Input-Aware Equivocable Commitments and UC-secure Commitments with Atomic Exchanges

Ioana Boureanu and Serge Vaudenay

Ecole Polytechnique Fédérale de Lausanne (EPFL)
{ioana.boureanu,serge.vaudenay}@epfl.ch

Abstract. We define a new primitive, *input-aware equivocable commitment*, baring similar hardness assumptions as plaintext-aware encryption and featuring equivocability. We construct an actual input-aware equivocable commitment protocol, based on a flavor of Diffie-Hellman assumptions allowing adversarially chosen domain parameters. On a parallel front, and since our commitment is extractable and equivocable in a straight-line way, we show that our commitment enjoys UC-security, when *atomic exchanges* are available as a UC setup. We further compare our protocol and our UC setup with similar, existing ones (i.e., in terms of efficiency, assumptions needed, etc.). Finally, we show that cryptography becomes UC-realizable in a natural way when participants are able to have "close encounters" or when atomic exchanges can be enforced onto the communication.

1 Introduction

An attractive, neat way to prove security of a protocol is to show that it realizes an ideal functionality [26,1,3,19] modelling a primitive. In this sense, a normal starting point is the well-known framework of Canetti's, i.e., the universal composability (UC) [7]. There are several versions of the UC framework (from [7] to [8]); slight differences are operated in the communication model, the order of quantifiers in the UC proofs, etc. In this paper, we will follow the original universal composability model, i.e., the one in [7], summarised below.

At a high level, a UC proof that a protocol is secure (in the bare UC model) means to show that no environment machine, $\mathcal{Z}$, can distinguish between the execution in the "real world" from the execution in the "ideal world". The "ideal world" contains "dummy" parties, the "target" ideal functionality (that the protocol is emulating) and the "ideal" adversary, I. These "dummy" parties simply send their inputs to the ideal functionality and wait for the response which they write on their output tapes. The environment $\mathcal{Z}$ gives the inputs to the parties and reads their local outputs and can communicate with I. The "real world" contains actual protocol participants, the environment $\mathcal{Z}$, the "real adversary" $\mathcal{A}$. The "ideal" adversary I or the "real" adversary $\mathcal{A}$ can corrupt protocol-parties, in which case the adversary will see the input of such a party, all communication sent to it, and $\mathcal{A}$ can decide its output. The communication channels between participants is assumed to be secure. So far, this perfectly describes the bare UC model which is often referred to as the *UC plain* model. In the UC plain model, several essential cryptographic protocols (e.g., commitments) are not realizable. Thus, the formalism is enhanced with some extra functionality, i.e., a *setup functionality*. Such an "empowering" add-on to the UC plain models yields the so-called *UC hybrid* models.

W. Susilo and R. Reyhanitabar (Eds.): ProvSec 2013, LNCS 8209, pp. 121–138, 2013.

UC Plain Models and Commitments. In the context of UC, we recall that if multiple commitments are UC-realized, then any multiparty computation can be UC-realized [11]. The UC functionality for single commitment is normally referred to as $\mathcal{F}_{COM}$ and can be assimilated to an ideal safe where to store the commitment. Another common functionality $\mathcal{F}_{MCOM}$ can deal with multiple commitments. Note that the general impossibility result of realizing UC commitment in the plain UC model is strongly linked to the notion of relay attacks.

UC Hybrid Models & Commitments. To achieve UC-secure (multiple) commitments, different UC setups have been used. We recall that UC-secure multi-commitment are generally realizable as follows: with a common reference string (CRS) setup [11], or with a public-key infrastructure (PKI) using a trusted party to manage the correct knowledge of respective public/secret keys [2], or with Katz's tamper-resistant hardware token [23] (under the computational Diffie-Hellman assumption and a static adversarial model), or with similar tokens to Katz's but susceptible to more powerful attacks [12], or with hardware tokens similar to those in [23], but used in a "receiver-empowering" fashion to minimize the computational assumptions. More recently, Damgård *et al.* [15] UC-realized multiple commitments by using a setup assumption that relaxes the tamper-resistant hardware token to a functionality that models the partial isolation and limited communication-power of a party. Unlike previous protocols, the protocol of Damgård's *et al.* [15] is in fact a general construction, relying on the following fact: if a functionality of isolated parties is available, then witness indistinguishable proofs of knowledge (WI PoK) can be realized, which further provide a PKI and make UC multiple commitments possible. In this setting, the UC-realization relies on the existence of one-way permutations and dense public key, IND-CPA secure encryption schemes with pseudorandom ciphertexts, but the adversarial model is strong (i.e., active and adaptive). In their paper, Damgård *et al.* [15] fully compare their functionality with that of tamper-evident hardware devices; we refer the reader to [15] for this comparison.

