pass the challenge δ_b to $\mathcal{D}$.
Guess.
$\mathcal{D}$ outputs a guess b'. Output the final guess b'' for b:

$$b'' = \begin{cases} b', & c = 0 \\ \overline{b'}, & c = 1 \end{cases}$$

Obviously, if $\mathcal{D}$ is PPT, so is $\mathcal{D}'$. Then the probability of success of $\mathcal{D}'$ with respect to anonymity is given by:

$$\begin{aligned}
Pr^{Ano}_{\mathcal{D}'}[Success] &= Pr[b'' = b|\delta_b] \\
&= \tfrac{1}{2}Pr[b' = b|\delta_b, c = 0] + \tfrac{1}{2}Pr[b' = \overline{b}|\delta_b, c = 1] \\
&= \tfrac{1}{4}Pr[b' = 0|\delta_0, c = 0] + \tfrac{1}{4}Pr[b' = 1|\delta_1, c = 0] \\
&\quad + \tfrac{1}{4}Pr[b' = 1|\delta_0, c = 1] + \tfrac{1}{4}Pr[b' = 0|\delta_1, c = 1] \\
&= \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m_0, C)\} : \mathcal{D}_{S0}(\delta) = 0] \\
&\quad + \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m_0, C)\} : \mathcal{D}_{S0}(\delta) = 1] \\
&\quad + \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m_1, C)\} : \mathcal{D}_{S0}(\delta) = 1] \\
&\quad + \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m_1, C)\} : \mathcal{D}_{S0}(\delta) = 0] \\
&= \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m_0, C)\} : \mathcal{D}_{S0}(\delta) = 0] \\
&\quad + \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m_1, C)\} : \mathcal{D}_{S0}(\delta) = 1] \\
&\quad + \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m_0, C)\} : \mathcal{D}_{S0}(\delta) = 1] \\
&\quad - \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m_1, C)\} : \mathcal{D}_{S0}(\delta) = 1] \\
&\quad + \tfrac{1}{4}
\end{aligned}$$

Note we use in the above expression the fact:

$$Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m_1, C)\} : \mathcal{D}_{S0}(\delta) = 0] + Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m_1, C)\} : \mathcal{D}_{S0}(\delta) = 1] = 1$$

Note also that the probability of success of $\mathcal{D}_{S0}$ with respect to invisibility is given by:

$$Pr^{Inv}_{\mathcal{D}}[Success] = \frac{1}{2}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m_0, C)\} : \mathcal{D}_{S0}(\delta) = 0] + \frac{1}{2}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S0}(m_1, C)\} : \mathcal{D}_{S0}(\delta) = 1]$$

Hence,

$$\begin{aligned}
\tfrac{1}{2}Pr^{Inv}_{\mathcal{D}}[Success] + \tfrac{1}{4} &= Pr^{Ano}_{\mathcal{D}'}[Success] + \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m_0, C)\} : \mathcal{D}_{S0}(\delta) = 1] \\
&\quad -\tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m_1, C)\} : \mathcal{D}_{S0}(\delta) = 1] \\
\tfrac{1}{2}\left(Pr^{Inv}_{\mathcal{D}}[Success] - \tfrac{1}{2}\right) &= \left(Pr^{Ano}_{\mathcal{D}'}[Success] - \tfrac{1}{2}\right) + \tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m_0, C)\} : \mathcal{D}_{S0}(\delta) = 1] \\
&\quad -\tfrac{1}{4}Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m_1, C)\} : \mathcal{D}_{S0}(\delta) = 1] \\
Adv^{Inv}_{\mathcal{D}} &\leq 2Adv^{Ano}_{\mathcal{D}'} + \tfrac{1}{2}\epsilon^{S1}_{S0:m_0,m_1}
\end{aligned}$$

where
$\epsilon^{S1}_{S0:m_0,m_1} = |Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m_0, C)\} : \mathcal{D}_{S0}(\delta) = 1] - Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m_1, C)\} : \mathcal{D}_{S0}(\delta) = 1]|$

Provided signatures from distinct, independent signing keys are independent, the signatures from $S1$, namely, $\mathsf{CVSig}_{S1}(m_0, C)$ and $\mathsf{CVSig}_{S1}(m_1, C)$, should be independent of the view of $\mathcal{D}_{S0}$ initialized for $S0$'s signatures. As a result, $\mathcal{D}_{S0}$ should not make a guess better than a random one. Hence, $Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m_0, C)\} : \mathcal{D}_{S0}(\delta) = 1] \approx Pr[\delta \leftarrow \{\mathsf{CVSig}_{S1}(m_1, C)\} : \mathcal{D}_{S0}(\delta) = 1] \approx \frac{1}{2}$ and the term $\epsilon^{S1}_{S0:m_0,m_1}$ should be negligible.

As a result, if $Adv^{Inv}_{\mathcal{D}}$ is non-negligible, then $Adv^{Ano}_{\mathcal{D}'}$ should also be non-negligible (a contradiction). In conclusion, if partial signatures generated from different signing keys are independent (which is usually true), then anonymity of a CVS scheme implies its invisibility. ■

Appendix B: Proofs — The Security of the Generic CVS Construction

Security of the Generic CVS Construction from IBE

Lemma 15. If the underlying ordinary signature scheme SIG is existentially unforgeable under a chosen message attack, then the generic CVS construction is unforgeable.

Proof of Lemma 15.

We prove the unforgeability property of the generic construction by contradiction. Assume SIG is existentially unforgeable under chosen message attacks. Suppose there is a PPT forging algorithm $\mathcal{F}$ which can forge a CVS partial signature with probability of success $p_{\mathcal{F}}^{CVS}$. We show how to construct another forging algorithm $\mathcal{F}'$ from $\mathcal{F}$ to forge a signature for SIG.

$\mathcal{F}'$

Setup.
Ask its challenger for the signer public key PK_S.
Run $Setup$ to get all the witness public/private key pairs $(PK_{W_i}, sk_{W_i}),\ 1 \leq i \leq N$.
Run $\mathcal{F}$ on PK_S and (PK_{W_i}, sk_{W_i}).
Query.
When $\mathcal{F}$ issues a O_S query for $\langle m_j, C_j \rangle$ where $C_j = \{(c_{ji}, W_{ji}) : 1 \leq i \leq N\}$,
ask its signing orale for an ordinary signature $\sigma_j = Sig(sk_s, m_j)$.
Randomly choose a_{ji} ($1 \leq i \leq N$) to create a partial signature:
$$\delta_j = \left\langle \sigma_j \oplus h\left(\bigoplus_i^N a_{ji}\right), \left\{Enc(PK_{W_{ji}}, c_{ji}, a_{ji})\right\}, Com\left(\sigma_j, h\left(\bigoplus_i^N a_{ji}\right)\right)\right\rangle$$
With a_{ji}'s, σ_j, and all random coins used, run the confirmation protocol with $\mathcal{F}$.
Guess.
$\mathcal{F}$ outputs a guess (m, σ). Output (m, σ).

Obviously, if F is PPT, then F' is also PPT (as Enc and Com are also PPT). Note that $\mathcal{F}$ should output $m \neq m_j,\ \forall j$ The probability of success of $\mathcal{F}'$ is:

$$p_{\mathcal{F}'}^{SIG} = Pr[Ver(m, \sigma, PK_S) = 1] = p_{\mathcal{F}}^{CVS}$$

If the CVS scheme is forgeable, that is, $p_{\mathcal{F}}^{CVS}$ is non-negligible, then $p_{\mathcal{F}'}^{SIG}$ is also non-negligible (a contradiction). Hence, if SIG is unforgeable in the sense that p_A^{SIG} is negligible for all PPT A, then so is the CVS scheme given by the generic construction. ■

Lemma 16. Given a pseudorandom generator and a computationally hiding commitment scheme, if the underlying IBE scheme is semantic secure, then the generic CVS construction is simulatable with respect to the given simulator Fake.

Proof of Lemma 16.

It is easy to show that the given CVS scheme with one witness is secure, then a CVS scheme with many witnesses is also secure. Hence, we will consider a single witness case.

Assume IBE is IND-ID-CPA secure, h is a pseudorandom generator, and COM is computationally hiding. Suppose $\mathcal{D}$ is a PPT distinguisher which has non-negligible advantage $Adv_{\mathcal{D}}^{Sim}$ in winning the simulatability game defined in Definition 6. We can base on $\mathcal{D}$ to construct another distinguisher $\mathcal{D}'$ to break the semantic security of IBE.

