Practical Covert Authentication

Stanislaw Jarecki

University of California Irvine, stasio@ics.uci.edu

Abstract. Von Ahn, Hopper, and Langford [vAHL05] introduced the notion of two-party *steganographic* a.k.a. *covert* computation, which assures that neither party can distinguish its counterparty from a random noise generator, except for what is revealed by the final output of the securely computed function. The flagship motivation for covert computation is *covert authentication*, where two parties want to authenticate each other, e.g. as some credential holders, but a party who lacks the credentials is not only unable to pass the authentication protocol, but cannot even distinguish a protocol instance from random noise.
Previous work on covert computation [vAHL05,CGOS07] showed general-purpose protocols whose efficiency is linear in the size of the circuit representation of the computed function. Here we show the first practical (assuming a large-enough random steganographic channel) covert protocol for the specific task of two-party mutual authentication, secure under the strong RSA, DQR, and DDH assumptions. The protocol takes 5 rounds (3 in ROM), $O(1)$ modular exponentiations, and supports revocation and identity escrow. The main technical contribution which enables it is a compiler from a special honest-verifier zero-knowledge proof to a *covert conditional key encapsulation mechanism* for the same language.

1 Introduction

Steganography addresses a security/privacy property which is not usually considered in cryptography, which is how to make the very fact of secure protocol execution hidden from the adversary. Such hiding of a protocol instance is in principle possible if the public channels connecting the communicating parties are *steganographic* in the sense that they have some intrinsic entropy. A protocol is steganographic, or *covert*, if its messages can be efficiently injected into such channels in a way that the resulting communication cannot be distinguished from the (assumed) a priori random behavior of these channels. A simple example of a steganographic channel is a *random channel*, which can be implemented e.g. using protocol nonces, random padding bits, lower bits of time stamps, and various other standard communication mechanisms which exhibit inherent entropy. Assuming such random channels between two parties $A \to B$ and $B \to A$, party A would encode its protocol messages as bitstrings which are indistinguishable from random, inject its out-going messages into the $A \to B$ channel, and interpret the messages on the $B \to A$ channel as B's responses in the protocol. A and B must synchronize the timing of using these channels, so they know which

bits to interpret as protocol messages, but this can be public information: The covertness of the protocol implies that the messages which A and B exchange cannot be distinguished from the a priori behavior of these channels.

Covert computation was formalized for two parties in [vAHL05] and in the multi-party setting in [CGOS07] as a protocol that lets the participants securely compute the desired functionality on their inputs, with the additional property that no party can distinguish the other participants from "random beacons" which send random bitstrings of fixed length instead of proscribed protocol messages, except for what is revealed by the final output of the computed function. Both [vAHL05] and [CGOS07] show protocols for covert computation of any functionality which tolerate malicious adversaries, resp. in the two-party and the multi-party setting, but the costs of these protocols are linear in the size of the circuit representation of the computed function. Moreover, these protocols are not constant-round, and the subsequent work of [GJ10] showed that this is a fundamental limitation on maliciously-secure covert computation in the standard model, i.e. without access to trusted parameters or public keys. Still, this begs the question whether useful two-party (or multi-party) tasks can be accomplished covertly in a more practical way, with constant-round protocols and constant number of public-key operations, in applications where common trusted parameters and/or public keys are naturally available.

Indeed, the flagship motivation for covert computation, including [vAHL05] and [CGOS07], was *covert authentication*, where two parties want to authenticate each other, e.g. as holders of mutually accepted certificates, but a party who lacks proper certificate is not only unable to pass in the authentication protocol, but cannot even distinguish an instance of such protocol from a random beacon. In this work we show the first practical covert Mutual Authentication (MA) protocol for the setting where mutually accepted certificates are defined as group membership certificates issued by the same group manager. A very similar mutual authentication setting was considered by "Secret Handshakes" a.k.a. *Private Mutual Authentication*, see e.g. [JL09], but the goal of *private* authentication is to protect the privacy of all authentication protocol inputs, including the group public key assumed by each party in the protocol, while *covert* authentication goes a step further, and aims to hide the very fact that the authentication protocol takes place.

Our covert MA protocol relies on a covert *Conditional Key Encapsulation Mechanism* (CKEM), a covert variant of Conditional Oblivious Transfer [COR99] and a variant of the *ZKSend* gadget used in [CGOS07]. A covert CKEM is a steganographic form of a zero-knowledge proof: It establishes a shared key between the prover and the verifier if and only if it is run on a true statement. Unlike a zero-knowledge proof which involves an explicit verification which distinguishes the prover from a random beacon, a CKEM instance could appear indistinguishable from a random noise to either participant. We show an efficient compiler which converts a special Σ-protocol, i.e. a public-coin HVZK proof of knowledge with certain (commonly satisfied) additional properties, into a covert CKEM for the same language. A key property of this compiler is that it con-

structs a CKEM with a proof-of-knowledge property, which ensures extraction of a witness for a verifier's statement given a prover who distinguishes the verifier from a random beacon. Witness-extraction makes covert CKEM's more useful as protocol building blocks, as we exemplify in our covert MA construction below.

Our covert MA scheme requires a group signature scheme which works by committing to a group membership certificate and then proving in ZK that the committed certificate is valid under the group public key. If the commitment is covert and the ZK proof is replaced with a covert CKEM for the same language, the result is a covert MA scheme. Crucially, if the CKEM enables extraction of a witness given an adversary who breaks protocol covertness, then a security reduction can extract a new membership certificate (and thus break the unforgeability of the underlying group signature scheme) given an adversary who distinguishes an MA counterparty from a random beacon.

Note that covert CKEM's without the proof-of-knowledge property, for relations involving discrete logarithm equalities, can be implemented using a Smooth Projective Hash Function (SPHF) [CS01], if the verifier sends to the prover the projection key, which is usually a tuple of random group elements, and so it can be encoded as a random bitstring. What makes our covert CKEM construction interesting is its proof-of-knowledge property, which is achieved as follows: On statement x, the prover covertly commits to its first message a as C, sends response z to the verifier's challenge c, and then the two parties run an SPHF on the statement that the prover's presumed first message a, which can be computed from (x, c, z), is indeed committed in C. Simulation follows from the covertness of the commitment, and extraction follows by the standard rewinding technique from the binding property of the commitment. The "special" property of the Σ-protocol required by our CKEM construction is that a can be efficiently computed given (x, c, z), and that z is an integer tuple distributed statistically close to uniform over some integer ranges (and thus can be encoded as a random bitstring), which is commonly the case in Σ-protocols for various arithmetic relations on discrete logarithm and representations.

Organization. Section 2 introduces basic concepts and tools related to covert computation. In Section 3 we define a covert CKEM and a covert MA scheme. In Section 4 we construct a covert CKEM for any language which admits a special Σ-protocol. In Section 5 we construct a covert MA scheme from (an interactive version of) a group signature and a covert CKEM for a related language. In Section 5.1 we instantiate this construction with the group signature of [ACJT00].

2 Preliminaries

Covertness. The paradigm of covert computation used in [vAHL05,CGOS07], as well as in the work on steganographic key exchange of [vAH04], assumes that the participants in a covert protocol are connected by a channel with sufficient entropy, henceforth called a *steganographic* channel, and that the participants communicate by using a steganographic algorithm, e.g. [HLvA02], to embed protocol messages into this steganographic channel. As was shown by Ahn et al.

[vAHL05], if a protocol is covert for a *random channel*, i.e. if its messages are indistinguishable from random bitstrings of some fixed length, then applying a steganographic encoding to each protocol message makes the protocol covert for the corresponding steganographic channel. Consequently we can limit our goal to creating protocols whose messages are indistinguishable from random bitstrings. Moreover, many steganographic channels are already uniform over fixed-length bitstrings, e.g. a channel provided by random nonces in TCP/IP control packets, in which case the steganographic encoding consists of simple splitting of protocol messages into segments of length dictated by this channel. We use the following notation to capture indistinguishability of a protocol participant from a random beacon, i.e. a source that broadcasts random bitstrings of fixed length. Let A be an interactive algorithm which engages in a fixed number k of protocol rounds in each protocol instance, and where for each $i = 1, .., k$, A's i-th message is a bitstring of length $u_i(\tau)$, where u_i is a polynomial and τ is a security parameter. Let $u = (u_1, ..., u_k)$. We denote by $A^{\$(u)}$ an interactive protocol which takes k rounds s.t. its i-th outgoing message is a random bitstring of length $u_i(\tau)$.

Covert Encodings. Our goal is to create efficient protocols whose messages are indistinguishable from random bitstrings of fixed length. We will accomplish this by designing protocols which communicate values which are indistinguishable from either random group elements or random integers on integer intervals, and then encoding these as fixed-length bitstrings using randomized encodings. Let $|R|$ denote the bit-length of R, and let $[R]$ and $\pm[R]$ denote sets $\{0, \ldots, R-1\}$ and $\{-R+1, \ldots, R-1\}$, respectively. Encoding $\mathsf{EC}_{[R]}$ maps $v \in [R]$ to an $(|R|+\tau)$-bit string by outputting $\overline{v} = v + Rk$ (over integers) for random k in $\{0, \ldots, \lfloor 2^{|R|+\tau}/R \rfloor\}$. Decoding $\mathsf{DC}_{|R|}(\overline{v})$ outputs $v = \overline{v} \bmod R$. Encoding $\mathsf{EC}_{\pm[R]}$ maps $v \in \pm[R]$ to an $(|2R-1|+\tau)$-bit string by outputting $\mathsf{EC}_{[2R-1]}(v+(R-1))$, while $\mathsf{DC}_{\pm[R]}$ reverses this process. Finally, if $\boldsymbol{I} = I_1 \times \ldots \times I_t$ is a cross-product of integer intervals then $\mathsf{EC}_{\boldsymbol{I}}$ maps $\boldsymbol{I}$ into bitstrings of length $|I_1|+\ldots+|I_t|+t\cdot\tau$ by outputting $\overline{v} = (\mathsf{EC}_{I_1}(v_1), \ldots, \mathsf{EC}_{I_t}(v_t))$ on input $v = (v_1, \ldots, v_t)$, while $\mathsf{DC}_{\boldsymbol{I}}$ reverses this process. All these encodings are covert on their respective message spaces in the following sense: $(\mathsf{EC}_S, \mathsf{DC}_S)$ is a *covert encoding* on space S if the distribution $\{\mathsf{EC}_S(v)\}_{v \leftarrow S}$ is statistically close to uniform over $\{0,1\}^t$ for some t.