UC Augmented Models & Commitments. In fact, a UC-like scenario that made commitment possible is that of a communication augmented with pre-specified delays: i.e., the timing model of Kalai's *et al.* [22]. The assumptions under which multi-party computation becomes possible in this model are similar to some of the aforementioned assumptions for UC commitments with setups, i.e., the existence of enhanced trapdoor permutations and dense cryptosystems. However, whilst commitment in itself is not an issue anymore (i.e., the relay is prevented), Kalai's *et al.* [22] state that their model has the drawback of not being usable with protocols that employ time themselves (e.g., distance-bounding protocols [6]). But this may be unfortunate: as we will see further (i.e., in Section 3.2), time-sensitive protocols can in fact be themselves tightly linked to UC-secure protocols and their realization.

Our Justification for UC Hybrid Models with Atomic Exchanges. Summing up the above paragraphs, we can see that the ℓ-isolated parties of Damgård's *et al.* [15] can clearly be viewed as a restriction of the UC communication, as much as Kalai's *et al.* [22] model can. Thus, the former can also prevent relay attacks; moreover, ℓ-isolated parties do allow (and, in fact, facilitate) the composition of/with protocols that involve time themselves. And, as we envisage the usage of timed protocols (e.g., distance-

bounding protocols [6]), thus setting *à la* Kalai with delayed messages would be difficult to handle in our context. So, we embark on the approach of using UC setups, rather than augmentation of models with time/delays. In order to realize UC (multiple) commitments (and thus all multi-party computation as per [11]), we will invoke a UC-setup similar to the recent ℓ-isolated parties of Damgård's *et al.* To this end, we put forward a UC setup called $\mathcal{F}_{atomic}$. By *atomic exchanges* we mean the communication between protocol parties produced via their interaction with $\mathcal{F}_{atomic}$.

Our functionality $\mathcal{F}_{atomic}$ is similar to the "ℓ-isolated parties" setup of Damgård *et al.* [15]. The intuition behind is that the $\mathcal{F}_{atomic}$ functionality allows two parties to have an elementary, "fully isolated" exchange of *just one* message each. This can be viewed as a specialization of the $\mathcal{F}_{\ell\text{-isolated}}$ functionality of Damgård's *et al.* [15] (namely, with $\ell = 0$ and an exchanges limited to two messages in "one-round"). On the one hand, it is not clear how to realize $\mathcal{F}_{0\text{-}isolate}$ using $\mathcal{F}_{atomic}$. Intuitively, we need several instances of $\mathcal{F}_{atomic}$ and it would mean to pass information from one to the other using non-malleable encryption. So, $\mathcal{F}_{atomic}$ may be weaker than $\mathcal{F}_{0\text{-}isolate}$. On the other hand, $\mathcal{F}_{atomic}$ may be simpler to implement. For instance, the responder may be subject to several constraints such as time-bound to respond (like in NFC tags in distance-bounding [6]), or may be in a tamper proof token (such as the one by Katz [23]), or may result from a "close encounter".

Extrapolating PAW. In parallel, in this paper, we will define *input-aware equivocable commitments* (outside the UC model), a scheme akin in its characteristics to plaintext-aware encryption [14,21,31]. Our definition also includes equivocability, which is crucial for UC-security. We propose a specific protocol that implements this scheme under special types Diffie-Hellman assumptions. I.e., one such assumption is an extension of the DH regular knowledge assumption to be required to hold in any group [17]. In our case, the DH knowledge assumption needed is supposed to hold further in any adversarially chosen group (which is a weaker assumption than assuming it holds in any group). Also, in our UC setting, such a scheme can be employed in, e.g., concurrent RFID/NFC-based contactless payment protocols [25] where some computation is to be done *atomically* (i.e., by the RFID/NFC tag alone) and the final result needs to be "independent" for other simultaneous such computations.