To avoid confusion, we should clarify that in the following discussion, we denote the challenge ciphertext of the IBE game by $C_b, b \in \{0, 1\}$ and the queried verifiability condition set by C_j.

$\mathcal{D}'(C_b), \quad b \in \{0,1\}$

Setup.
Ask its challenger for the public key PK_G of the PKG. Use it as the witness public key for W.
Run CVKGS to generate the signer public/private key pair (PK_S, sk_S).
Run $\mathcal{D}$ on PK_G and (PK_S, sk_S).

Query.
Signing Queries (O_S) on $\langle m_j, C_j \rangle$ where $C_j == (c_j, W)$.
- Generate $\sigma_j = Sig(m_j, sk_S)$
- Randomly pick a_j and encrypts itself to generate the partial signature:
  $\delta_j = \langle \sigma_j \oplus h(a_j),\ Enc(PK_G, c_j, a_j),\ Com(\sigma_j, h(a_j)) \rangle$
- Based on all the random coins used, run the confirmation protocol with $\mathcal{D}$.

Endorsement Queries (O_E) on (c_j, W).
- Pass all endorsement queries (c_j, W) from $\mathcal{D}$ as extraction queries on c_j to its oracle to get d_j.
- d_j is equivalent to $\sigma_W(c_j)$.

Challenge.
$\mathcal{D}$ outputs m and (c, W) to ask for a challenge.
Create a signature σ_t on a message m using Sig.
Randomly pick $\sigma_f \in \mathcal{S}_\sigma$.
Randomly pick $a_t, a_f \in \mathcal{P}_{IBE}$. Output a_t and a_f to ask for a challenge C_b where

$$C_b = \begin{cases} Enc(PK_G, c, a_t), & b = 0 \\ Enc(PK_G, c, a_f), & b = 1. \end{cases}$$

Flip a coin $e \in \{0,1\}$ and send the following challenge to $\mathcal{D}$:

$$\delta_e = \begin{cases} \langle \sigma_t \oplus h(a_t),\ C_b,\ Com(\sigma_t, h(a_t)) \rangle, & e = 0 \\ \langle \sigma_f \oplus h(a_f),\ C_b,\ Com(\sigma_f, h(a_f)) \rangle, & e = 1 \end{cases}$$

Guess. $\mathcal{D}$ outputs a guess b'. Output b' as a guess for b.

Note: $\langle \sigma_t \oplus h(a_t), Enc(PK_G, c, a_t), Com(\sigma_t, h(a_t)) \rangle$ is equivalent to $\mathsf{CVSig}_S(m, C)$ and $\langle \sigma_f \oplus h(a_f), Enc(PK_G, c, a_f), Com(\sigma_f, h(a_f)) \rangle$ is equivalent to $\mathsf{Fake}(C)$.

Obviously, if $\mathcal{D}$ is PPT, so is $\mathcal{D}'$ (assuming Enc, h and Com are all PPT). In the following discussion, we abuse the notation — we write $\mathcal{D}(\delta)$ instead the full notation $\mathcal{D}(\delta, m, C)$. Hence, (m, C) is always part of the input to $\mathcal{D}$ and the associated algorithms. Again, we abuse the notation by writing $Enc(PK_G, c, a)$ as $Enc(a)$.
The probability of success of $\mathcal{D}'$ is given by:

$$\begin{aligned} Pr^{IBE}_{\mathcal{D}'}[Success] &= Pr[b' = b | C_b] \\ &= \tfrac{1}{2} Pr[\mathcal{D}(\delta_e) = 0 | b = 0] + \tfrac{1}{2} Pr[\mathcal{D}(\delta_e) = 1 | b = 1] \\ &= \tfrac{1}{4} Pr[\mathcal{D}(\delta_e) = 0 | \delta_e = \langle \sigma_t \oplus h(a_t), Enc(a_t), Com(\sigma_t, h(a_t)) \rangle] \\ &\quad + \tfrac{1}{4} Pr[\mathcal{D}(\delta_e) = 0 | \delta_e = \langle \sigma_f \oplus h(a_f), Enc(a_t), Com(\sigma_f, h(a_f)) \rangle] \\ &\quad + \tfrac{1}{4} Pr[\mathcal{D}(\delta_e) = 1 | \delta_e = \langle \sigma_t \oplus h(a_t), Enc(a_f), Com(\sigma_t, h(a_t)) \rangle] \\ &\quad + \tfrac{1}{4} Pr[\mathcal{D}(\delta_e) = 1 | \delta_e = \langle \sigma_f \oplus h(a_f), Enc(a_f), Com(\sigma_f, h(a_f)) \rangle]. \end{aligned}$$

Note that

$$\begin{aligned} Pr^{Sim}_{\mathcal{D}}[Success] &= \tfrac{1}{2} Pr[\mathcal{D}(\delta_e) = 0 | \delta_e = \langle \sigma_t \oplus h(a_t), Enc(a_t), Com(\sigma_t, h(a_t)) \rangle] \\ &\quad + \tfrac{1}{2} Pr[\mathcal{D}(\delta_e) = 1 | \delta_e = \langle \sigma_f \oplus h(a_f), Enc(a_f), Com(\sigma_f, h(a_f)) \rangle]. \end{aligned}$$

Substituting $Pr^{IBE}_{\mathcal{D}'}[Success]$ into $Pr^{Sim}_{\mathcal{D}}[Success]$, we have

$$\begin{aligned}\tfrac{1}{2}Pr^{Sim}_{\mathcal{D}}[Success] &= Pr^{IBE}_{\mathcal{D}'}[Success] \\ &\quad -\tfrac{1}{4}Pr[\mathcal{D}(\delta_e)=0|\delta_e=\langle\sigma_f\oplus h(a_f), Enc(a_t), Com(\sigma_f, h(a_f))\rangle] \\ &\quad -\tfrac{1}{4}Pr[\mathcal{D}(\delta_e)=1|\delta_e=\langle\sigma_t\oplus h(a_t), Enc(a_f), Com(\sigma_t, h(a_t))\rangle] \\ &= Pr^{IBE}_{\mathcal{D}'}[Success]-\tfrac{1}{4} \\ &\quad +\tfrac{1}{4}Pr[\mathcal{D}(\delta_e)=1|\delta_e=\langle\sigma_f\oplus h(a_f), Enc(a_t), Com(\sigma_f, h(a_f))\rangle] \\ &\quad -\tfrac{1}{4}Pr[\mathcal{D}(\delta_e)=1|\delta_e=\langle\sigma_t\oplus h(a_t), Enc(a_f), Com(\sigma_t, h(a_t))\rangle].\end{aligned}$$

Subtracting $\frac{1}{4}$ and then taking absolute values on both sides, we have

$$\begin{aligned}\tfrac{1}{2}Adv^{Sim}_{\mathcal{D}} \leq\ & Adv^{IBE}_{\mathcal{D}'}+\tfrac{1}{4}|Pr[\mathcal{D}(\delta_e)=1|\delta_e=\langle\sigma_f\oplus h(a_f), Enc(a_t), Com(\sigma_f, h(a_f))\rangle] \\ & -Pr[\mathcal{D}(\delta_e)=1|\delta_e=\langle\sigma_t\oplus h(a_t), Enc(a_f), Com(\sigma_t, h(a_t))\rangle]|.\end{aligned}$$

Let $\varepsilon_{\mathcal{D}}$ denote $|Pr[\mathcal{D}(\delta_e)=1|\delta_e=\langle\sigma_f\oplus h(a_f), Enc(a_t), Com(\sigma_f, h(a_f))\rangle]-Pr[\mathcal{D}(\delta_e)=1|\delta_e=\langle\sigma_t\oplus h(a_t), Enc(a_f), Com(\sigma_t, h(a_t))\rangle]|$. Then we could view $\varepsilon_{\mathcal{D}}$ as the advantage of $\mathcal{D}$ in distinguishing the following two distributions:

$$\begin{aligned}\Delta_f &= \{m\leftarrow\mathcal{M}; c\leftarrow\mathcal{C}; \sigma_f\leftarrow\mathcal{S}_\sigma; a, a'\leftarrow\mathcal{P}_{IBE}: (\sigma_f\oplus h(a), Enc_W(c,a'), Com(\sigma_f, h(a)))\}, \\ \Delta_t &= \{m\leftarrow\mathcal{M}; c\leftarrow\mathcal{C}; \sigma_t\leftarrow\{Sig_S(m)\}; a, a'\leftarrow\mathcal{P}_{IBE}: (\sigma_t\oplus h(a), Enc_W(c,a'), Com(\sigma_t, h(a)))\}\end{aligned}$$