3 Covert KEM and Authentication Definitions

Covert Conditional KEM. Conditional OT (COT) for an NP relation $\mathcal{R}$ (and an associated language $\mathcal{L}^{\mathcal{R}}$), introduced by Di Crescenzo et al. [COR99], is a pair of algorithms for sender S and receiver R, where S runs on a message m and a statement x and R runs on a witness w, s.t. the receiver learns m if $(x, w) \in \mathcal{R}$, while the sender learns nothing from the protocol. COT sender's *privacy* requires that the receiver learns nothing about *both* m *and* x unless $(x, w) \in \mathcal{R}$ [COR99,Cre00]. Since COT can be thought of as an interactive encryption, we introduce a KEM-like version of this notion, a *Conditional Key Encapsulation Mechanism* (CKEM). We define CKEM in a *public parameter model*, as a tuple $(\mathsf{PG}, \mathsf{S}, \mathsf{R})$ where S and R are interactive algorithms running on respective

inputs (π, x_{S}) and (π, x_{R}, w), for $\pi \leftarrow \mathsf{PG}(1^\tau)$. Both S and R output τ-bit keys, respectively K and K', s.t. key K generated by S is a random bitstring, while K' output by R is equal to K if $((\pi, x_{\mathsf{S}}), w) \in \mathcal{R}$ (and $x_{\mathsf{R}} = x_{\mathsf{S}}$), and independent from K if $((\pi, x_{\mathsf{S}}), w) \notin \mathcal{R}$. Note that CKEM implies COT if S encrypts its message m under key K. Jarecki and Liu [JL09] introduced *strong* sender security for COT, where an efficient extractor can extract w s.t. $(x_{\mathsf{S}}, w) \in \mathcal{R}$ from an adversary which breaks sender's security. We adapt this notion because witness-extraction makes CKEM into a more useful protocol building block. Indeed, our mutual authentication scheme of Section 5 relies on strong sender covertness of CKEM to enable the reduction to extract a valid certificate (and thus forge a certificate) from an adversary who breaks authentication security/privacy.

Definition 1 (Receiver Covertness). *A CKEM* $(\mathsf{PG}, \mathsf{S}, \mathsf{R})$ *for relation* $\mathcal{R}$ *(and language* $\mathcal{L}^{\mathcal{R}}$*) is* receiver covert *if for some polynomial sequence* $u = (u_1, u_2, ...)$*, for any efficient algorithm* $\mathcal{A}$*, the difference between the probability of* $\mathcal{A}$ *outputting* 1 *in the following two experiments is a negligible function of* τ*. Both experiments run* $\mathsf{PG}(1^\tau)$ *to choose parameter* π*, and* $\mathcal{A}(\pi)$ *chooses* (x, w)*, and then in the first experiment* $\mathcal{A}$ *interacts with* $\mathsf{R}(\pi, x, w)$*, while in the second experiment* $\mathcal{A}$ *interacts with* $\mathsf{R}^{\$(u)}$*.*

Definition 2 (Strong Sender Covertness). *A CKEM* $(\mathsf{PG}, \mathsf{S}, \mathsf{R})$ *for relation* $\mathcal{R}$ *(and language* $\mathcal{L}^{\mathcal{R}}$*) is* strong sender covert *if there is a polynomial sequence* u*, an efficient algorithm* Ext*, and a polynomial* p *s.t. for any efficient algorithm* $\mathcal{A}$ *there exists a negligible function* δ*, s.t. for any* τ*, any* π *output by* $\mathsf{PG}(1^\tau)$*, and any* x *of size polynomial in* τ*, it holds that* Ext *on input* (π, x) *and an oracle access to* $\mathcal{A}$ *outputs* w *s.t.* $((x, \pi), w) \in \mathcal{R}$ *with probability at least* $p(\epsilon_{\mathcal{A},\pi,x,u} - \delta(\tau))$*, where* $\epsilon_{\mathcal{A},\pi,x,u}$ *is defined as the difference between the probability that* $\mathcal{A}$ *outputs* 1 *in the following two games: In the "real" game,* $\mathcal{A}$ *interacts with* $\mathsf{S}(\pi, x)$ *and then receives key* K *output by this* S *instance, while in the "random" game,* $\mathcal{A}$ *interacts with* $\mathsf{S}^{\$(u)}$ *and then receives* K *generated as a random* τ*-bit string.*

Note on Computational Restrictions. We define CKEM covertness only for computationally bounded adversaries because our CKEM construction in Section 4 depends on these bounds in both directions. Receiver's covertness is computational because it encrypts the first Σ-protocol message using a commitment which is only computationally hiding/covert, while sender's covertness relies on collision-resistance of a hash function. (Additionally, the 2-round version of this CKEM, which works in the Random Oracle Model (ROM) for hash functions, requires a polynomial bound on the number of adversary's hash function queries.)

CKEM vs. Zero-Knowledge Proofs. One can view CKEM as an encryption counterpart to a Zero-Knowledge Proof, with S playing the role of the Verifier and R that of a Prover, except that in CKEM, the point is not for S to learn anything about statement x, but for R to receive S's key only if R has w s.t. $(x, w) \in \mathcal{R}$. In particular, one can view CKEM receiver privacy as a form of zero-knowledge and CKEM strong sender security as a form of strong soundness, i.e. a proof of knowledge. Indeed, both strong sender covertness and strong soundness of

an interactive proof require that if some algorithm $\mathcal{A}$ "ϵ-succeeds" on statement x, then an efficient extractor can use $\mathcal{A}$ to extract w s.t. $(x, w) \in \mathcal{R}$. In an interactive proof $\mathcal{A}$'s success is defined as convincing a verifier that $x \in \mathcal{L}^{\mathcal{R}}$, while CKEM covertness defines $\mathcal{A}$'s success as distinguishing an interaction with $\mathsf{S}(\pi, x)$ followed by the key K output by this instance of S, from an interaction with a random beacon followed by a random τ-bit string.

CKEM vs. SPHF. CKEM's can be seen as a generalization of Smooth Projective Hash Functions [CS01] to interactive protocols. An SPHF gives rise to a one-round CKEM by sending the projection key and treating the hash value as the key K. Such CKEM is covert if the projection key can be covertly encoded, but it is not *strongly* covert because it does not assure witness extraction.

CKEM vs. Covert 2PC. Our CKEM construction of Section 4 satisfies the above game-based CKEM definition, but it is not a covert secure computation of a CKEM functionality [vAHL05,CGOS07]. In particular, it enables extraction of the witness w input by R but not the statement x input by S.

Covert Mutual Authentication. Consider a group manager GM who issues certificates to group members and publishes revocation tokens for the users whose membership it wants to revoke. An (implicit) *Mutual Authentication* (MA) scheme, with verifier-local revocation, is a tuple of algorithms $(\mathsf{KGen}, \mathsf{CG}, \mathsf{Auth})$ which work as follows. KGen on security parameter τ outputs a master secret key msk and a public key mpk. To issue a membership certificate to user P_i, GM gives her a *certificate* generated as $(\mathsf{sk}_i, \mathsf{rt}_i) \leftarrow \mathsf{CG}(\mathsf{msk})$. To revoke membership, GM adds rt_i to an initially empty revocation list CRL, which should then be propagated to all current group members. If two players P_i and P_j want to authenticate to each other, each player follows the interactive algorithm Auth, where P_i runs on private inputs $(\mathsf{mpk}, (\mathsf{sk}_i, \mathsf{rt}_i), \mathsf{CRL})$ while P_j runs on $(\mathsf{mpk}', (\mathsf{sk}_j, \mathsf{rt}_j), \mathsf{CRL}')$. Each participant's local outputs is a τ-bit session key, respectively K and K'. If both parties follows the protocol then $K = K'$ if (1) $\mathsf{mpk} = \mathsf{mpk}'$, (2) both $(\mathsf{sk}_i, \mathsf{rt}_i)$ and $(\mathsf{sk}_j, \mathsf{rt}_j)$ are valid certificates under mpk, (3) neither certificate is revoked in the CRL of the other player, i.e. $\mathsf{rt}_i \notin \mathsf{CRL}'$ and $\mathsf{rt}_j \notin \mathsf{CRL}$.

Intuitively, we call an MA scheme *covert* if no one except a valid group member can distinguish an interaction in the authentication scheme with a member of the same group from an interaction with a random beacon. Formally, we define MA covertness via the following game between an adversary $\mathcal{A}$ and a game G. Let k be the number of message rounds in protocol Auth, and let $u = (u_1, u_2, ..., u_k)$ be some sequence of polynomials. The MA security experiment, denoted $G_{\mathcal{A}}(1^\tau, b)$, is defined by an interaction between game G and an attacker $\mathcal{A}$ which proceeds as follows:

Init. G on input $(1^\tau, b)$ for bit b sets $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{KGen}(1^\tau)$, $\mathsf{CRL} \leftarrow \emptyset$, and generates $(\mathsf{sk}_i, \mathsf{rt}_i) \leftarrow \mathsf{CG}(\mathsf{msk})$ for $i = 1, \ldots, N(\tau)$ for a fixed polynomial N.
Corruptions. $\mathcal{A}$, on input $(1^\tau, \mathsf{mpk})$, specifies a subset CorSet of corrupt players, and for each $i \in \mathsf{CorSet}$, $\mathcal{A}$ receives $(\mathsf{sk}_i, \mathsf{rt}_i)$, and rt_i is added to CRL.
Queries. $\mathcal{A}$ can (concurrently) make any number of Exec queries and a single Test query, to which G responds as follows:

$\mathsf{Exec}(i, \mathsf{CRL}^*)$: Execute $\mathsf{Auth}(\mathsf{mpk}, (\mathsf{sk}_i, \mathsf{rt}_i), \mathsf{CRL}^*)$, interacting with $\mathcal{A}$.
$\mathsf{Test}(i)$: If $i \notin \mathsf{CorSet}$, respond as follows:
If $b = 1$, execute $\mathsf{Auth}(\mathsf{mpk}, (\mathsf{sk}_i, \mathsf{rt}_i), \mathsf{CRL})$, interacting with $\mathcal{A}$, and send the local output K of this Auth instance to $\mathcal{A}$;
If $b = 0$, execute $\mathsf{Auth}^{\$(u)}$, and send a random τ-bit string K' to $\mathcal{A}$.
Guess. If $\mathcal{A}$ halts and outputs a bit, G halts and outputs the same bit.