UC Commitments and Their Assumptions. UC multiple commitments are possible under the different UC-setups. A short list of such setups is as follows: 1. Katz's tamper-resistant hardware tokens [23] (where under the computational Diffie-Hellman assumption and a static adversarial model); 2. similar tokens to Katz's but susceptible to more powerful attacks [12]; 3. hardware tokens similar to those in [23], but used in an asymmetric fashion to minimize the computational assumptions [28]; 4. the more recent [15] relaxation of the tamper-resistant hardware tokens to a functionality modelling the partial isolation and limited communication power of a party (under the assumptions of one-way permutations and dense public key, IND-CPA secure encryption schemes with pseudorandom ciphertexts, but the adversarial model is strong (i.e., active and adaptive).

There are some UC lines [10,16] in which the ideas underlying the ideal-world simulation of (multiple) commitment can be loosely linked to the one that we are going to put forward. Firstly, in [10], Canetti *et al.* achieve a $\mathcal{F}_{MCOM}$-realization with non-erasing parties, in the CRS-hybrid model using an encryption scheme obviously

samplable[14]. In this case, the trick that allows I to run its simulations (i.e., that gives I the oblivious-sampling coins for its ciphertext) is to sample ciphertexts without running the encryption algorithm. Note that an encryption obviously samplable (with respect to chosen-ciphertext attacks) [14] is possible under the Decisional Diffie-Hellman (DDH) assumption. Similarly, our protocol is possible if some special Diffie-Hellman assumptions are used.

Using several instances of $\mathcal{F}_{COM}$, ZK is UC-realized in the $\mathcal{F}_{COM}$-hybrid model [10] by mainstream ideas: by repeating t times, in parallel, Blum's protocol for Hamiltonian-Cycles (HC) [4], where the commitments of the provers are calls to $\mathcal{F}_{COM}$. Damgård and Nielsen [16] construct ZK more efficiently, but in a similar way, using the SAT protocol which proves satisfiability of boolean circuits. Along similar lines, our one-bit commitment can be used to $\mathcal{F}_{atomic}$-UC realize ZK in the same complexity as the Canetti's *et al.* [10]. In Appendix A we included a discussion about some further, "unconventional" commitments.

Our Contribution. In this paper, we introduce the notion of input-aware equivocable commitment, i.e., commitments that include both extractability and equivocability. We further propose some extensions of the Diffie-Hellman hardness assumptions or of the discrete logarithm hardness assumption, for the case where the adversary can maliciously select the group structure. We call it an *adversarially-chosen group* extension of the DH assumption. We propose the $\mathcal{F}_{atomic}$ functionality as a new setup assumption. This is a new, easy to implement UC setup, drawing upon un-aided local computation. Finally, we propose an input-aware equivocable commitment in the plain model, which we then prove to UC-realize $\mathcal{F}_{COM}$ in presence of the $\mathcal{F}_{atomic}$ setup.

2 Input-Aware Commitments in Classical Cryptography

In this section, we formalize the notion of input-aware equivocable commitments and present one protocol. On our way to doing so, we specify different flavors of Diffie-Hellman (DH) assumptions.

2.1 Commitment Scheme

The following definition reiterates the usual meaning of a commitment scheme in conformity with traditional (i.e., non-composable) cryptography.

Definition 1 (Commitment Scheme). *A* bit-commitment scheme *in terms of a security parameter λ is a pair of polynomially bounded protocols $((S_{COM}, R_{COM}), (S_{OPEN}, R_{OPEN}))$ where S_{COM} has an input bit b, and R_{OPEN} has an output bit $\bar{b}$. The protocols may abort. The*

$$S_{COM}(1^{\lambda}, b; r_S) \leftrightarrow R_{COM}(1^{\lambda}; r_R)$$

execution[1] *is called the* commitment phase. *For simplicity, 1^{λ} is omitted from the notation. Let View_S, respectively View_R, denote the view of S_{COM}, respectively the view of R_{COM}. The*

$$S_{OPEN}(\mathsf{View}_S; r'_S) \leftrightarrow R_{OPEN}(\mathsf{View}_R; r'_R)$$

[1] This execution is understood as any standard interactive system [20].

execution is called the opening phase*. It produces the final output from R, i.e., $\bar{b}$. A commitment scheme is expected to be* correct*: i.e., when correctly executed, no protocol aborts and $\bar{b} = b$.*

The following definition completes the above by formalizing the usual requirements of a commitment scheme in conformity with traditional (i.e., non-composable) cryptography.