We argue that $Enc_W(c,a')$ would not have useful information to help $\mathcal{D}$ in distinguishing the above two distributions as a and a' are picked independently; even if one know how to decrypt $Enc_W(c,a')$ to obtain a', a' has no useful information about a which is needed to tell whether a given δ comes from Δ_f or Δ_t. If $\varepsilon_{\mathcal{D}}$ is non-negligible, then it is straightforward to construct from $\mathcal{D}$ another algorithm D'' with an advantage $\varepsilon_{\mathcal{D}''}=\varepsilon_{\mathcal{D}}$ to distinguish the following two distributions:

$$\begin{aligned}\Pi_f &= \{m\leftarrow\mathcal{M}; \sigma_f\leftarrow\mathcal{S}_\sigma; a\leftarrow\mathcal{P}_{IBE}: (\sigma_f\oplus h(a), Com(\sigma_f, h(a)))\}, \\ \Pi_t &= \{m\leftarrow\mathcal{M}; \sigma_t\leftarrow\{Sig_S(m)\}; a\leftarrow\mathcal{P}_{IBE}: (\sigma_t\oplus h(a), Com(\sigma_t, h(a)))\}\end{aligned}$$

The idea of the construction of $\mathcal{D}''$ is when a challenge $(\sigma\oplus h(a), Com(\sigma, h(a)))$ (where σ could be equal to σ_t or σ_f) is received, $\mathcal{D}''$ randomly picks $a'\in\mathcal{P}_{IBE}$, creates $Enc_W(c,a')$, and add it to the challenge to create a new challenge $(\sigma\oplus h(a), Enc_W(c,a'), Com(\sigma, h(a)))$ for $\mathcal{D}$.

The advantage of reducing the problem of distinguishing Π_f/Π_t to that of distinguishing Δ_f/Δ_t is the adaptive queries, more specifically, the endorsement queries, in the simulatability game would not help in any way in distinguishing Π_f and Π_f. In other words, we do not need to take into account of adaptive queries while showing the indistinguishability between Π_f and Π_t. Besides, the indistinguishability between Π_f and Π_t implies that of Δ_f and Δ_t in the simulatability game.

Let ϵ_h and ϵ_{COM} be the indistinguishability coefficients of the pesudorandom generator and the commitment scheme. Recall that ϵ_h denotes the advantage of the best PPT distinguisher in distinguishing between the output distribution of a pseudorandom generator $h: \{0,1\}^{l_p}\rightarrow\{0,1\}^{l_s}$ and a uniform distribution over the output space of h, that is, between $\{x\leftarrow\{0,1\}^{l_p}: h(x)\}$ and $\{y\leftarrow\{0,1\}^{l_s}: y\}$. Whereas, ϵ_{COM} denotes the advantage of the best PPT distinguisher in distinguishing between the output distributions of the commitments of two different input values, say σ_f and σ_t, that is, between $\{r\leftarrow\{0,1\}^*: Com(\sigma_f, r)\}$ and $\{r\leftarrow\{0,1\}^*: Com(\sigma_t, r)\}$. Now, we can show the indistinguishability between Π_f and Π_t. In the following discussion, if X and Y are computationally indistinguishable, we

denote $X \cong Y$. The proof below is based on the standard hybrid argument and Lemma 1.

$$
\begin{aligned}
\Pi_t &= \{m \leftarrow \mathcal{M}; \sigma_t \leftarrow \{Sig_S(m)\}; a \leftarrow \mathcal{P}_{IBE} : (\sigma_t \oplus h(a), Com(\sigma_t, h(a)))\} \\
&\cong \{m \leftarrow \mathcal{M}; \sigma_t \leftarrow \{Sig_S(m)\}; r \leftarrow \{0,1\}^{l_s} : (\sigma_t \oplus r, Com(\sigma_t, r))\} && \text{(with } \epsilon_h\text{)} \\
&\cong \{m \leftarrow \mathcal{M}; \sigma_t \leftarrow \{Sig_S(m)\}; r, r' \leftarrow \{0,1\}^{l_s} : (r', Com(\sigma_t, r))\} \\
&\cong \{m \leftarrow \mathcal{M}; \sigma_f \leftarrow \mathcal{S}_\sigma; r, r' \leftarrow \{0,1\}^{l_s} : (r', Com(\sigma_f, r))\} && \text{(with } \epsilon_{COM}\text{)} \\
&\cong \{m \leftarrow \mathcal{M}; \sigma_f \leftarrow \mathcal{S}_\sigma; r \leftarrow \{0,1\}^{l_s} : (\sigma_f \oplus r, Com(\sigma_f, r))\} \\
&\cong \{m \leftarrow \mathcal{M}; \sigma_f \leftarrow \mathcal{S}_\sigma; a \leftarrow \mathcal{P}_{IBE} : (\sigma_f \oplus h(a), Com(\sigma_f, h(a)))\} && \text{(with } \epsilon_h\text{)} \\
&= \Pi_f
\end{aligned}
$$

As a result, $\varepsilon_{\mathcal{D}} = \varepsilon_{\mathcal{D}''} < 2\epsilon_h + \epsilon_{COM}$. Substituting back, we have

$$
\begin{aligned}
\tfrac{1}{2} Adv_{\mathcal{D}}^{Sim} &< Adv_{\mathcal{D}'}^{IBE} + \tfrac{1}{2}\epsilon_h + \tfrac{1}{4}\epsilon_{COM} \\
Adv_{\mathcal{D}}^{Sim} &< 2Adv_{\mathcal{D}'}^{IBE} + \epsilon_h + \tfrac{1}{2}\epsilon_{COM}.
\end{aligned}
$$

If we assume COM is computationally hiding and h is a pseudorandom generator, then both ϵ_h and ϵ_{COM} should be negligible in their security parameters. Consequently, if $Adv_{\mathcal{D}}^{Sim}$ is non-negligible, the only possibility is either $Adv_{\mathcal{D}'}^{IBE}$ is non-negligible, meaning $\mathcal{D}'$ could break the semantic security of the IBE scheme (a contradiction). In other words, the semantic security of the IBE scheme implies the simulatability of the CVS construction with respect to the given construction of Fake. Since Fake is PPT, we could conclude that the given generic CVS construction is simulatable. ■

Appendix C: Proof — Semantic Security of the IBE Construction from CVS

Proof of Theorem 18.

Now, we show that the above construction satisfies the conditions for IND-ID-CPA secure IBE. We assume the CVS scheme is simulatable with respect to Fake. Suppose the above constructed IBE scheme is not IND-ID-CPA secure, that is, there exists an adversary $\mathcal{D}$ which can win the IND-ID-CPA game with a non-negligible advantage $Adv_{\mathcal{D}}^{IBE}$. In other words, given a ciphertext (m, δ_b, PK_S) where δ_b is a valid/fake partial signature when $b = 0/1$, $\mathcal{D}$ could tell whether the plaintext bit $b = 0$ or $b = 1$ with a non-negligible advantage. Up to this point, it is clear that $\mathcal{D}$ could be used to break the simulatability property of the underlying CVS scheme with respect to Fake. However, for completeness, we show how to construct another adversary $\mathcal{D}'$ from $\mathcal{D}$ to tell whether a given partial signature δ_b originates from CVSig or Fake.

$\mathcal{D}'(\delta_b)$

Setup.
Get the public key PK_G of the witness from its challenger. Run $\mathcal{D}$ on PK_G.
Get the signer's public/private key pair (PK_S, sk_S).
Query.
Extraction Query $\langle ID_j \rangle$. Pass all extraction queries from $\mathcal{D}$ to its endorsement oracle.
Challenge.
$\mathcal{D}$ outputs ID to be challenged. (Note the plaintext could only be 0 or 1.)
Randomly select a message $m \in \mathcal{M}$.
Pass m, ID to its challenger and receive the challenge δ_b.
Pass $C_b = (m, \delta_b, PK_S)$ as a challenged ciphertext to $\mathcal{D}$.
Guess.
$\mathcal{D}$ outputs a guess b'. Output b' as a guess for b.