Definition 3 (MA Covertness). *We call an MA scheme* $(\mathsf{KGen}, \mathsf{CG}, \mathsf{Auth})$ covert *if for some polynomial sequence* u *function* $\epsilon_{\mathcal{A}}(\tau) = |\Pr[G_{\mathcal{A}}(1^\tau, 0) = 1] - \Pr[G_{\mathcal{A}}(1^\tau, 1) = 1]|$ *is negligible for any efficient algorithm* $\mathcal{A}$.

Revocation and Escrow. The MA definition implies that $\mathcal{A}$ can corrupt or participate in Auth instances with any party, but this will not help $\mathcal{A}$ in distinguishing an Auth instance ran by a *non-corrupted* party from a random beacon. This can hold only if the honest party executes on a revocation list containing revocation tokens of all corrupted players. (Otherwise the adversary could run an Auth instance on a certificate of a corrupted player.) Note that we allow $\mathcal{A}$ to interact with Auth instances executing on wrong revocation lists, to model the fact that honest parties can execute on outdated or otherwise incorrect revocation lists. While such instances can be recognizable to $\mathcal{A}$, they should not endanger covertness of instances which use the correct revocation list. One limitation of our "verifier-local" revocation model, which we adopt from the work on group signatures by Boneh and Shacham [BS04], is the lack of "perfect-forward covertness", i.e. an adversary who learns some party's certificate can break the covertness of all past protocol instances executed by this party. We model this in the security experiment by requiring that the tested player is not on the revocation list. However, this revocation model naturally supports *identity escrow*, because GM can use revocation tokens to link protocol transcripts to users.

Authentication Security. MA covertness implies standard authentication security because an attacker without a valid certificate cannot distinguish the key output by a group member from a random string. However, our MA notion is quite far from a full-fledged Authenticated Key Exchange (AKE) [BCK98,CK01]. First of all, an adversary gets to see a session key only on a single tested session, so there are no guarantees of independence between keys created by different instances, and no guarantees of security against the man-in-the-middle attacks. In other limitations, we offer only static security, because all corruptions must precede protocol instance executions, and we offer limited security against malicious insiders, because we never expose the session keys on $\mathsf{Exec}(i, \mathsf{CRL}^*)$ instances.

4 Covert Conditional KEM Construction

We show a general compiler which uses a covert commitment with associated SPHF to convert a special Σ-protocol for a given language, a form of three-round public-coin Honest-Verifier Zero-Knowledge (HVZK) proof of knowledge, into a covert CKEM for the same language. When the covert commitment is

instantiated as we explain below, the CKEM construction relies on the DDH assumption on a prime-order subgroup of a prime residue group Z_p^*, and its cost is that of the underlying Σ-protocol plus 2 exponentiations in Z_p^* for the sender and 3 for the receiver, assuming that the encoding of bitstrings output by the CKEM into the underlying steganographic channel is not computationally intensive, e.g. because the underlying steganographic channel is a random channel. Below we first introduce our tools, the special Σ-protocol and the covert commitment with associated SPHF, and then we show the covert CKEM construction.

Special Σ-Protocol. The notion of *Σ-protocol* was used by Damgard (see e.g. [Dam10]) to describe common features of HVZK proof systems which extend Schnorr's proof of knowledge of the discrete logarithm to various arithmetic relations on discrete logarithms and representations. Let algorithm triple (P_1, P_2, V) define a 3-round public-coin proof system for relation $\mathcal{R}$, where P_1 on input $(x, w) \in \mathcal{R}$ and internal randomness r outputs the prover's first message a, P_2 on input (x, w, r) and a τ-bit challenge c outputs the prover's second message z, and V on input (x, a, c, z) outputs the verifier's accept/reject decision bit. We say that (P_1, P_2, V) is a *Special Σ-Protocol* for $\mathcal{R}$ if it satisfies the following additional properties: (1) ("special soundness") There exists an efficient extractor which outputs w s.t. $(x, w) \in \mathcal{R}$ given any two accepting transcripts that share the same prover's first message a but differ on the challenge c, i.e. given (x, a, c, z, c', z') s.t. $V(x, a, c, z) = V(x, a, c', z') = 1$ and $c' \neq c$; (2) The prover's second message z is a sequence of integers distributed statistically close to uniform over some integer ranges, i.e. for any $(x, w) \in \mathcal{R}$ and $c \in \{0,1\}^\tau$, the distribution of z's output by $P_2(x, w, r, c)$ on random r is statistically close to uniform over $\boldsymbol{I} = I_1 \times \ldots \times I_t$ for some integer ranges $I_1, \ldots, I_t$; (3) ("special simulation") There exists an efficiently computable function f_V s.t. $V(x, a, c, z) = 1$ iff $a = f_V(x, c, z)$, $c \in \{0,1\}^\tau$, and $z \in \boldsymbol{I'}$ for some cross-product of ranges $\boldsymbol{I'}$. These properties are satisfied by Σ-protocols for many relations on discrete logarithms and representations (see e.g. [CM99] for examples). Such Σ-protocols are usually given for prime order groups, but they extend to the QR_n subgroup of Z_n^* for a safe RSA modulus n, such as the Σ-protocol for ACJT group signature[ACJT00] possession (see Appendix A) used in the instantiation of our covert MA construction in Section 5.1.

Covert Commitment with Associated SPHF. We call a tuple of efficient algorithms $(\mathsf{PG}, \mathsf{Com}, \mathsf{Hash}, \mathsf{PHash})$ a *perfectly binding covert commitment with associated smooth projective hash function (SPHF)* if the following requirements are satisfied. (1) First, pair $(\mathsf{PG}, \mathsf{Com})$ is a *covert commitment* defined as follows: There is a polynomial $l(\cdot)$ s.t. for any efficient algorithm $\mathcal{A}$, quantity $|p_0 - p_1|$ is a negligible function of τ, where p_β is defined as the probability that $b = 1$ in the following experiment: Generate $\pi \leftarrow \mathsf{PG}(1^\tau)$, pick $\mathcal{A}$'s randomness r, and generate $m \leftarrow \mathcal{A}(\pi; r)$. If $\beta = 1$ generate $C \leftarrow \mathsf{Com}(\pi, m)$, otherwise $C \leftarrow \{0,1\}^{l(\tau)}$. Finally, let $b \leftarrow \mathcal{A}(\pi, C; r)$. Note that commitment covertness implies the standard notion of hiding for a commitment scheme. (2) Secondly, this commitment must be *perfectly binding*, i.e. for any τ, any $\pi \leftarrow \mathsf{PG}(1^\tau)$, any m, m', r, r', if $\mathsf{Com}((\pi, m); r) = \mathsf{Com}((\pi, m'); r')$ then $m = m'$. (3) Thirdly, $(\mathsf{Hash}, \mathsf{PHash})$ is

an SPHF system for the language of correct commitments, i.e. $\mathsf{Hash}(\pi, C, m)$ outputs hash value h and projection key pk s.t. (3a) the SPHF is correct in the sense that $\mathsf{PHash}(\pi, pk, m, r) = h$ if $C = \mathsf{Com}((\pi, m); r)$ for some r, and (3b) the SPHF is *covert* in the sense that for any τ, any $\pi \leftarrow \mathsf{PG}(1^\tau)$, any C, m s.t. $C \neq \mathsf{Com}((\pi, m); r)$ for all r, the pair (h, pk) output by $\mathsf{Hash}(\pi, C, m)$ is statistically close to a random bitstring of some length $u(\tau)$. Note that SPHF covertness implies the standard notion of SPHF smoothness, because if m is not committed in C then the hash value h is statistically independent of the projection key pk.