Definition 2 (The Hiding Property). *A commitment scheme is said to be* hiding *if the following holds. For any polynomially bounded R^*_{COM}, if $S_{COM}(b;r_S) \leftrightarrow R^*_{COM}(r_R)$ ends up with the final view View_R for R^*_{COM}, then $\mathsf{View}_R|b=0$ and $\mathsf{View}_R|b=1$ are computationally indistinguishable.*

In the above, $\mathsf{View}_R|b=x$ (with $x \in \{0,1\}$) denotes the marginal distribution (over all random coins and inputs) of View_R as a random variable, conditioned to the event $b=x$. Note that we can assume without loss of generality that R^*_{COM} is deterministic (since r_R could be hard-coded in it).

Definition 3 (The Binding Property). *A commitment scheme is said to be* binding *if the following holds. For any polynomially bounded S^*_{COM} and S^*_{OPEN}, if the $S^*_{COM}(r_S) \leftrightarrow R_{COM}(r_R)$ and then the $S^*_{OPEN}(\mathsf{View}_S; r'_S) \leftrightarrow R_{OPEN}(\mathsf{View}_R; r'_R)$ experiment occur, then $\min(\Pr[\bar{b}=0|r_S,r_R],\Pr[\bar{b}=1|r_S,r_R]) = \mathsf{negl}(\lambda)$, where this probability is taken in the random choices of S^*_{OPEN} and R_{OPEN}.*

This means that once the commitment is made (i.e., r_S and r_R are fixed), S^*_{OPEN} cannot open to both $\bar{b}=0$ and $\bar{b}=1$. We recall that $f(\lambda)=\mathsf{negl}(\lambda)$ means that for all $c>0$, we have $f(\lambda)=O(\lambda^{-c})$.

2.2 Diffie-Hellman Assumptions

In this subsection, we specify several Diffie-Hellman assumptions.

Definition 4 (DH Key Generator). *A* DH key *is a tuple $K=(G,q,g)$ such that G is a group, q is a prime dividing the order of G, g is an element of G of order q. A* DH key-generator *is a ppt. algorithm Gen producing DH keys K such that $|K| = Poly(\log q)$ and the operations (i.e., multiplication, comparison, and membership checking in the group $\langle g\rangle$ generated by g) over their domain can be computed in time $Poly(\log q)$. We say that (S,S') is a valid K-*DH pair for g^{σ} *if $S \in \langle g\rangle$ and $S' = S^{\sigma}$, where $\sigma \in \mathbb{Z}_q$. Given $K=(G,q,g)$, we define a function DH_K with a variable number of inputs from G by $\mathsf{DH}_K(g^{x_1},\ldots,g^{x_n}) = g^{x_1\cdots x_n}$.*

An example of a DH key is $(\mathbb{Z}^*_p,q,g)$ where p and q are primes and $p=2q+1$, $g \in \mathsf{QR}(p)$, $g \neq 1$.

We now strengthen the Decisional Diffie-Hellman (DDH) assumption. Below, we use an arbitrary ppt. algorithm $\mathcal{B}$ generating some coins ρ and a state $\mathtt{state}$. Such coins ρ and/or state $\mathtt{state}$ will be sometimes used as auxiliary inputs to some ITMs in the security games formalized below.

Definition 5 (DDH Asmpt. in an Adversarially-Chosen Group (ag-$\mathsf{DDH}_{\mathsf{Gen}}$)). *The* ag-$\mathsf{DDH}_{\mathsf{Gen}}$ *assumption over a domain of DH keys $\mathcal{K}$ states that for any ppt. algorithms $\mathcal{A}$ and $\mathcal{B}$ in the next game,* $\Pr[b = \overline{b}] - \frac{1}{2} = \mathsf{negl}(\lambda)$*:*

1: $(\rho, \mathsf{state}) := \mathcal{B}(1^\lambda; r_{\mathcal{B}})$
2: $K := Gen(1^\lambda; \rho)$
3: *define* (G, q, g) *from* K
4: *pick* $\alpha, \beta, \gamma \in_U \mathbf{Z}_q$
5: $A := g^\alpha$; $B := g^\beta$; $C_0 := g^\gamma$; $C_1 := g^{\alpha\beta}$
6: *pick* $b \in_U \{0,1\}$
7: $\overline{b} := \mathcal{A}(1^\lambda, \mathsf{state}, A, B, C_b; r)$

The probability stands over the random coins $r_{\mathcal{B}}$, r, $b \in_U \{0,1\}$ and $\alpha, \beta, \gamma \in_U \mathbf{Z}_q$ and is negligible in terms of $\log q$*. $\mathcal{A}$ (and $\mathcal{B}$) run in ppt. in terms of* $\log q$*.*

It should be clear that ag-$\mathsf{CDH}_{\mathsf{Gen}}$, the computational version of this problem can be defined as well.