It obvious that the advantage of $\mathcal{D}'$ with respect to CVS simulatability is the same as the advantage of $\mathcal{D}$ on breaking the semantic security of the IBE scheme. Hence, if the latter is non-negligible, so is the former, a contradiction as we assume the given CVS scheme is simulatable with respect to Fake. In conclusion, the constructed IBE scheme is semantically secure as long as the CVS scheme is simulatable. ■

Appendix D: A CVS Construction based on Verifiable Encryption (VE)

In [1, 5], a fairly general technique called verifiable encryption (VE) which encrypts and runs a proof protocol simultaneously is proposed. This technique has been used for constructing designated confirmer signature schemes by Goldwasser et. al [28]. For any binary relation $\mathcal{R}$ on which a Σ-protocol for proof of knowledge exists, the VE technique could be used to encrypt the witness w for a certain x (such that $(x, w) \in \mathcal{R}$) while at the same time prove to the recipient that he is really receiving an encryption of w. In VE, the resulting communication transcript is used as the encryption of w. As this technique is fairly general, we could use it to construct a CVS scheme with any IBE schemes if there is no design restriction forbidding the merge of the blinding mechanism and the confirmation protocol into one entity. To use VE for efficient CVS construction, the only restriction is that the verification function of the underlying signature scheme is a certain homomorphic one-way function on some encoding of the message. In fact, most standard signature schemes like RSA and ElGamal belong to this type as illustrated in the following example.

Example 1

RSA *We consider the simple hash-and-sign RSA signature. The public key is (n, e) where $n = pq$ for some large primes p and q, and $ed \equiv 1 \ (mod\ \phi(n))$. The signature of a message m is $\sigma = h(m)^d$ mod n. To verify, check if $\sigma^e \stackrel{?}{=} h(m)$ mod n.*

Let $f(x) = x^e$ mod n. Given two signatures σ_0 and σ_1 for two messages m_0 and m_1, it is easy to see that $f(\sigma_0\sigma_1) = (\sigma_0\sigma_1)^e = \sigma_0^e\sigma_1^e = f(\sigma_0)f(\sigma_1) = h(m_0)h(m_1)$. That is, a signature is the homomorphic pre-image of f on the hashed message.

For simplicity, we show how to construct a CVS scheme with a single witness out of VE; a straightforward extension with multiple witness is possible.

Suppose the verification equation of a certain existentially unforgeable signature scheme SIG for a message signature pair (m, σ) is: $f(\sigma) \stackrel{?}{=} \hat{m}$ and f is homomorphic in the sense that $f(\sigma_0\sigma_1) = f(\sigma_0)f(\sigma_1)$ where $\hat{m}$ denotes some encoding on m. Denote the signer and the recipient by S and V respectively, and let $Enc(r, ID, x)$ be the encryption function of a semantically secure IBE on a message x for an identity ID with a random coin r.

Given a message m, a condition statement c and its ordinary signature σ, the partial signature generation and confirmation protocol constructed based on VE is as follows (Depicted below is just a single round of iteration):

1. **Commit**: S randomly picks $\gamma \in \mathcal{S}$ from the signature space of σ, encrypts γ and $\gamma\sigma$ respectively to get $e_0 = Enc(r_0, c, \gamma)$ and $e_1 = Enc(r_1, c, \gamma\sigma)$ where r_0, r_1 are just random coins for encryption. S computes $\beta = f(\gamma)$. S gives V the following: β, e_0, e_1. Note that f is the signature verification equation.

2. **Challenge**: V flips a coin $b \in \{0, 1\}$ and sends b as a challenge for S.

3. **Response**: S replies V with the following:
$$(u_b, v_b) = \begin{cases} (r_0, \gamma), & b = 0 \\ (r_1, \gamma\sigma), & b = 1 \end{cases}$$

4. **Verify**: V checks the following:
$$\text{If } b = 0, \quad \text{check } \beta \stackrel{?}{=} f(v_0);\ e_0 \stackrel{?}{=} Enc(u_0, v_0)$$
$$\text{If } b = 1, \quad \text{check } \beta\hat{m} \stackrel{?}{=} f(v_1);\ e_1 \stackrel{?}{=} Enc(u_1, v_1)$$

Suppose $\overline{b_i}$ is the negation of b_i with definition as follows:

$$\overline{b_i} = \begin{cases} 0, & b_i = 1 \\ 1, & b_i = 0 \end{cases}$$

In each round, the probability that the signer S could cheat successfully is $\frac{1}{2}$, the successful cheating probability for all k rounds becomes $\frac{1}{2^k}$. Besides, b cannot be all zero in all the k rounds. If the verification test is passed for all k rounds, the resulting partial signature would be the k-tuple $\{(b_i, v_{b_i}, e_{\overline{b_i}}) : 1 \leq i \leq k\}$. In each round, the responses to all (two) possible challenges $b = 0$ or $b = 1$ are computed (resulting in γ and $\gamma\sigma$) and encrypted (to give e_0 and e_1) by S which in response to the challenge b would reveal one of them e_b. Once the remaining encryption $e_{\overline{b}}$ is decrypted, the recipient could recover σ using the previously reveled response and the challenge. For example, if the challenge $b = 0$ in a particular round, $(0, \gamma, e_1)$ would be the partial signature output for that round; once e_1 is decrypted using the witness signatures, $\gamma\sigma$ for that round is recovered, from which σ could be recovered dividing $\gamma\sigma$ (obtainable from e_1) with γ. In order to ensure a reasonably high probability to recover σ, all k tuples need to be stored by the recipient although one of them is sufficient for recovering σ if the signer is honest.

As can be seen, the partial signature size could be considerably large in some cases.

Appendix E: A Possible Construction for the Homomorphic Mapping

The norm $N(e)$ for the extension field $\mathbb{F}_{p^l}$ is defined as follows.

Definition 16 *For any $e \in \mathbb{F}_{p^l}$, $N(e) \in \mathbb{F}_p$ is defined by:*

$$N(e) = e \times e^p \ldots \times e^{p^{l-1}} = e^{(p^l-1)/(p-1)}$$

The norm satisfies the following properties:

1. $N(e_1 e_2) = N(e_1)N(e_2), \quad \forall e_1, e_2 \in \mathbb{F}_{p^l}$;
2. N maps $\mathbb{F}_{p^l}$ onto $\mathbb{F}_p$ and $\mathbb{F}^*_{p^l}$ onto $\mathbb{F}^*_p$;
3. $N(a) = a^l, \quad \forall a \in \mathbb{F}_p$;

A Example Construction

Let $s \equiv 1/l \bmod (p-1)$. We construct the invertible group homomorphism : $\mathbb{F}_p \rightarrow \mathbb{F}_{p^l}$ as follows:

$$\begin{array}{ll} f(a) = a, & \forall a \in \mathbb{F}_p \\ f^{-1}(e) = N(e)^s, & \forall e \in \mathbb{F}_{p^l} \end{array}$$

We could check the correctness of the inverse as follows.
For all $a \in \mathbb{F}_p$,

$$f^{-1}(f(a)) = f^{-1}(a) = N(a)^s = (a^l)^s = a$$

We could check the homomorphic property as follows:
For any $a_1, a_2 \in \mathbb{F}_p$,

$$f(a_1 a_2) = a_1 a_2 = f(a_1) f(a_2)$$

For any $e_1, e_2 \in \mathbb{F}_{p^l}$,

$$f^{-1}(e_1 e_2) = N(e_1 e_2)^s = N(e_1)^s N(e_2)^s = f^{-1}(e_1) f^{-1}(e_2)$$

Since p is a safe prime, $p - 1 = 2n$ where n is a composite of large primes. In most cases, l should be much smaller that any prime in n. As a result, if l is odd, l is prime to $p-1$ and we could easily find s as l^{-1} mod $(p-1)$ using the extended Euclidean algorithm. If l is even, we need to find the image of the inverse mapping as the $l - th$ root in $\mathbb{Z}^*_p$. To find s, we could break down l as $2l'$ so that l' is odd and should be prime to $(p-1)$. The inverse of l' in $\mathbb{Z}_{p-1}$ can be computed using the extended Euclidean algorithm, from which we could find the $l' - th$ root in $\mathbb{Z}^*_p$. To find the $l - th$ root, we can take square root mod p on the $l' - th$ root, which has an efficient algorithm [32].