We construct such commitment using ElGamal encryption in a prime residue group: Let $\mathsf{PG}(1^\tau)$ output $\pi = (p, q, k, g, \mathcal{H})$ where p, q are primes s.t. $p = qk + 1$ and $\gcd(q, k) = 1$, g is a generator of subgroup G of order q in Z_p^*, and $\mathcal{H}$ is a universal hash from Z_p^* to $\{0,1\}^\tau$ s.t. for any distribution D over Z_p^* which has at least $|p|$ bits of entropy, $\{\mathcal{H}(x)\}_{x \leftarrow D}$ is statistically close to $\{0,1\}^\tau$. Algorithm $\mathsf{Com}(\pi, m)$ for m in message space $\mathbb{Z}_q$ picks $(t, z_1, z_2) \leftarrow \mathbb{Z}_q \times \mathsf{Z}_p^* \times \mathsf{Z}_p^*$, and outputs $C = (\mathsf{EC}_{[p]}(e), \mathsf{EC}_{[p]}(f))$, where $e = g^t \cdot {z_1}^q \bmod p$ and $f = g^{t^2+m} \cdot {z_2}^q \bmod p$. Under the DDH assumption on subgroup G of Z_p^* this commitment is covert, because then pair (g^t, g^{t^2}) is indistinguishable from two random G elements, which makes pair (e, f) indistinguishable from two random Z_p^* elements. Algorithm $\mathsf{Hash}(\pi, C, m)$ decodes e and f from C, picks $\alpha, \beta \leftarrow \mathbb{Z}_q$ and $z_3 \leftarrow \mathsf{Z}_p^*$, and outputs (h, pk) where $h = \mathcal{H}\big(e^{k^2 \cdot \alpha} \cdot (fg^{-m})^{k^2 \cdot \beta}\big)$ and $pk = \mathsf{EC}_{[p]}\big(g^{k \cdot \alpha} \cdot e^{k \cdot \beta} \cdot {z_3}^q\big)$. Algorithm $\mathsf{PHash}(\pi, pk, m, r)$ for $r = (t, z_1, z_2)$ decodes $v \leftarrow \mathsf{DC}_{[p]}(pk)$ and outputs $h = \mathcal{H}(v^{kt})$. Note that for both parties $h = \mathcal{H}(w)$ for $w = g^{tk^2\alpha + t^2k^2\beta}$ because ${(z_i}^q)^k = {z_i}^{p-1} = 1$. On the other hand, if C is not a commitment to m then (h, pk) output by $\mathsf{Hash}(\pi, C, m)$ are distributed as $(\mathcal{H}(w), \mathsf{EC}_{[p]}(x))$ for $w = g^{k^2 \cdot (t\alpha + t^2\beta + \delta_m \beta)}$ and $x = g^{k \cdot (\alpha + t\beta)} {z_3}^q$ for $\delta_m \neq 0 \bmod q$. Since (α, β, z_3) is random in $\mathbb{Z}_q \times \mathbb{Z}_q \times \mathsf{Z}_p^*$, pair (w, x) is uniform in $G \times \mathsf{Z}_p^*$, and therefore $(\mathcal{H}(w), \mathsf{EC}_{[p]}(x))$ is statistically close to uniform in $\{0,1\}^\tau \times \{0,1\}^{|p|+\tau}$.

Covert CKEM Construction. Let (P_1, P_2, V) be a special Σ-protocol for relation $\mathcal{R}$, with the associated integer ranges $\boldsymbol{I}$, let $(\mathsf{PG}, \mathsf{Com}, \mathsf{Hash}, \mathsf{PHash})$ be a perfectly binding covert commitment with associated SPHF, and let H be a collision-resistant hash onto the message space of the commitment. Fig. 1 shows algorithms S and R for a covert CKEM $(\mathsf{PG}, \mathsf{S}, \mathsf{R})$ for relation $\mathcal{R}$. Note that the security argument for this construction uses rewinding, which degrades exact security. Using the 2-round ROM version of this construction (see below), τ should be at least 160, and H should hash onto at least 480-bit strings. If the covert commitment is implemented using group Z_p^* as shown above, this means that the order q of the subgroup G of Z_p^* must satisfy $|q| \geq 480$.

Theorem 1. *Tuple* $(\mathsf{PG}, \mathsf{S}, \mathsf{R})$ *where* S, R *are specified in Fig. 1 is a* receiver covert and strong sender covert CKEM for relation $\mathcal{R}$ *if* $(\mathsf{PG}, \mathsf{Com}, \mathsf{Hash}, \mathsf{PHash})$ *is a perfectly binding covert commitment with associated SPHF,* (P_1, P_2, V) *is a special* Σ*-protocol for* $\mathcal{R}$*, and* H *is a collision resistant hash function.*

Proof Sketch. To argue receiver covertness note that in the real execution the adversary sees $(C, \overline{z})$ generated as in Fig. 1, and this pair is indistinguishable from

On R's inputs (π, x, w) and S's inputs (π, x) for π generated by $\mathsf{PG}(1^\tau)$.

R: Pick random (r, r'), compute Σ-protocol first message $a \leftarrow P_1(x, w, r)$, compute its hash $C \leftarrow \mathsf{Com}((\pi, H(a)); r')$, and send C to S.

S: Pick challenge $c \leftarrow \{0,1\}^\tau$ and send c to R.

R: Compute Σ-protocol response as $z \leftarrow P_2(x, w, r, c)$, send $\overline{z} \leftarrow \mathsf{EC}_{\boldsymbol{I}}(z)$ to S.

S: Decode $z \leftarrow \mathsf{DC}_{\boldsymbol{I}}(\overline{z})$, use Σ-protocol verification to compute $a \leftarrow f_V(x, c, z)$, compute $(h, pk) \leftarrow \mathsf{Hash}(\pi, C, H(a))$, send pk to R and output key $K = h$.

R: On sender's message pk, output key $K' = \mathsf{PHash}(\pi, pk, H(a), r')$.

Fig. 1. A Covert CKEM for relation $\mathcal{R}$

two random bitstrings of appropriate size: First, by covertness of the commitment scheme, commitment C can be replaced by a random bitstring incurring at most negligible change in the adversary's behavior. Secondly, since z is statistically close to random in $\boldsymbol{I}$ by the properties of the Σ-protocol, and $\mathsf{EC}_{\boldsymbol{I}}$ is covert on $\boldsymbol{I}$, it follows that $\overline{z}$ is statistically close to a random bitstring. For strong sender covertness, take any τ, any π output by $\mathsf{PG}(1^\tau)$, any x polynomial in τ, and an efficient algorithm $\mathcal{A}$. Let $\epsilon_{\mathcal{A},\pi,x,u} = |p_0 - p_1|$ where p_0 is the probability that $\mathcal{A}(\pi, x)$ outputs 1 in an interaction where it gets $(pk, K) = (pk, h)$ computed by $\mathsf{S}(\pi, x)$, and p_1 is the probability that $\mathcal{A}(\pi, x)$ outputs 1 given a random $u(\tau)$-bit string where u is given by the covertness property of the SPHF for the commitment scheme (see property 3b in the definition above). First consider executions where $\mathcal{A}$ sends $(C, \overline{z})$ to S s.t. C is not a commitment to $H(a)$ for $a = f_V(x, c, \mathsf{DC}_{\boldsymbol{I}}(\overline{z}))$. By the covertness of the SPHF for the commitment scheme, in such executions pair (pk, h) is statistically indistinguishable random $u(\tau)$-bit string. Let $\epsilon_{\mathsf{sphf}}(\tau)$ be the upper-bound on the (negligible) amount such executions can contribute to $\mathcal{A}$'s distinguishing advantage $\epsilon_{\mathcal{A},\pi,x,u}$. We conclude that with probability at least $\epsilon' = \epsilon_{\mathcal{A},\pi,x,u} - \epsilon_{\mathsf{sphf}}(\tau)$, a random interaction with $\mathcal{A}(\pi, x)$ outputs $(C, c, \overline{z})$ s.t. C is a commitment to $H(a)$ for $a = f_V(x, c, \mathsf{DC}_{\boldsymbol{I}}(\overline{z}))$. Running such interaction twice with $\mathcal{A}$'s initial randomness fixed until $\mathcal{A}$ outputs C creates a "fork" with two transcripts $(C, c, \overline{z})$ and $(C, c', \overline{z'})$ s.t. with probability at least $\epsilon'' = (\epsilon')^2/2$ (if $\epsilon' \geq 2 \cdot 2^{-\tau}$) we have that $c \neq c'$ and both transcripts are successful in the sense that C is a commitment to $H(a)$ for $a = f_V(x, c, z)$ and C is a commitment to $H(a')$ for $a' = f_V(x, c', z')$, for $(z, z') = \mathsf{DC}_{\boldsymbol{I}}(\overline{z}, \overline{z}')$. $H(a) = H(a')$ by perfect binding of Com. Let $\epsilon_{\mathsf{crh}}(\tau)$ be the (negligible) upper-bound on the probability that this forked execution, running $\mathcal{A}(\pi, x)$ twice, produces a collision in H. Therefore with probability at least $\epsilon'' - \epsilon_{\mathsf{crh}}(\tau)$ we have that $a = a'$, in which case the extractor implied by the special soundness of the Σ-protocol outputs w s.t. $(x, w) \in \mathcal{R}$ when executed on input (x, a, c, z, c', z'), which implies strong sender covertness for $\delta(\tau) = 2^{-\tau+2} + 2\epsilon_{\mathsf{sphf}}(\tau) + 4\sqrt{\epsilon_{\mathsf{crh}}(\tau)}$ and $p(\epsilon) = \epsilon^2/16$.

2-round Covert CKEM in ROM. The same construction becomes a 2-round CKEM in the Random Oracle Model (ROM), if c is computed as $c = H'(x, C)$

for a hash function H' onto $\{0,1\}^\tau$ modeled as a random oracle. If $\mathcal{A}(\pi, x)$ can make at most $q_H(\tau)$ hash queries then using the version of the forking lemma in [BN06] we get a (forking) algorithm which on input (π, x) runs two executions of $\mathcal{A}(\pi, x)$ and creates the same two transcripts as above with probability $\epsilon'' = (\epsilon')^2/(2q_H(\tau))$ given $\epsilon' \geq 2 \cdot q_H(\tau)/2^\tau$, which implies sender covertness for $\delta(\tau) = q_H(\tau) \cdot 2^{-\tau+2} + 2\epsilon_{\mathsf{sphf}}(\tau) + 4\sqrt{q_H(\tau)\epsilon_{\mathsf{crh}}(\tau)}$ and $p(\epsilon) = \epsilon^2/(16q_H(\tau))$.

5 Covert Mutual Authentication Scheme

We construct a covert Mutual Authentication (MA) from an Identity Escrow (IE) scheme [KP98] where a group member commits to its certificate and then proves in zero-knowledge that the committed value is a valid certificate under the group public key. We turn such IE scheme into a covert MA scheme by replacing the zero-knowledge proof with a covert CKEM for the same relation. For revocation we require that each commitment can be linked to a committed certificate given the revocation token corresponding to this certificate, and to assure covertness we need this certificate commitment to be covert until the revocation token is made public. Identity Escrow [KP98] is an interactive form of a group signature [CvH91], and many group signatures can be converted to an IE scheme which fits the above structure. Below we formalize the properties our MA scheme construction requires of an IE scheme, and we show how to build a covert MA protocol from such IE scheme and a covert CKEM for committed certificate validity. In Section 5.1 we show how to instantiate this construction by modifying the Ateniese-Camenisch-Joye-Tsudik (ACJT) group signature [ACJT00] into an IE scheme that satisfies the properties required by this construction.