Definition 6 (CDH^n Asmpt. in an Adversarially-Chosen Group (ag-$\mathsf{CDH}^n_{\mathsf{Gen}}$)). *The* ag-$\mathsf{CDH}^n_{\mathsf{Gen}}$ *assumption over a domain of DH keys $\mathcal{K}$ states that for any ppt. algorithms $\mathcal{A}$ and $\mathcal{B}$ in the next game, the probability that $S_0 = \mathsf{DH}_K(A, B, S_1, \ldots, S_n)$ and that $S_i \neq 1$ for $i = 1, \ldots, n$ is negligible:*

1: $(\rho, \mathsf{state}) := \mathcal{B}(1^\lambda; r_{\mathcal{B}})$
2: $K := Gen(1^\lambda; \rho)$
3: *define* (G, q, g) *from* K
4: *pick* $\alpha, \beta \in_U \mathbf{Z}_q$
5: $A := g^\alpha$; $B := g^\beta$
6: $(S_0, S_1, \ldots, S_n) := \mathcal{A}(1^\lambda, \mathsf{state}, A, B; r)$

The probability stands over the random coins $r_{\mathcal{B}}$, r, and $\alpha, \beta \in_U \mathbf{Z}_q$. The probability is negligible in terms of $\log q$*. $\mathcal{A}$ (and $\mathcal{B}$) run in ppt. in terms of* $\log q$*.*

The standard Diffie-Hellman computational problem corresponds to the CDH^0 problem. Clearly, the CDH^n assumption implies the CDH^{n-1} assumption for all $n > 0$, but the opposite implication is an open problem. In what follows, we will use the CDH^1 assumption.

We now similarly strengthen the Diffie-Hellman knowledge (DHK0) assumption (for a summary the latter, refer to [17]).

Definition 7 (DHK0 Asmpt. in an Adversarially-Chosen Group (ag-$\mathsf{DHK0}_{\mathsf{Gen}}$)). *The* ag-$\mathsf{DHK0}_{\mathsf{Gen}}$ *assumption over a domain of DH keys $\mathcal{K}$ states that for any ppt. algorithm $\mathcal{A}$ and $\mathcal{B}$ in the next game, there is a polynomially bounded algorithm $\mathcal{E}$ such that the probability of the below experiment outputting 1 is negligible:*

1: $(\rho, \mathsf{state}) := \mathcal{B}(1^\lambda; r_{\mathcal{B}})$
2: $K := Gen(1^\lambda; \rho_\lambda)$
3: *define* (G, q, g) *from* K
4: *pick* $\sigma \in_U \mathbf{Z}_q$
5: $(S, S') := \mathcal{A}(1^\lambda, \mathsf{state}, g^\sigma; r)$
6: *if* (S, S') *is not a valid K-DH pair for* g^σ*, then return* 0

7: $s := \mathcal{E}(1^\lambda, \mathsf{state}, g^\sigma, r)$
8: *if* $S = g^s$, *then return 0*
9: *return 1*

The probability stands over the random coins $r_{\mathcal{B}}$, r *and* $\sigma \in_U \mathbf{Z}_q$ *and is negligible in terms of* $\log q$. *The running time of* $\mathcal{E}$ *(and* $\mathcal{B}$*) is ppt. in terms of* $\log q$.

This assumption means that whatever the algorithm producing valid DH pairs for a random g^σ with σ unknown, this algorithm must know the discrete logarithm of their components except for some negligible cases.

The algorithm $\mathcal{B}$ used in the games above is denoted as the *biotope* algorithm.

What distinguishes these assumptions from the mainstream DDH and DHK0 assumptions [17] is that these should hold for all K selected by a ppt. biotope algorithm (even by a malicious one) and not only for some K which is randomly selected by an honest participant. In fact, when selecting a DH key without a CRS in a two party protocol, the above assumption must hold for any maliciously selected K (since we ignore a priori which party is honest). Hence, the name we use: DH assumptions in an adversarially-chosen group. As we mentioned in the introduction, the latter assumption is a special case of the DH knowledge assumption required to hold in any group, or, equivalently, for any $\mathcal{B}$ and $r_{\mathcal{B}}$. Such assumptions were originally introduced by Dent in [17]. Here, we do not require the assumption to hold in any group, but rather in those groups G for which we can produce a seed for *Gen* to use in generating G, or equivalently, for any, $\mathcal{B}$ on average over $r_{\mathcal{B}}$.