To ensure that such computation is possible for the recipient while running the confirmation protocol, we need to restrict to using even r in computing the pairings.

Appendix F: Security Analyses of the Pairing-based CVS Constructions

Proof of the CVS Unforgeability Theorem

We are going to prove that given a signature scheme existentially unforgeable against a chosen message attack, a CVS scheme constructed from it using a PPT blinding mechanism is also unforgeable. When we say a given signature scheme SIG is existentially unforgeable against a chosen message attack, we mean that any PPT adversary, allowed access to a signing oracle which returns valid signatures on messages chosen by the adversary, cannot create a valid signature on a message not previously queried to the signing oracle except with negligible probability in terms of the security parameter.

Proof of Theorem 20.

Assume that SIG is existentially unforgeable under a chosen message attack and B is a PPT blinding mechanism in CVS for creating partial signatures from an ordinary signature of SIG. Suppose there exists a PPT forger $\mathcal{F}$ which can break the unforgeability of CVS. Note that B would only take an ordinary signature of SIG, witness public keys and condition statements as input, otherwise, it is inapplicable for generating partial signatures. Then, we could use $\mathcal{F}$ to construct a forger $\mathcal{F}'$ to create an existential forgery for SIG. The forging algorithm $\mathcal{F}'$ runs as follows:

$\mathcal{F}'$ gets the signer public key PK_S from its SIG challenger and runs CVKGW to generate a set of witness public/private key pairs $\{(PK_i, sk_i)\}$ and run $\mathcal{F}$ on the keys PK_S, $\{(PK_i, sk_i)\}$. When $\mathcal{F}$ makes a signing query on a message m_j and a set of verifiability conditions C_j, $\mathcal{F}'$ passes m_j to its own signing oracle to query an ordinary signature σ_j, and then runs B on σ_j to create a partial signature δ_j. $\mathcal{F}'$ returns δ_j to $\mathcal{F}$ and uses the knowledge of σ_j and all the random coins used by B to carry out a confirmation protocol with $\mathcal{F}$, and this completes the reply to the query made by $\mathcal{F}$. Finally, $\mathcal{F}$ has to output a message signature pair (m, σ); $\mathcal{F}'$ passes this as its output to the SIG challenger.

Obviously, if $\mathcal{F}$, B and the key generation algorithms are all PPT, so is $\mathcal{F}'$. Besides, $\mathcal{F}'$ perfectly simulates the adversary environment for $\mathcal{F}$, and $\mathcal{F}$ should return, with a probability of success $p_{\mathcal{F}}^{UF-CVS}$, a valid message signature pair (m, σ) with $m \neq m_j$, $\forall j$. Note that σ is a valid ordinary signature of SIG for m. Then the probability of success $p_{\mathcal{F}'}^{UF-SIG}$ of $\mathcal{F}'$ in creating an existential forgery for SIG is $p_{\mathcal{F}}^{UF-CVS}$. If $p_{\mathcal{F}}^{UF-CVS}$ is non-negligible in the security parameter λ, so is $p_{\mathcal{F}'}^{UF-SIG}$, which is contradictory to the assumption that SIG is existentially unforgeable. Hence, the resulting CVS scheme built on SIG and B must be unforgeable if SIG is unforgeable. ■

Security Analysis for the Pairing-based CVS Construction for Elgamal Signatures

Proof of the Simulatability Property (Claim 21)

Given a signer public key pair (g, y_s), witness public keys $(P_i, Y_i), 1 \leq i \leq N$, and a set of condition statements c_i, breaking the simulatability property of the Elgamal based CVS scheme is in essence to distinguish which of the following two distributions a given tuple $\delta = (\gamma, z, U_1, U_2, \ldots, U_N)$ belongs to:

- $\mathcal{CVS}(N) = \left\{(\gamma, z, U_1, U_2, \ldots, U_N) : \gamma \leftarrow G; a = Dlog_\gamma(g^{h(m)} y_s^\gamma); r \leftarrow \mathbb{Z}_q^*; U_i = rP_i; z = f(a) \prod_{i=1}^N y_i^r\right\}$
- $\mathcal{FAKE}(N) = \left\{(\gamma, z, U_1, U_2, \ldots, U_N) : \gamma \leftarrow G; a \leftarrow \mathbb{Z}_p^*; r \leftarrow \mathbb{Z}_q^*; U_i = rP_i; z = f(a) \prod_{i=1}^N y_i^r\right\}$

where $y_i = \hat{e}(Y_i, H(c_i))$. The first one is the distribution of a partial signature CVSig whereas the second one is that of the output of a simulator Fake. Of course, an adversary is allowed to make queries on other partial signatures and simulator outputs before receiving such a problem as a challenge.

Proof We first show that simulatability could be achieved by the Elgamal CVS construction for the single-witness case in the random oracle model if the decisional bilinear Diffie Hellman problem is hard. Then we give a security analysis to discuss why the simulatability property for the single-witness case implies that of the multiple-witness case.

Security Analysis for the Single-Witness Case

To prove the simulatability property, we need to show that there is no PPT algorithm which can distinguish the following two distributions with a probability of success significantly better than a wild guess (even allowed to make CVS signing queries O_S and endorsement queries O_E):

- $\mathcal{CVS}(1) = \{(\gamma, z, U) : \gamma \leftarrow G; a = Dlog_\gamma(g^{h(m)} y_s^\gamma); r \leftarrow \mathbb{Z}_q^*; U = rP_1; z = f(a) y_1^r\}$
- $\mathcal{FAKE}(1) = \{(\gamma, z, U) : \gamma \leftarrow G; a \leftarrow \mathbb{Z}_p^*; r \leftarrow \mathbb{Z}_q^*; U = rP_1; z = f(a) y_1^r\}$

where (P_1, Y_1) is the public key of the witness and $y_1 = \hat{e}(Y_1, H(c))$ for a condition statement c.

Instead of proving the simulatability game, we prove another simpler one. We replace a in the first distribution $\mathcal{CVS}(1)$ by a random number picked by the adversary instead of a part of the Elgamal signature of a message picked by the adversary. That is, we do not restrict a to be part of the Elgamal signature but a random number picked by the adversary. Details of the new game is as follows:

$\underline{GameA}$

The challenger runs CVKGW(1^λ) to generate the witness private key x_1 and public key (P_1, Y_1) where $Y_1 = x_1 P_1$, but no signer key is generated as before. In fact, the part of the signer is absent in this new game. The adversary is allowed to make endorsement queries on any condition statement c_j of his choice as before to obtain a witness signature $\sigma_j = x_1 H(c_j)$. When the adversary is ready for a challenge, it outputs a random number a and a condition statement c. The challenger flips a coin $b \in \{0, 1\}$ and outputs $\delta_b = (z_b, U)$ as the challenge, where $z_b = f(a_b)\hat{e}(P_1, H(c))^{r x_1}$ and $U = rP_1$ for some random r picked by the challenger but unknown to the adversary. When $b = 0$, the challenger sets $a_b = a$; when $b = 1$, the challenger randomly picks a' and sets $a_b = a'$. The adversary $\mathcal{A}$ has to output a guess b' for b and its advantage is defined as $Adv_{\mathcal{A}}^{GameA} = \left|Pr[b' = b] - \frac{1}{2}\right|$. The adversary $\mathcal{A}$ wins if $Adv_{\mathcal{A}}^{GameA}$ is non-negligible in the security parameter λ.