Compatible Identity Escrow Scheme. An IE scheme is a tuple of algorithms $(\mathsf{KG}, \mathsf{CG}, \mathsf{Ver}, \mathsf{IECom}, \mathsf{TraceCom})$, where $\mathsf{KG}(1^\tau)$ outputs a group secret key gsk and a public key gpk, $\mathsf{CG}(\mathsf{gsk})$ generates a certificate $(\mathsf{sk}, \mathsf{rt})$, where sk is a user secret and rt a revocation token, s.t. $\mathsf{Ver}(\mathsf{gpk}, (\mathsf{sk}, \mathsf{rt})) = 1$, $\mathsf{IECom}(\mathsf{gpk}, (\mathsf{sk}, \mathsf{rt}))$ generates a commitment C to $(\mathsf{sk}, \mathsf{rt})$, and $\mathsf{TraceCom}(\mathsf{gpk}, C, \mathsf{rt}) = 1$ if $C \leftarrow \mathsf{IECom}(\mathsf{gpk}, (\mathsf{sk}, \mathsf{rt}))$. We call an IE scheme *covert-MA-compatible* if it satisfies the following four properties. (1) First, $(\mathsf{KG}, \mathsf{Ver})$ must form an *unforgeable certificate scheme*, i.e. for any efficient algorithm $\mathcal{A}$, the probability that $\mathcal{A}(\mathsf{gpk})$, on access to an oracle $\mathsf{CG}(\mathsf{gsk})$, generates $(\mathsf{sk}^*, \mathsf{rt}^*)$ s.t. $\mathsf{Ver}(\mathsf{gpk}, (\mathsf{sk}^*, \mathsf{rt}^*)) = 1$ and $\mathsf{rt}^* \neq \mathsf{rt}_i$ for all $(\mathsf{sk}_i, \mathsf{rt}_i)$ pairs $\mathcal{A}$ receives from $\mathsf{CG}(\mathsf{gsk})$, is negligible, for $(\mathsf{gsk}, \mathsf{gpk})$ randomly generated by $\mathsf{KG}(1^\tau)$. (2) Second, the scheme must be *traceable*, i.e. for any τ, any $(\mathsf{gsk}, \mathsf{gpk})$ output by $\mathsf{KG}(1^\tau)$, and any C and rt, it holds that $\mathsf{TraceCom}(\mathsf{gpk}, C, \mathsf{rt}) = 1$ if and only if $C = \mathsf{IECom}((\mathsf{gpk}, (\mathsf{sk}, \mathsf{rt})); r)$ for some sk, r. (3) We define a *committed certificate validity* relation $\mathcal{R}^{IE}$ as the set $((\mathsf{gpk}, C), (\mathsf{sk}, \mathsf{rt}, r))$ s.t. $C = \mathsf{IECom}((\mathsf{gpk}, (\mathsf{sk}, \mathsf{rt})); r)$ and $\mathsf{Ver}(\mathsf{gpk}, (\mathsf{sk}, \mathsf{rt})) = 1$. The third property of an IE scheme is that $\mathcal{R}^{IE}$ admits a *special Σ-protocol*, so that it can be converted into a covert CKEM by the construction in Fig. 1.

(4) The last property is the covertness of the commitment IECom. Note that traceability implies that IECom cannot be semantically secure because the rt part of the committed plaintext can be efficiently linked to the commitment.

However, the commitment must hide the committed certificate $(\mathsf{sk}, \mathsf{rt})$ as long as the revocation token rt is not made public, and we need this commitment to be covert and not just plaintext-hiding. Thus, we require the IE scheme to be *revocably covert* in the sense that there exists some function l polynomial in τ s.t. for any efficient algorithm $\mathcal{A}$, quantity $|p_0 - p_1|$ is a negligible function of τ, where p_β is defined as the probability that $b = 1$ in the following experiment: Generate $(\mathsf{gsk}, \mathsf{gpk}) \leftarrow \mathsf{KG}(1^\tau)$ and $(\mathsf{sk}_t, \mathsf{rt}_t) \leftarrow \mathsf{CG}(\mathsf{gsk})$, and then let $\mathcal{A}(\mathsf{gpk})$ repeatedly query the $\mathsf{CG}(\mathsf{gsk})$ oracle which generates $(\mathsf{sk}, \mathsf{rt})$ and gives it to $\mathcal{A}$, and an oracle which returns $C \leftarrow \mathsf{IECom}(\mathsf{gpk}, (\mathsf{sk}_t, \mathsf{rt}_t))$ for $\beta = 1$, or $C \leftarrow \{0,1\}^{l(\tau)}$ for $\beta = 0$. $\mathcal{A}$ outputs bit b, its guess of bit β, after polynomially many queries of both types.

Covert MA Scheme Construction. Fig. 2 constructs a covert MA scheme given a covert-MA-compatible IE scheme $(\mathsf{KG}, \mathsf{CG}, \mathsf{Ver}, \mathsf{IECom}, \mathsf{TraceCom})$ and a receiver covert and strong sender covert CKEM $(\mathsf{PG}, \mathsf{S}, \mathsf{R})$ for the associated committed certificate validity relation $\mathcal{R}^{IE}$. In the figure, u_S stands for the polynomial sequence implied by CKEM strong sender covertness.

$\underline{\mathsf{KGen}}(1^\tau)$: Set $(\mathsf{gsk}, \mathsf{gpk}) \leftarrow \mathsf{KG}(1^\tau)$, $\pi \leftarrow \mathsf{PG}(1^\tau)$, $\mathsf{mpk} = (\mathsf{gpk}, \pi)$, and $\mathsf{msk} = \mathsf{gsk}$.

$\underline{\mathsf{CG}}(\mathsf{gsk})$: Generate $(\mathsf{sk}, \mathsf{rt})$ following the $\mathsf{CG}(\mathsf{gsk})$ algorithm of the IE scheme.

$\underline{\mathsf{Auth}}$ protocol for $P_i((\mathsf{gpk}, \pi), (\mathsf{sk}_i, \mathsf{rt}_i), \mathsf{CRL}_i)$ and $P_j((\mathsf{gpk}, \pi), (\mathsf{sk}_j, \mathsf{rt}_j), \mathsf{CRL}_j)$:

1. P_i sets $C_i \leftarrow \mathsf{IECom}((\mathsf{gpk}, (\mathsf{sk}_i, \mathsf{rt}_i)); r_i)$ for random r_i and sends C_i to P_j.
P_j sets $C_j \leftarrow \mathsf{IECom}((\mathsf{gpk}, (\mathsf{sk}_j, \mathsf{rt}_j)); r_j)$ for random r_j and sends C_j to P_i.

P_i sets $\mathsf{F}_i \leftarrow 1$ if $\mathsf{TraceCom}(\mathsf{gpk}, C_j, \mathsf{rt}) = 1$ for any $\mathsf{rt} \in \mathsf{CRL}_i \cup \{\mathsf{rt}_i\}$, and $\mathsf{F}_i \leftarrow 0$ otherwise.
P_j sets $\mathsf{F}_j \leftarrow 1$ if $\mathsf{TraceCom}(\mathsf{gpk}, C_i, \mathsf{rt}) = 1$ for any $\mathsf{rt} \in \mathsf{CRL}_j \cup \{\mathsf{rt}_j\}$, and $\mathsf{F}_j \leftarrow 0$ otherwise.

2. P_i runs protocol R on $(\pi, (\mathsf{gpk}, C_i), (\mathsf{sk}_i, \mathsf{rt}_i, r_i))$, interacting with P_j who runs protocol S on $(\pi, (\mathsf{gpk}, C_i))$ if $\mathsf{F}_j = 0$, or runs $\mathsf{S}^{\$(u_\mathsf{S})}$ if $\mathsf{F}_j = 1$.
P_i sets $K_{i,\mathsf{R}}$ as its local output in R.
P_j sets $K_{j,\mathsf{S}}$ as its local output in S if $\mathsf{F}_j = 0$, otherwise $K_{j,\mathsf{S}} \leftarrow \{0,1\}^\tau$.

3. P_j runs protocol R on $(\pi, (\mathsf{gpk}, C_j), (\mathsf{sk}_j, \mathsf{rt}_j, r_j))$, interacting with P_i who runs protocol S on $(\pi, (\mathsf{gpk}, C_j))$ if $\mathsf{F}_i = 0$, or runs $\mathsf{S}^{\$(u_\mathsf{S})}$ if $\mathsf{F}_i = 1$.
P_j sets $K_{j,\mathsf{R}}$ as its local output in R.
P_i sets $K_{i,\mathsf{S}}$ as its local output in S if $\mathsf{F}_i = 0$, otherwise $K_{i,\mathsf{S}} \leftarrow \{0,1\}^\tau$.

P_i's local output is $K_i = K_{i,\mathsf{R}} \oplus K_{i,\mathsf{S}}$ and P_j's local output is $K_j = K_{j,\mathsf{R}} \oplus K_{j,\mathsf{S}}$.

Fig. 2. A Covert Mutual Authentication Scheme $(\mathsf{KGen}, \mathsf{CG}, \mathsf{Auth})$.

Theorem 2. $(\mathsf{KGen}, \mathsf{CG}, \mathsf{Auth})$ *in Fig. 2 is a* Covert Mutual Authentication Scheme *if* $(\mathsf{KG}, \mathsf{CG}, \mathsf{Ver}, \mathsf{IECom}, \mathsf{TraceCom})$ *is a covert-MA-compatible IE scheme and* $(\mathsf{PG}, \mathsf{S}, \mathsf{R})$ *is a receiver covert and strong sender covert CKEM for* $\mathcal{R}^{IE}$.