In the next, for readability purposes, we will omit the additional-input 1^λ from the inputs of the machines that take it, its presence being implicit.

2.3 Input-Aware Equivocable Bit-Commitment

Definition 8 (Input-Aware Equivocable Commitment Scheme). *An* input-aware equivocable bit-commitment (IAEC) scheme *is a commitment scheme* $((S_{COM}, R_{COM}), (S_{OPEN}, R_{OPEN}))$ *as per Def. 1, with the following additional properties. Let* b *denote the input of* S_{COM}, $\bar{b}$ *be the output of* R_{OPEN} *or* R^*_{OPEN}, *and* View_S, *respectively* View_R, *be the view of* S_{COM} *or* S^*_{COM} *and, respectively, of* R_{COM} *or* R^*_{COM} *in the commitment phase.*

– *(sender input-awareness aka extractability) For any polynomially bounded algorithms* S^*_{COM} *and* S^*_{OPEN}, *there is a polynomially bounded algorithm* $\mathsf{Extract}$ *such that the following holds. When running the commitment phase*

$$S^*_{COM}(r_S) \leftrightarrow R_{COM}(r_R),$$

followed by the opening phase

$$S^*_{OPEN}(\mathsf{View}_S; r'_S) \leftrightarrow R_{OPEN}(\mathsf{View}_R; r'_R),$$

the next holds with probability $1 - \mathsf{negl}(\lambda)$, *taken over the random* r_S, r'_S, r_R, r'_R:
– $\bar{b} = \mathsf{Extract}(\mathsf{View}_S)$ *and no protocol aborts,*
– *or* $\mathsf{Extract}(\mathsf{View}_S)$ *aborts and the commitment phase as well,*
– *or the opening phase aborts.*

– *(receiver self-equivocability) For any polynomially bounded algorithm* R^*_{COM} *and* R^*_{OPEN}*, there is a polynomially bounded algorithm* Equiv *such that the following holds. When running the commitment phase*

$$S_{COM}(b;r_S) \leftrightarrow R^*_{COM}(r_R),$$

followed by the flipping a coin b' *to run the opening phase*

$$\begin{cases} S_{OPEN}(\mathsf{View}_S; r'_S) \leftrightarrow R^*_{OPEN}(\mathsf{View}_R; r'_R), & \text{if } b' = b \\ \mathsf{Equiv}(b', \mathsf{View}_R; r'_S) \leftrightarrow R^*_{OPEN}(\mathsf{View}_R; r'_R), & \text{if } b' = 1-b, \end{cases}$$

it all results in a final view View'_R *of* R^*_{OPEN} *and this is such that* $\mathsf{View}'_R|b=0$ *and* $\mathsf{View}'_R|b=1$ *are computationally indistinguishable over the random* r_S, r_R, r'_R, r'_S *and* b'.

The above definition implies the classical notions of security (i.e., notions of hiding and binding commitments as per Defs. 2, 3). Equivocability already says that $\mathsf{View}_R|b=0$ and $\mathsf{View}_R|b=1$ are indistinguishable since View_R is included in View'_R; so the commitment is hiding. Furthermore, a malicious sender who could open a commitment to both $b=0$ and $b=1$ with a probability which is negligible would contradict $\bar{b} = \mathsf{Extract}(\mathsf{View}_S)$; so, the commitment is binding.

We will now construct an IAEC based on the ag-$\mathsf{DHK0}_{\mathsf{Gen}}$, the ag-$\mathsf{DDH}_{\mathsf{Gen}}$ and the ag-$\mathsf{CDH}^1_{\mathsf{Gen}}$ assumptions. We denote it as protocol Π_{Gen} (see Fig. 1). As per Section 3.2, the label "atomic" in Fig. 1, applies only in the context of the use of a UC functionality for atomic exchanges when building the protocol to be UC-secure. It shall be ignored in the current section.

Protocol Π_{Gen}

The commitment phase (i.e., to be described by the S_{COM} and R_{COM} protocols) works as follows.