We could prove that the CVS construction for Elgamal signatures is simulatable with respect to Fake if the probability of winning $GameA$ is negligible for all PPT adversaries. The argument is as follows: We could show by contradiction. Suppose $GameA$ is hard but there is a PPT distinguisher D which could break the simulatability property of the CVS construction, we could construct D' based on D to win $GameA$. First, D' generates the needed signer public/private keys and pass them to D together with the witness public key it gets from its challenger. When there is a signing query from D, D' creates a partial signature itself. When there is an endorsement query, D' queries its challenger and relays the reply back to D. Finally, D outputs a message m and a condition statement c to be challenged. D' creates an Elgamal signature (γ, a) on m and outputs a and c as its own challenge request. When D' gets its challenge $\delta_b = (z_b, U)$, it passes (γ, z, U) as a challenge to D. Note that there is only one possible value for a in $\mathbb{Z}_p^*$ (the one picked by D') that would fit γ to satisfy the verification equation of the Elgamal scheme; hence, for any a' picked by the challenger, it will not satisfy the Elgamal verification equation. In other words, when $b = 0$, the challenge δ_b is a CVS partial signature for message m and condition statement c, otherwise, δ_b is indistinguishable from a simulator output for c. This thus perfectly simulate a challenge for D. Finally, D outputs its guess b' of b; D' outputs b' as its guess. Obviously, if D can break the simulatability property with non-negligible advantage Adv_D^{Sim}, then D' can win $GameA$ with the same advantage, that is, $Adv_{D'}^{GameA} = Adv_D^{Sim}$. This concludes the reduction.

We could show that if H is a random oracle, making polynomially many endorsement queries of $c_j \neq c$ (where c is the challenged condition statement) would not help in winning $GameA$. The steps are similar to those in Boneh and Franklin's IBE [2]. In details, we define a new game $GameB$ and show that the difficulty of winning $GameB$ implies the difficulty of winning $GameA$. A detailed description of $GameB$ is as follows:

$\underline{GameB}$
The challenger picks a witness public key (P, Y) where $Y = xP$ for some randomly picked $x \in \mathbb{Z}_q^*$. Then the challenger picks a random $Q \in \mathbb{G}_1$, and gives (P, Y) and Q to the adversary. The adversary outputs a number $a \in \mathbb{Z}_p$ to be challenged. The challenger flips a coin $b \in \{0, 1\}$ and outputs $\varphi_b = (f(a_b)\hat{e}(P, Q)^{xr}, rP)$ for a randomly picked $r \in \mathbb{Z}_q^*$. When $b = 0$, $a_b = a$; when $b = 1$, the challenger randomly picks $a' \in \mathbb{Z}_p$ and sets $a_b = a'$. Finally, the adversary has to output a guess b' for b. The adversary wins the game if $b' = b$ and its advantage is defined as $Adv_{\mathcal{A}}^{GameB} = \left|Pr[b' = b] - \frac{1}{2}\right|$. $GameB$ is said to be hard if $Adv_{\mathcal{A}}^{GameB}$ is negligible in the security parameter λ for all PPT adversaries.

We now show how an adversary D with a non-negligible advantage of winning $GameA$ could be used to construct another adversary D' for $GameB$ if H is a random oracle and why other (polynomially many) endorsement queries would help considerably in help in solving $GameA$ for a particular c.

D' gets from its challenger Q and the witness public key (P, Y) where $Y = xP$ for some unknown $x \in \mathbb{Z}_q^*$ and gives D the public key (P, Y) as well as the random oracle hash function H. Note that in the random oracle model, D is forced to query an oracle under full control of D' in order to evaluate H. Here H is controlled by D' as described below; how the endorsement queries are handled is also described.

H **Queries:** D can query the random oracle H at any time. To respond to these queries, D' maintains a list $H - list$ of tuples $\langle c_j, Q_j, t_j, coin_j \rangle$ whose details are as follows. Note that $c_j \in \{0, 1\}^*$ is the condition statement, Q_j is the response $H(c_j)$, $t_j \in \mathbb{Z}_q^*$, and $coin_j \in 0, 1$. Initially, $H - list$ is empty. When D queries the oracle H with a condition statement c_j, D' responds as follows:

1. All the previous queries are kept in $H - list$; if the current query c_j is in the list, return the previous response $H(c_j) = Q_j$.
2. If not, it generates a new one as follows: it first picks a random number $t_j \in \mathbb{Z}_q^*$ and then flips a coin $coin_j$ so that $Pr[coin_j = 0] = \alpha$. If $coin_j = 0$, it computes $Q_j = t_j Q$ returning $H(c_j) = t_j Q$; otherwise, it computes $Q_j = t_j P$ returning $H(c_j) = t_j P$. The new entry $(c_j, Q_j, t_j, coin_j)$ is added to $H - list$. It is clear that D' cannot distinguish the query output from a random one.

Endorsement Queries: When D' is asked for an endorsement query c_j, it responds as follows: If c_j is in the $H - list$, it retrieves the corresponding tuple, otherwise generates a new one and adds the tuple back to the $H - list$. Note that if $coin_j = 1$, D' could answer the query, otherwise, this run fails. The response of D' to endorsement queries is described below.

1. If $H(c_j) = Q_j = t_j Q$, this run of D' fails.
2. Otherwise, $H(c_j) = t_j P$ and D' returns the query result $xH(c_j) = t_j Y$. Note that $t_j Y = t_j xP = xt_j P = xH(c_j)$.

Then, D outputs a number $a \in \mathbb{Z}_p^*$ and a condition statement c for challenge. D' looks up the $H - list$ for c; if the random coin in the tuple is 1, then $H(c) = tP$ (for some $t \in \mathbb{Z}_q^*$) and D would

not help in solving $GameB$ and this runs of D' fails. Otherwise, $H(c) = tQ$, and D' sends out a as a challenge request to its challenger which return the challenge $\varphi_b = (z_b, U) = (f(a_b)\hat{e}(P,Q)^{xr}, rP)$ where $a_b = a$ when $b = 0$ and $a_b = a'$ when $b = 1$ for some unknown random number a'. D' computes $V = t^{-1}U$ sends out $\delta_b = (z_b, V)$ as a challenge to D. It could be seen that $V = t^{-1}U = t^{-1}rP$ and $\hat{e}(P, H(c))^{t^{-1}rx} = \hat{e}(P, tQ)^{t^{-1}rx} = \hat{e}(P,Q)^{rx} = z_b$; hence, δ_b is a valid challenge to D. D' continues answer queries as before. Finally, D outputs its guess b' for b and returns b' as its own guess.

If D' does not abort during the simulation, the adversary environment viewed by D is identical to its view in the real attack, and the $Adv_{D'}^{GameB} = Adv_{D}^{GameA}$. What remains is to calculate the probability that D' aborts during the simulation. Suppose D' makes at most q_E endorsement queries. The probability that D' does not terminate equals to $p_{succ} = \alpha(1-\alpha)^{q_E}$. Note that $p_{succ} \leq \left(\frac{1}{1+q_E}\right)\left(1 - \frac{1}{1+q_E}\right) = \frac{1}{e(1+q_E)}$ (where e is the base of natural logarithm), and by choosing δ properly, we could achieve this maximum probability of successfully running D'. Taking the optimal δ, we have $Adv_{D'}^{GameB} = \frac{1}{e(1+q_E)} Adv_{D}^{GameA}$.

We now can show that $GameB$ is hard based on the decisional bilinear Diffie Hellman (DBDH) assumption. We show how an adversary D which have a non-negligible advantage of winning $GameB$ could be used to solve the DBDH problem in the following construction of D': Given a problem instance (P, xP, yP, zP, e) for the DBDH problem, D' sets $(P, Y) = (P, xP)$ and $Q = yP$ and sends them to D. D outputs a number a for challenge. In return, D' sets $\psi_b = (f(a)e, zP)$ as a challenge for D. Finally, D outputs its guess b'. If $b' = 0$, D' outputs that (P, xP, yP, zP, e) is a BDH tuple, otherwise, it outputs not. Note that, if $e = \hat{e}(P,P)^{xyz}$, then $f(a)e = f(a)\hat{e}^{xyz}$ and $\psi_b = \varphi_0$ in $GameB$. Whereas, if $e \neq \hat{e}(P,P)^{xyz}$, $f(a)e$ could be re-written as $f(a)e = f(a')\hat{e}(P,P)^{xyz}$ for some unknown a' where $f(a') = \frac{f(a)e}{\hat{e}(P,P)^{xyz}}$ and $\psi_b = \varphi_1$ in $GameB$. This is perfectly simulated adversary environment the same as that in $GameB$. The advantage $Adv_{D'}^{DBDH}$ of D' in solving the DBDH problem is the same as Adv_{D}^{GameB}.