Proof Sketch. By the symmetry of the Auth protocol we can assume that in all the Auth protocol instances adversary invokes its counterparty plays the role of P_i in Fig. 2. Let $l(\cdot)$ be the length polynomial implied by revocable covertness of the IE scheme, and let u_{R} and u_{S} be the polynomial sequences implied by the receiver and sender covertness of the CKEM. The polynomial sequence u which defines the random beacon $\mathsf{Auth}^{\$(u)}$ is composed of $l(\cdot)$ followed by the elements of u_{R} and then the elements of u_{S}, because P_i first sends C_i, then performs R, and then S (or $\mathsf{S}^{\$(u)}$). Let $\mathcal{A}$ be an efficient algorithm with the distinguishing advantage $\epsilon_{\mathcal{A}}$ in the MA covertness experiment (see Definition 3). For any $i \in \{0, \ldots, N(\tau)\}$, consider a game $G(1^\tau, b, i^*)$ which follows $G(1^\tau, b)$ but fixes the index i used by $\mathcal{A}$ in the Test query by halting and outputting 1 if $\mathcal{A}$ calls the $\mathsf{Test}(i)$ query for $i \neq i^*$. There must exist an index i^* s.t. $\mathcal{A}$'s advantage in distinguishing between $G_1 = G(1^\tau, 1, i^*)$ and $G_0 = G^(1^\tau, 0, i^*)$ is at least $\epsilon_{\mathcal{A}}/N(\tau)$. By a series of modifications starting from game G_1 we show that $\mathcal{A}$'s distinguishing advantage between G_1 and G_0 must be negligible, implying that $\epsilon_{\mathcal{A}}$ is negligible. In the following we will only consider $\mathsf{Exec}(i, \mathsf{CRL}^*)$ queries for i s.t. $\mathsf{rt}_i \notin \mathsf{CRL}$, because $\mathcal{A}$ can execute the game response on such queries for $i \in \mathsf{CRL}$ using the $(\mathsf{sk}_i, \mathsf{rt}_i)$ certificate $\mathcal{A}$ received by corrupting P_i.

A hybrid argument shows that G_1 is indistinguishable from G_2 where all Auth instances followed by P_i on $\mathsf{Exec}(i, \mathsf{CRL}^*)$ queries are modified by replacing $\mathsf{R}(\pi, (\mathsf{gpk}, C_i), (\mathsf{sk}_i, \mathsf{rt}_i, r_i))$ with $\mathsf{R}^{\$(u_{\mathsf{R}})}$ in step (2) of Auth. Let $G_1(t)$ be a hybrid between G_1 and G_2 which responds to the first t of Exec queries as in G_2, and to the remaining ones as in G_1. $\mathcal{A}$'s advantage in distinguishing $G_1(t-1)$ and $G_1(t)$ must be negligible for each t by CKEM receiver covertness. A reduction which shows it runs on input π, generates $(\mathsf{gsk}, \mathsf{gpk})$, interacts with either $\mathsf{R}(\pi, (\mathsf{gpk}, C_i), (\mathsf{sk}_i, \mathsf{rt}_i, r_i))$ or $\mathsf{R}^{\$(u_{\mathsf{R}})}$ on $\mathcal{A}$'s t-th query $\mathsf{Exec}(i, \mathsf{CRL}^*)$, and simulates the rest of $\mathcal{A}$'s view in either game.

Let $\mathsf{CorSet}^+ = \mathsf{CorSet} \cup \{i^*\}$ and $\mathsf{CRL}^+ = \mathsf{CRL} \cup \{\mathsf{rt}_{i^*}\}$. By another hybrid we modify G_2 into G_3 by replacing the C_i values generated in the Auth instances by each P_i for $i \notin \mathsf{CorSet}^+$, with random strings of length $l(\tau)$. This hybrid goes over the players rather than over the Exec sessions. Let $G_2(t)$ be a game which follows G_2 in servicing each $\mathsf{Exec}(i, \mathsf{CRL}^*)$ query for $i > t$, but on queries $\mathsf{Exec}(i, \mathsf{CRL}^*)$ for $i \leq t$ and $i \notin \mathsf{CorSet}^+$ it replaces C_i generated as $C_i \leftarrow \mathsf{IECom}(\mathsf{gpk}, (\mathsf{sk}_i, \mathsf{rt}_i))$ with a random $l(\tau)$-bit string. Note that the subsequent steps of P_t in the Auth instances triggered by Exec queries in G_2 do not depend on either C_t or $(\mathsf{sk}_t, \mathsf{rt}_t, r_t)$, which allows us to reduce $\mathcal{A}$'s advantage in distinguishing $G_2(t-1)$ and $G_2(t)$ to an attack on the revocable covertness of the IE scheme: The challenger generates $(\mathsf{gsk}, \mathsf{gpk}) \leftarrow \mathsf{KG}(1^\tau)$ and $(\mathsf{sk}_t, \mathsf{rt}_t) \leftarrow \mathsf{CG}(\mathsf{gsk})$, the reduction on input gpk receives certificates $(\mathsf{sk}_i, \mathsf{rt}_i)$ for all $i \neq t$ from the $\mathsf{CG}(\mathsf{gsk})$ oracle, receives either a sequence of C_t's computed as $C_t \leftarrow \mathsf{IECom}(\mathsf{gpk}, (\mathsf{sk}_t, \mathsf{rt}_t))$ or as a sequence of random bitstrings, and simulates everything else $\mathcal{A}$ sees in either game.

Note that G_3 responds to each $\mathsf{Exec}(i, \mathsf{CRL}^*)$ query for $i \notin \mathsf{CorSet}^+$ by picking C_i as a random string in step (1), running $\mathsf{R}^{\$(u_{\mathsf{R}})}$ in step (2), and running $\mathsf{S}(\pi, (\mathsf{gpk}, C_j))$ for C_j supplied by $\mathcal{A}$ in step (3). Therefore G_3 can be simulated given π, gpk, and the certificates $(\mathsf{rt}_i, \mathsf{sk}_i)$ for $i \in \mathsf{CorSet}^+$. Let G_4 be G_3

with P_{i^*}'s code in the Auth instance triggered by the $\mathsf{Test}(i^*)$ query modified by replacing the $\mathsf{S}(\pi, (\mathsf{gpk}, C_j))$ protocol P_{i^*} follows if $\mathsf{F}_{i^*} = 0$ with a random beacon $\mathsf{S}^{\$(u_\mathsf{S})}$ and a random key $K_{i^*,\mathsf{S}}$. If we assume that $\mathcal{A}$'s advantage in distinguishing between G_3 and G_4 is non-negligible, then by the strong sender covertness of CKEM it follows that there is an efficient extractor which, on input $(\mathsf{gpk}, \pi, \{\mathsf{sk}_i, \mathsf{rt}_i\}_{i \in \mathsf{CorSet}^+})$, extracts with non-negligible probability a witness $(\mathsf{sk}, \mathsf{rt}, r)$ s.t. $((\mathsf{gpk}, C_j), (\mathsf{sk}, \mathsf{rt}, r)) \in \mathcal{R}^{IE}$, i.e. $C_j = \mathsf{IECom}((\mathsf{gpk}, (\mathsf{sk}, \mathsf{rt})); r)$ and $\mathsf{Ver}(\mathsf{gpk}, (\mathsf{sk}, \mathsf{rt})) = 1$. Since the difference in this modification appears only for $\mathsf{F}_{i^*} = 0$ (otherwise P_{i^*} executes $\mathsf{S}^{\$(u_\mathsf{S})}$ in either case), we can consider only sessions where $\mathsf{TraceCom}(\mathsf{gpk}, C_j, \mathsf{rt}_i) = 0$ for all $\mathsf{rt}_i \in \mathsf{CRL}^+$. By the traceability property this implies that the extracted witness $(\mathsf{sk}, \mathsf{rt}, r)$ must satisfy $\mathsf{rt} \notin \mathsf{CRL}^+$. Therefore a reduction which simulates $\mathcal{A}$'s view on input gpk, and on $(\mathsf{sk}_i, \mathsf{rt}_i)$ pairs for $i \in \mathsf{CorSet}^+$, can with non-negligible probability compute $(\mathsf{sk}, \mathsf{rt})$ s.t. $\mathsf{Ver}(\mathsf{gpk}, (\mathsf{sk}, \mathsf{rt})) = 1$ and $\mathsf{rt} \neq \mathsf{rt}_i$ for all $i \in \mathsf{CorSet}^+$, which breaks the unforgeability of the $(\mathsf{KG}, \mathsf{Ver})$ certificate scheme.

Note that in G_4 key $K_{i^*,\mathsf{S}}$, computed in the $\mathsf{Test}(i^*)$ query, masks key $K_{i^*,\mathsf{R}}$, so now the latter key becomes irrelevant to $\mathcal{A}$'s view and K_{i^*} can be picked at random. This allows us to modify G_4 into G_5, by replacing $\mathsf{R}(\pi, (\mathsf{gpk}, C_{i^*}), (\mathsf{sk}_{i^*}, \mathsf{rt}_{i^*}, r_{i^*}))$ in the Auth instance triggered by the $\mathsf{Test}(i^*)$ query with $\mathsf{R}^{\$(u_\mathsf{R})}$. By CKEM receiver covertness we get that $G_4 \approx G_5$, via a reduction similar to the one which shows that $G_1(t-1) \approx G_1(t)$. We then modify G_5 into G_6, by replacing C_{i^*} in all Auth instances (in both $\mathsf{Test}(i^*)$ and $\mathsf{Exec}(i^*, \mathsf{CRL}^*)$) with a random $l(\tau)$-bit string. By revocable covertness of the IE scheme we get that $G_5 \approx G_6$, via a reduction similar to the one which shows that $G_2(t-1) \approx G_2(t)$. Note that in G_6 player P_{i^*} responds to the $\mathsf{Test}(i^*)$ query as $\mathsf{Auth}^{\$(u)}$ and outputs a random τ-bit string as key K_{i^*}, but also each P_i for $i \notin \mathsf{CorSet}$ responds to every $\mathsf{Exec}(i, \mathsf{CRL}^*)$ query by sending a random string instead of C_i in step (1) and following $\mathsf{R}^{\$(u_\mathsf{R})}$ instead of R in step (2). However, we can roll back those changes in responses to $\mathsf{Exec}(i, \mathsf{CRL}^*)$ queries. Using a similar argument as above for arguing indistinguishability of G_2 and G_3, we first change P_i's responses in $\mathsf{Exec}(i, \mathsf{CRL}^*)$ queries by replacing random C_i's back with $C_i \leftarrow \mathsf{IECom}(\mathsf{gpk}, (\mathsf{sk}_i, \mathsf{rt}_i))$. Then, using a similar argument as above for arguing indistinguishability of G_1 and G_2 we change P_i's responses to $\mathsf{Exec}(i, \mathsf{CRL}^*)$ queries by replacing $\mathsf{R}^{\$(u_\mathsf{R})}$ back with $\mathsf{R}(\pi, (\mathsf{gpk}, C_i), (\mathsf{sk}_i, \mathsf{rt}_i, r_i))$. After these modifications the game is identical to G_0, which completes the proof.