Putting the pieces together, the hardness of the $DBDH$ problem implies the hardness of $GameB$ which in turn implies the hardness of $GameA$ if q_E is polynomial. Finally, the hardness of $GameA$ implies the hardness of breaking the simulatability property of the Elgamal CVS construction. Overall, if D could break the simulatability property of the CVS construction with a non-negligible advantage Adv_{D}^{Sim}, then there exists an algorithm D' (constructed based on D) which could solve the decisional bilinear Diffie Hellman problem with an advantage $Adv_{D'}^{DBDH} = \frac{1}{e(1+q_E)} Adv_{D}^{Sim}$.

Security Analysis for the Multiple-Witness Case

We prove by contradiction. We assume the simulatability property is achieved in the single witness case. Suppose there is a PPT distinguisher D_N which can break the simulatability property for $N > 1$ where N is the number of witnesses. We show how to construct another distinguisher D_1, based on D_N, which could break the simulatability of the single-witness case, that is, distinguishing which of the following two distributions a given tuple (γ, z, U) belongs to:

- $\mathcal{CVS}(1) = \left\{(\gamma, z, U) : \gamma \leftarrow G; a = Dlog_{\gamma}(g^{h(m)} y_s^{\gamma}); r \leftarrow \mathbb{Z}_q^*; U = rP_1; z = f(a) y_1^r\right\}$
- $\mathcal{FAKE}(1) = \left\{(\gamma, z, U) : \gamma \leftarrow G; a \leftarrow \mathbb{Z}_p^*; r \leftarrow \mathbb{Z}_q^*; U = rP_1; z = f(a) y_1^r\right\}$

where (P_1, Y_1) is the public key of the witness and $y_1 = \hat{e}(Y_1, H(c))$ for a condition statement c.

The construction of D_1 (based on D_N) is as follows:

In the setup, D_1 asks its challenger for the signer's private and public keys $sk_S = x_s$ and $PK_S = y_s$ respectively, and the witness public key $PK_1 = (P_1, Y_1)$ where $Y_1 = x_1 P_1$ for some unknown $x_1 \in \mathbb{Z}_q^*$. Without loss of generality, we set this as W_1 for the multiple-witness case. Then, D_1 creates the public and private keys for other witnesses $W_i, 2 \leq i \leq N$ as follows: Uniformly pick random $x_i, t_i \in \mathbb{Z}_q^*$ and compute $P_i = t_i P_1$ and $Y_i = x_i P_i$. The public key for witness W_i is $(P_i, Y_i) = (t_i P_1, x_i t_i P_1)$. Since t_i and x_i are randomly picked, the resulting distribution of the public/private keys of each one of the last $N-1$ witnesses are the same as that generated by CVKGW.

D_1 answer queries from D_N in the following way: When D_N makes a signing query, D_1 creates a partial signature itself as it knows the signer's private key x_s. To answer any endorsement queries on a condition statement for witness W_1, D_1 makes an endorsement query to its challenger on the same condition statement and passes the result back to D_N. For the endorsement queries for other witnesses $W_i, 2 \leq i \leq N$, D_1 answers them itself using the private key x_i.

When D_N outputs a message m and a condition set $C = \{(c_i, W_i) : 1 \leq i \leq N\}$ asking for a challenge, D_1 outputs m and (c_1, W_1) as its challenge request. It is possible that (c_1, W_1) has been queried before as there is no restriction in our definition of simulatability that (c_1, W_1) has to be a new one; at least one of (c_i, W_i) not previously queried would constitute a valid challenge request. We will discuss later about abortion probability of this. Let us continue assuming (c_1, W_1) is a new condition. D_1 receives its challenge $\delta_b = (\gamma, z, U_1)$ where $\delta_b \in \mathcal{CVS}(1)$ when $b = 0$ and $\delta_b \in \mathcal{FAKE}(1)$ when $b = 1$. Note that $U_1 = rP_1$ for some unknown $r \in \mathbb{Z}_q^*$ and $z = f(a_b) y_1^r = f(a_b) e_1^{x_1 r}$ (where $y_1 = \hat{e}(Y_1, H(c_1))$ and $e_1 = \hat{e}(P_1, H(c_1))$) with a_0 being part of a Elgamal signature and a_1 some randomly picked number. D_1 computes the following for $2 \leq i \leq N$: $U_i = t_i U_1$ (Note the $t_i U_1 = t_i r P_1 = r P_i$.), $y_i = \hat{e}(U_i, H(c_i))^{x_i} = \hat{e}(P_i, H(c_i))^{r x_i}$ (using the secret keys $x_i, 2 \leq i \leq N$) and $z' = z \prod_{i=2}^N y_i^r = f(a_b) y_1^r \prod_{i=2}^N y_i^r = f(a_b) \prod_{i=1}^N y_i^r$. D_1 passes the following as a challenge for D_N: $\delta_b' = (\gamma, z', U_1, U_2, \ldots, U_N)$. It could be seen that $\delta_b' \in \mathcal{CVS}(N)$ if $\delta_b \in \mathcal{CVS}(1)$ ($b = 0$), whereas $\delta_b' \in \mathcal{FAKE}(N)$ if $\delta_b \in \mathcal{FAKE}(1)$ ($b = 1$).

D_N could continue making signing and endorsement queries. If (c_1, W_1) is in the query, then this run fails. Otherwise, when D_N outputs its guess b' for b, D_1 outputs b' as its guess for b. Obviously, if D_N is PPT, so is D_1 and the advantage of D_1 is the same as that of D_N, that is, $Adv_{D_1}^{Sim} = Adv_{D_N}^{Sim}$, provided D_1 does not abort in the simulation. Now, it remains to find out the probability of success of D_1. Note that no matter how many queries out of the requested challenge condition set $\{(c_i, W_i) : 1 \leq i \leq N\}$ are made by D_N, D_N must answer at least one of them directly according to the definition. In that case, if (c_1, W_1) is in the remaining subset, D_1 makes a successful run, and the probability of that is $p_{succ} = \frac{1}{N}$. Overall, the advantage of D_1 is $Adv_{D_1}^{Sim} = \frac{1}{N} Adv_{D_N}^{Sim}$. Taking the results of the single witness case here, if there exists a PPT distinguisher making q_E endorsement queries in breaking the simulatability of the Elgamal CVS construction with N witnesses with a non-negligible advantage $Adv_D^{Sim}(N)$, then there exists D' which could solve the DBDH problem with an advantage $Adv_{D'}^{DBDH} = \frac{1}{Ne(1+q_E)} Adv_D^{Sim}(N)$. As a result, if Adv_D^{DBDH} is negligible in the security parameter λ for all PPT algorithm D, so is $Adv_D^{Sim}(N)$ provided both N and q_E are polynomially many. In fact, in the real cases, N would usually be a very small integer, usually < 10, so the restriction would be fulfilled without mentioning. ■

Proof of the Zero Knowledge Property of the Confirmation Protocol (Claim 22)

The confirmation protocol for the Elgamal based CVS construction satisfies the property of completeness, soundness, and zero knowledge as follows.

- **Completeness.** Since the signer knows r and a, he could always computes (in response to the challenge b) θ in step 3 which passes the verification in step 4 provided he follows all the steps. Considering $b = 0$, $\theta = u$, then

$$e_i^\theta w_i^{-b} = e_i^\theta = e_i^u = t_i;$$
$$\gamma^{(1-b)f^{-1}(z\prod_{i=1}^N y_i^\theta)}\psi^{bf^{-1}(\prod_{i=1}^N y_i^\theta)} = \gamma^{f^{-1}(z\prod_{i=1}^N y_i^u)} = t.$$

  Considering $b = 1$, $\theta = u + r$, then

$$e_i^\theta w_i^{-b} = e_i^\theta = e_i^{u+r} w_i^{-1} = e_i^u e_i^r e_i^{-r} = t_i;$$
$$\gamma^{(1-b)f^{-1}(z\prod_{i=1}^N y_i^\theta)}\psi^{bf^{-1}(\prod_{i=1}^N y_i^\theta)} = \psi^{f^{-1}(\prod_{i=1}^N y_i^{u+r})};$$
$$t = \gamma^{f^{-1}(z\prod_{i=1}^N y_i^u)} = \gamma^{f^{-1}(f(a)\prod_{i=1}^N y_i^r \prod_{i=1}^N y_i^u)} = \gamma^{f^{-1}(f(a))f^{-1}(\prod_{i=1}^N y_i^{u+r})} = (\gamma^a)^{f^{-1}(\prod_{i=1}^N y_i^{u+r})}.$$

  Note for the case $b = 1$, we use the homomorphic property that $f^{-1}(f(a)e) = af^{-1}(e)$, $\forall a \in \mathbb{Z}_p, e \in \mathbb{F}_p^l$. If a valid signature on m could be recovered from δ, then γ^a should be equal to $g^{h(m)}y_s^\gamma$ which is equal to ψ. Hence, $t = \psi^{f^{-1}(\prod_{i=1}^N y_i^{u+r})}$ which concludes for the case $b = 1$.