5.1 Covert MA Instantiation from ACJT Group Signature

RSA Setting. We first introduce the cryptographic setting required by the ACJT group signature scheme and by the covert encodings we will apply to it. The *safe* RSA setting modulus of length $l_n = 2l + 2$, for l polynomial in security parameter τ, is a product $n = pq$ of two primes p, q s.t. $p = 2p' + 1$ and $q = 2q' + 1$ where p', q' are also primes and $|p'| = |q'| = l$. The subgroup of quadratic residues in Z_n^*, denoted QR_n, is a cyclic group of order $n' = p'q'$. Let g be a generator of QR_n. Note that $-1 \notin \mathsf{QR}_n$ but $J_n(-1)$, the Jacobi symbol of -1 mod n, is equal to 1. We use $\pm\mathsf{QR}_n$ to denote the set of elements whose

Jacobi symbol is 1. ($\pm\mathsf{QR}_n$ contains x and $-x$ for $x \in \mathsf{QR}_n$.) We use the following assumptions on safe RSA moduli, where negl stands for a negligible function:

Definition 4 (Strong RSA Assumption). *For all efficient algorithms $\mathcal{A}$ there is a negligible function negl s.t. if n is a random safe RSA modulus of length l_n, and z is a random element in Z_n^*, the probability that $\mathcal{A}(n,z)$ outputs (x,e) s.t. $e \neq 1$ and $x^e = z \bmod n$, is upper-bounded by $\mathsf{negl}(l_n)$. (Note that since QR_n makes $1/4$-th of Z_n^*, same assumption holds if z is sampled from QR_n.)*

Definition 5 (Decisional Quadratic Residuosity (DQR) Assumption). *For all efficient algorithms $\mathcal{A}$ there is a negligible function negl s.t. if n is a random safe RSA modulus of length l_n, the distinguishability advantage $|\epsilon_0 - \epsilon_1|$, where $\epsilon_0 = \Pr[1 \leftarrow \mathcal{A}(n,a)]$ for $a \in \mathsf{QR}_n$ and $\epsilon_1 = \Pr[1 \leftarrow \mathcal{A}(n,a)]$ for $a \pm \mathsf{QR}_n$, is upper-bounded by $\mathsf{negl}(l_n)$.*

Definition 6 (Decisional Diffie-Hellman (DDH) Assumption on QR_n). *For all efficient algorithms $\mathcal{A}$ there is a negligible function negl s.t. if n is a random safe RSA modulus of length l_n, and $\hat{g}$ is a random generator of QR_n, the distinguishability advantage $|\epsilon_0 - \epsilon_1|$, where $\epsilon_0 = \Pr[1 \leftarrow \mathcal{A}(\hat{g}, \hat{g}^a, \hat{g}^b, \hat{g}^c)]$ for $a,b,c \leftarrow \mathsf{Z}_{n'}$ and $\epsilon_1 = \Pr[1 \leftarrow \mathcal{A}(n, \hat{g}, \hat{g}^a, \hat{g}^b, \hat{g}^{ab})]$ for $a,b \leftarrow \mathsf{Z}_{n'}$, is upper-bounded by $\mathsf{negl}(l_n)$.*

Covert Encoding for QR_n. The ACJT group signature works in the QR_n subgroup of Z_n^*, but a protocol whose messages are elements of QR_n would not be covert because one can distinguish QR_n from Z_n^* by computing a Jacobi symbol mod n. We can handle it using the DQR assumption as follows. Let ν be any element in Z_n^* of order $2n'$ s.t. $J_n(\nu) = -1$. Let $\mathsf{EC}_{\pm\mathsf{QR}_n}$ be an encoding of $\pm\mathsf{QR}_n$ where $\mathsf{EC}_{\pm\mathsf{QR}_n}(v)$ picks a random bit β and returns $\mathsf{EC}_{[n]}(\nu^\beta \cdot v)$. The decoding $\mathsf{DC}_{\pm\mathsf{QR}_n}(\overline{v})$ computes $v' \leftarrow \mathsf{DC}_{[n]}(\overline{v})$ and outputs $v = v'$ if $J(v',n) = 1$ and $v = v'/\nu \bmod n$ if $J(v',n) = -1$. $\mathsf{EC}_{\pm\mathsf{QR}_n}$ is covert for message space $\pm\mathsf{QR}_n$ because $\pm\mathsf{QR}_n \times \{1,\nu\}$ is isomorphic to Z_n^* and Z_n^* is statistically indistinguishable from $[n]$. Since under the DQR assumption QR_n is indistinguishable from $\pm\mathsf{QR}_n$, the same encoding is also covert for message space QR_n, assuming DQR.

Covert-MA-Compatible IE Scheme from ACJT Group Signature. We explain how the ACJT group signature [ACJT00] can be transformed into a *covert-MA-compatible* IE scheme $(\mathsf{KG}, \mathsf{CG}, \mathsf{Ver}, \mathsf{IECom}, \mathsf{TraceCom})$ which we will call a *ACJT-IE*. This provides an instantiation of the covert MA construction of Fig. 2 because by the property (4) of a covert-MA-compatible IE scheme, we can construct a receiver covert and strong sender covert CKEM for the $\mathcal{R}^{IE}$ relation associated with this IE scheme using the CKEM construction in Fig. 1, and then we can use this CKEM together with the rest of the IE scheme in the covert MA construction in Fig. 2. By combining the assumptions required for the ACJT-IE scheme and for the CKEM construction (as stated in Theorem 1), we get the following corollary of Theorem 2:

Corollary 1. *The* $(\mathsf{KGen}, \mathsf{CG}, \mathsf{Auth})$ *in Fig. 2 instantiated with the ACJT-IE scheme and the CKEM scheme of Fig.1, is a* Covert Mutual Authentication

Scheme, *assuming the strong RSA and DQR assumptions on* Z_n^* *for the safe RSA modulus* n*, the DDH assumption on the* QR_n *subgroup of* Z_n^**, and the DDH assumption on a prime-order subgroup of a prime residue group.*

We show the ACJT-IE scheme $(\mathsf{KG}, \mathsf{CG}, \mathsf{Ver}, \mathsf{IECom}, \mathsf{TraceCom})$ and explain how it relies on the strong RSA, DDH, and DQR assumptions stated above. Algorithm KG sets the group public key as $\mathsf{gpk} = (n, a, a_0, y, g, h)$, as in the original ACJT group signature [ACJT00], where n is a safe RSA modulus and a, a_0, y, g, h are all random generators of QR_n. The group secret key gsk is the factorization of n. CG outputs $(\mathsf{sk}_i, \mathsf{rt}_i) = ((A_i, e_i), x_i)$ where $x_i \leftarrow 2^{\lambda_1} \pm [2^{\lambda_2}]$, e_i is a random prime in $2^{\gamma_1} \pm [2^{\gamma_2}]$, and $A_i = (a^{x_i} a_0)^{1/e_i} \bmod n$, for parameters $\lambda_1, \lambda_2, \gamma_1, \gamma_2$ set as $\lambda_2 \approx 2l_n = 2|n|$, $\lambda_1 \approx \lambda_2 + \tau$, $\gamma_2 \approx \lambda_1 + 2$, and $\gamma_1 \approx \gamma_2 + \tau$. Algorithm $\mathsf{Ver}(\mathsf{gpk}, (A_i, e_i), x_i)$ returns 1 if $A_i^{e_i} = a^{x_i} a_0 \bmod n$ and 0 otherwise. Commitment IECom on inputs $(\mathsf{gpk}, ((A_i, e_i), x_i))$ picks $w \leftarrow [n/4]$ and computes $T_1 \leftarrow A_i y^w$, $T_2 \leftarrow g^w$, and $T_3 \leftarrow g^{e_i} h^w$, just like in the ACJT scheme, but in addition it picks a random QR_n element T_4, computes $T_5 \leftarrow (T_4)^{x_i}$, and outputs $C = (\overline{T}_1, \ldots, \overline{T}_5)$ where $\overline{T}_i \leftarrow \mathsf{EC}_{\pm\mathsf{QR}_n}(T_i)$ for each i. $\mathsf{TraceCom}(\mathsf{gpk}, C, x_i)$ outputs 1 iff $T_5 = (T_4)^{x_i}$ for T_4, T_5 decoded from $\overline{T}_4, \overline{T}_5$ in C.

Unforgeability of the $(\mathsf{KG}, \mathsf{Ver})$ certificate scheme is argued in [ACJT00] under the strong RSA assumption on QR_n. Traceability follows by the fact that procedure $\mathsf{TraceCom}(\mathsf{gpk}, C, x_i)$ computes T_5 from T_4 in the same way as IECom on x_i. As for revocable covertness, since $\lambda_2 \geq 2|n|$ we have that for x_i uniform in $2^{\lambda_1} \pm [2^{\lambda_2}]$ value $(x_i \bmod n')$ is statistically indistinguishable from uniform over $\mathsf{Z}_{n'}$. Therefore, for secret x_i, under DDH assumption on QR_n the 5-tuple $(T_1, \ldots, T_5)$ is indistinguishable from uniform over $(\mathsf{QR}_n)^5$, and therefore by covertness of $\mathsf{EC}_{\pm\mathsf{QR}_n}$ commitment $\overline{T}$ is indistinguishable from a random bit-string. Finally, the HVZK proof system given for the ACJT group signature in [ACJT00], amended by the simple consistency check for the new (T_4, T_5) values, is a special Σ-protocol for the associated relation $\mathcal{R}^{IE}$. We include this amended proof system of [ACJT00] in Appendix A.

Efficiency of the Resulting Covert MA Scheme. The Covert MA protocol of Fig. 2 can be condensed to three rounds in ROM: Player P_i can piggyback R's message in the CKEM instance of step 2 with the commitment C_i it sends in step 1. Then player P_j can piggyback its commitment C_j with S's response in the CKEM instance of Step 2 and with R's message in the CKEM instance of step 3. Finally P_i would respond with S's response in the CKEM instance of step 3. As for the computational cost of this scheme instantiated with ACJT-IE scheme, note that ACJT-IE uses 4 multi-exp's in the certificate commitment IECom and that the Σ-protocol for the associated relation $\mathcal{R}^{IE}$ uses 5 multi-exp's for each party. Since each party plays the prover in one direction and the verifier in the other, the total comes to 14 (multi-)exp's in Z_n^*. The CKEM protocol in Fig. 1 adds 5 exp's in Z_p^* for each party (2 as the sender and 3 as the receiver). Moduli p and n can both be 2048 bits long, but exp's in Z_p^* are with much smaller exponents. Looking closer at the 14 multi-exp's in Z_n^* in the computation of T_i's, and d_i's in either step 1 for the prover or step 4 for the verifier (see the Σ-protocol in Appendix A), for $|n| = 2048$ and $\tau = 160$ this

makes four 2048-bit exp's (i.e. T_1, T_2, and d_3 for both parties) and ten exp's with exponents between 4000 and 5000 bits. By comparison, the five exp's in Z_p^* have only 480-bit exponents. The total cost for each party, of these 14 exp's in Z_n^* and 5 exp's in Z_p^*, can be approximated as 30 full exp's in Z_n^* for $|n| = 2048$. However, each party additionally performs $|\mathsf{CRL}| + 1$ exp's in Z_n^* in the $\mathsf{TraceCom}$ checks for each rt in CRL and for one's own rt. Since exponents x_i are roughly twice longer than $|n|$, the total cost is approximately $32 + 2|\mathsf{CRL}|$ full exp's in Z_n^* with $|n| = 2048$ and $\tau = 160$. The bandwidth is about 29Kb in each direction. Note that these costs are almost exactly as in the underlying ACJT group signature scheme, so the practicality of our ACJT-based covert MA scheme depends on whether the two parties have access to a random steganographic channel with enough capacity to transmit 29Kb.

References

[ACJT00] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO*, 2000.

[BCK98] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols. In *STOC '98*, pages 419–428, 1998.

[BN06] M. Bellare and G. Neven. Multisignatures in the plain publickey model and a general forking lemma. In *Proceedings of ACM CCS*, 2006.

[BS04] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In *ACM Conference on Computer and Communications Security*, pages 168–177, 2004.

[CGOS07] Nishanth Chandran, Vipul Goyal, Rafail Ostrovsky, and Amit Sahai. Covert multi-party computation. In *FOCS*, pages 238–248, 2007.

[CK01] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In *EUROCRYPT*, pages 453–474, 2001.

[CM99] Jan Camenisch and Markus Michels. Proving in zero-knowledge that a number is the product of two safe primes. In *EUROCRYPT*, 1999.

[COR99] Giovanni Di Crescenzo, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. Conditional oblivious transfer and timed-release encryption. In *EUROCRYPT*, pages 74–89, 1999.

[Cre00] Giovanni Di Crescenzo. Private selective payment protocols. In *Financial Cryptography*, pages 72–89, 2000.

[CS01] Ronald Cramer and Victor Shoup. Universal hash proofs and and a paradigm for adaptive chosen ciphertext secure public-key encryption. *Electronic Colloquium on Computational Complexity (ECCC)*, 8(072), 2001.

[CvH91] D. Chaum and E. van Heyst. Group signatures. In *Advances in Cryptology – EUROCRYPT '91*, pages 257 –265, 1991.

[Dam10] Ivan Damgard. On Σ-protocols, 2010. *url:www.cs.au.dk/ ivan/Sigma.pdf.*

[GJ10] Vipul Goyal and Abhishek Jain. On the round complexity of covert computation. In *STOC*, 2010.

[HLvA02] Nicholas J. Hopper, John Langford, and Luis von Ahn. Provably secure steganography. In *CRYPTO*, pages 77–92, 2002.

[JL09] Stanislaw Jarecki and Xiaomin Liu. Private mutual authentication and conditional oblivious transfer. In *CRYPTO*, pages 90–107, 2009.

[KP98] J. Kilian and E. Petrank. Identity escrow. In *Advances in Cryptography - CRYPT0 1998*, Santa Barbara, CA, August 1998.

[vAH04] Luis von Ahn and Nicholas J. Hopper. Public-key steganography. In *EUROCRYPT*, pages 323–341, 2004.

[vAHL05] Luis von Ahn, Nicholas J. Hopper, and John Langford. Covert two-party computation. In *STOC*, pages 513–522, 2005.

A Special Σ-Protocol for the ACJT-IE Scheme

We show a proof system for the committed certificate validity relation $\mathcal{R}^{IE}$ in the ACJT-IE scheme of Section 5.1, which satisfies the properties of a *special Σ-protocol*, and hence it can be compiled into a covert CKEM for the same relation using our CKEM construction in Fig. 1. The proof system below is a simple modification of the proof system for the ACJT group signature [ACJT00] extended by a check that $T_5 = T_4^{x_i}$. Relation $\mathcal{R}^{IE}$ for the ACJT-IE scheme consists of pairs $(\hat{x}, \hat{w}) = (((n, a, a_0, y, g, h), (\overline{T}_1, \ldots, \overline{T}_5)), ((A_i, e_i), x_i, w))$ which satisfy the following set of relations for T_i's decoded from $\overline{T}_i$'s using $\mathsf{DC}_{\pm\mathsf{QR}_n}$:

$$T_1 = A_i y^w \ , \ T_2 = g^w \ , \ T_3 = g^{e_i} h^w \ , \ T_5 = T_4^{x_i} \ , \ A_i^{e_i} = a^{x_i} a_0 \ , \ x_i \in 2^{\lambda_1} \pm [2^{\lambda_2 + 2\tau}]$$

Below is the special Σ-protocol for this relation, which the honest prover executes on $(x_i, e_i, w) \in \left(\ 2^{\lambda_1} \pm [2^{\lambda_2}] \ \times \ 2^{\gamma_1} \pm [2^{\gamma_2}] \ \times \ [2^{l_n - 2}] \ \right)$:

1. P_1 picks $(r_1, r_2, r_3, r_4) \leftarrow \pm[2^{\gamma_2 + 2\tau}] \times \pm[2^{\lambda_2 + 2\tau}] \times \pm[2^{\gamma_1 + l_n + 2\tau}] \times \pm[2^{l_n + 2\tau}]$, sets $(d_1, d_2, d_3, d_4, d_5) \leftarrow (T_1^{r_1} / (a^{r_2} y^{r_3}) \, , \, T_2^{r_1} / g^{r3} \, , \, g^{r_4} \, , \, g^{r_1} h^{r_4} \, , \, T_4^{r_2} \,)$, sets $r = (r_1, r_2, r_3, r_4)$, and outputs $a = (d_1, d_2, d_3, d_4, d_5)$.
2. Public coin challenge c is chosen as $c \leftarrow \{0,1\}^\tau$.
3. P_2 sets $z = (z_1, z_2, z_3, z_4)$ for $z_1 \leftarrow r_1 - c(e_i - 2^{\gamma_1})$, $z_2 \leftarrow r_2 - c(x_i - 2^{\lambda_1})$, $z_3 \leftarrow r_3 - c e_i w$, $z_4 \leftarrow r_4 - cw$ [all computed over integers]
4. V accepts if $z = (z_1, \ldots, z_4)$ lies in the cross-space $\boldsymbol{I'} = (I'_1 \times I'_2 \times I'_3 \times I'_4)$, for $I'_1 = \pm[2^{\gamma_2 + 2\tau + 1}]$, $I'_2 = \pm[2^{\lambda_2 + 2\tau + 1}]$, $I'_3 = \pm[2^{\gamma_1 + l_n + 2\tau + 1}]$, $I'_4 = \pm[2^{l_n + 2\tau + 1}]$, and if $a = f_V(\hat{x}, c, z)$ where $f_V(\hat{x}, c, z)$ computes $(d_1, \ldots, d_5)$ as follows:

$$d_1 \stackrel{?}{=} a_0^c T_1^{z_1 - c2^{\gamma_1}} / (a^{z_2 - c2^{\lambda_1}} y^{z_3}) \qquad d_2 \stackrel{?}{=} T_2^{z_1 - c2^{\gamma_1}} / g^{z_3}$$

$$d_3 \stackrel{?}{=} T_2^c g^{z_4} \qquad d_4 \stackrel{?}{=} T_3^c g^{z_1 - c2^{\gamma_1}} h^{z_4} \qquad d_5 \stackrel{?}{=} T_5^c T_4^{z_2 - c2^{\lambda_1}}$$

By the constraints on (x_i, e_i, w) used by an honest prover, z is statistically close to uniform over $\boldsymbol{I} = I_1 \times I_2 \times I_3 \times I_4$ where $I_1 = \pm[2^{\gamma_2 + 2\tau}]$, $I_2 = \pm[2^{\lambda_2 + 2\tau}]$, $I_3 = \pm[2^{\gamma_1 + l_n + 2\tau}]$, $I_4 = \pm[2^{l_n + 2\tau}]$. The proof of knowledge property of the ACJT proof system [ACJT00] satisfies the requirement that a valid witness $\hat{w} = ((A_i, x_i), e_i, w)$ is efficiently extractable from two accepting proof transcripts (a, c, z) and (a, c', z') s.t. $c' \neq c$, and this property holds for our extension which involves the check that $T_5 = T_4^{x_i}$.