- **Soundness.** Suppose the set of equations does not hold, in particular, $f^{-1}(\frac{z}{\prod_{i=1}^N y_i^r}) \neq a$ (where $\gamma^a = \psi$) or $\exists i,\ U_i \neq rG_i$. By following the protocol procedures, the signer could always give a correct response θ when the challenge is 0. However, when the challenge is 1, to pass the test in step 4, the signer needs to find a solution θ for either $e_i^\theta = t_i w_i$ or $f^{-1}(z\prod_{i=1}^N y_i^u) = af^{-1}(\prod_{i=1}^N y_i^\theta)$. Either one is equivalent to the DL problem in $\mathbb{G}_2$.

  On the other hand, suppose now the signer bet that the challenge will be 1, he tries to deviate from the protocol. He could randomly pick a θ and compute t_i and t which satisfy the verification equation in step 4, and give these t_i and t to the verifier in step 2. However, if the challenge is 0, he could not find u satisfying the equations $e_i^u = t_i = e_i^\theta w_i^{-1}$ which is the DL problem again.

  Hence, the signer could cheat successfully in each round with a probability of success equal to $\frac{1}{2}$.

- **Zero-Knowledge.** We need to find a PPT simulator which can simulate the output transcript without any interaction with the signer. As in the soundness part, the simulator even having no knowledge about a and r could always give a correct response to a prepared challenge (out of the two possible challenges) in each round. Using this strategy with the standard rewinding technique, the simulator could generate a transcript indistinguishable from the true transcript recorded during a confirmation protocol run. The operation is as follows: The simulator runs the signer and verifier algorithms of the confirmation protocol to emulate a proof carried out. In each round, if the challenge is the same as the prepared one, the simulator goes on to the next round, otherwise, it rewinds the protocol back to the start of the current round and starts with a new prepared challenge. On average, 2 iterations would enable the simulator to complete the generation of one round of transcript. Hence, the transcript simulator is PPT.

Security Analysis for the Pairing-based CVS Construction for RSA Signatures

Proof of the Simulatability Property (Claim 23)

The steps of proving that the RSA CVS construction is simulatable is almost the same as that in the Elgamal CVS construction. What we need to show is the hardness of $GameA$ implies simulatability in the single witness case described below.

Proof

Security Analysis for the Single-Witness Case

We could prove that the CVS construction for RSA signatures is simulatable with respect to Fake if the probability of winning $GameA$ is negligible for all PPT adversaries. The argument is as follows: We could show by contradiction. Suppose $GameA$ is hard but there is a PPT distinguisher D which could break the simulatability property of the CVS construction, we could construct D' based on D to win $GameA$. First, D' generates the needed signer public/private keys and pass them to D together with the witness public key it gets from its challenger. When there is a signing query from D, D' creates a partial signature itself. When there is an endorsement query, D' queries its challenger and relays the reply back to D. Finally, D outputs a message m and a condition statement c to be challenged. D' creates an RSA signature σ on m and randomly picks a number $\alpha \in \mathbb{Z}_n^*$ to create $\gamma = \alpha\sigma$. D' outputs $a = f_1(\alpha)$ and c as its own challenge request. When D' gets its challenge $\delta_b = (z_b, U)$, it passes (γ, z, U) as a challenge to D. Note that there is only one possible value for a in $\mathbb{Z}_p^*$ (the one picked by D') that would have $f^{-1}(a) = \alpha$ able to retrieve σ from γ; hence, for any a' picked by the challenger, it will not recover an RSA signature satisfying the RSA verification equation. In other words, when $b = 0$, the challenge δ_b is a CVS partial signature for message m and condition statement c, otherwise, δ_b is indistinguishable from a simulator output for c. This thus perfectly simulate a challenge for D. Finally, D outputs its guess b' of b; D' outputs b' as its guess. Obviously, if D can break the simulatability property with non-negligible advantage Adv_D^{Sim}, then D' can win $GameA$ with the same advantage, that is, $Adv_{D'}^{GameA} = Adv_D^{Sim}$. This concludes the reduction.

Security Analysis for the Multiple-Witness Case

The reduction is the same as that in the Elgamal CVS construction by replacing f by $f \circ f_1$.

■

Proof of the Zero Knowledge Property of the Confirmation Protocol (Claim 24)

The confirmation protocol satisfies the completeness, soundness and zero knowledge properties as follows:

- **Completeness.** Since the signer knows r and a, he could always computes (in response to the challenge b) θ in step 3 which passes the verification in step 4 provided he follows all the steps. Considering $b = 0$, then $\theta = u$ and $\psi = v \bmod p$, then

$$\begin{aligned} &s(\gamma^e)^b = s = v^e \bmod n = (\psi \bmod n)^e \bmod n, \\ &e_i^\theta w_i^{-b} = e_i^\theta = e_i^u = t_i \text{ and} \\ &f(\psi) \textstyle\prod_{i=1}^N y_i^\theta = f(v \bmod p) \prod_{i=1}^N y_i^u = t. \end{aligned}$$

  Considering $b = 1$, then $\theta = u + r$ and $\psi = (a \bmod n)(v \bmod n) \bmod p$, then

$$\begin{aligned} &s(\gamma^e)^b \bmod n = s\gamma^e \bmod n = v^e(a\sigma)^e \bmod n = v^e a^e \sigma^e \bmod n = (av)^e \sigma^e \bmod n \\ &\qquad = (\psi \bmod n)^e h(m) \bmod n, \\ &e_i^\theta w_i^{-b} = e_i^\theta = e_i^{u+r} w_i^{-1} = e_i^u e_i^r e_i^{-r} = t_i, \\ &f(\psi) \textstyle\prod_{i=1}^N y_i^\theta = f(av \bmod p) \prod_{i=1}^N y_i^{u+r} = f(a \bmod p) f(v \bmod p) \prod_{i=1}^N y_i^u \prod_{i=1}^N y_i^r \\ &\qquad = \left(f(a \bmod p) \textstyle\prod_{i=1}^N y_i^r\right) \left(f(v \bmod p) \prod_{i=1}^N y_i^u\right) = zt. \end{aligned}$$

  Note here we use the homomorphic property that $f(a_1 a_2) = f(a_1) f(a_2),\ \forall a_1, a_2 \in \mathbb{Z}_p$.

- **Soundness.** Suppose the set of equations does not hold. By following the protocol procedures, the signer could always give a correct response θ when the challenge is 0. However, when the challenge is 1, to pass the test in step 4,there is only a single ψ which could satisfy the first equation. To find a value of θ which could satisfy the last two equation with a fixed ψ, the signer needs to solve the DL problem. Unlike the ElGamal case discussed previously, there is only a single θ which is the right answer and finding it is equivalent to the DL problem.

  On the other hand, suppose now the signer bet that the challenge will be 1, he tries to deviate from the protocol. He could randomly pick θ and ψ to compute a set of values for s, t_i and t satisfying the verification equation in step 4, and give these s, t_i and t to the verifier in step 2. Since there are three constraint equation for two variables, there is a single set which works for a particular choice of (θ, ψ). If the challenge is 0, to satisfying the first equation for signature verification, he needs to find u to make the last two equations hold, which is equivalent to the DL problem again.

  As a result, the signer could cheat successfully in each round with a probability of success equal to $\frac{1}{2}$.

- **Zero-Knowledge.** As in the soundness part, the verifier could always give a correct response of a prepared challenge (out of the two possible challenges) in each round. Using this strategy with the standard rewinding technique, the signer could simulate a transcript that is indistinguishable from the true transcript recorded during a real confirmation protocol run. The construction is similar to that in the ElGamal based CVS construction.