1. S generates ρ for Gen, i.e., it does $K := Gen(\rho)$, and S sends ρ to R.
2. Then, R also computed $K := Gen(\rho)$ and R selects[2] some $\alpha \in \mathbf{Z}^*_q$ and sends $X_0 := g^\alpha$ to S.
3. S verifies[3] that $X_0 \in \langle g \rangle$, selects $x \in \mathbf{Z}^*_q$, calculates $X := g^x$ and $X' := X_0^x$, and sends X, X' to R. S picks $\beta \in \mathbf{Z}^*_q$ and calculates $Y_0 := g^\beta$. S sends Y_0 to R.
4. R verifies that $X \in \langle g \rangle$, $X' = X^\alpha$, and that $Y_0 \in \langle g \rangle$. Then, R selects $y \in \mathbf{Z}^*_q$ and calculates $Y := g^y$ and $Y' := Y_0^y$. Then, R sends Y, Y', and α to S. Then, R selects some $z_0, z_1 \in \mathbf{Z}^*_q$ and calculates Z_0 and Z_1 as follows: $Z_i := g^{z_i}$, for $i \in \{0,1\}$. The R party sends Z_0 and Z_1 to S.
5. The party S verifies that $Y, Z_0, Z_1 \in \langle g \rangle$, that $Y' = Y^\beta$, and that $X_0 = g^\alpha$. S further selects $r \in \mathbf{Z}_q$ and sends $U := g^r$, $V := Z_b X^r$ and β to R, where b is the bit that S is in the process of committing to.
6. R verifies that $U, V \in \langle g \rangle$ and that $Y_0 = g^\beta$.

[2] All occurrences of "selects" in this description denote "picks uniformly".

[3] If a verification fails, then the party running it aborts.

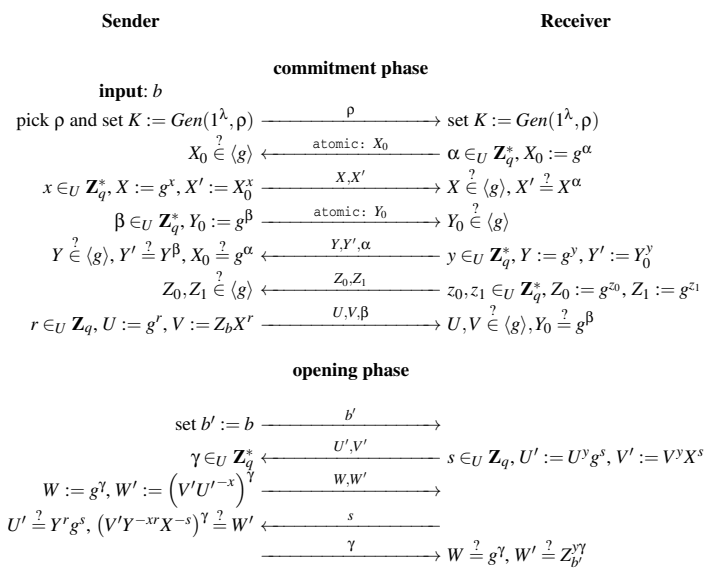

Fig. 1. Input-aware Equivocable Commitment Protocol Π_{Gen}

The opening phase (i.e., to be described by the S_{OPEN} and R_{OPEN} protocols) works as follows.

1. S sends a bit b' with $b' = b$.
2. Then, R selects $s \in \mathbf{Z}_q$ and calculates $U' := U^y g^s$ and $V' := V^y x^s$. Then, R sends U' and V' to S.
3. S selects $\gamma \in \mathbf{Z}_q^*$ and calculates $W := g^\gamma$ and $W' := (V'U'^{-x})^\gamma$. Then, S sends W, W' to R.
4. R sends s to S.
5. S verifies that $U' = Y^r g^s$ and $(V'Y^{-xr}X^{-s})^\gamma = W'$. Then, R sends γ to S.
6. S verifies that $W = g^\gamma$, $W' = Z_{b'}^{y\gamma}$ and outputs $\bar{b} := b'$.

The commitment is an ElGamal encryption (U,V) of Z_b with a self-made public key X. The opening uses the homomorphic properties of the encryption to transform (U,V) into an encryption of Z_b^y such that the following holds: if $Z_{b'}$ were not the correct decryption of (U,V), then decrypting $Z_{b'}^y$ would require to know y or $z_{b'}$ (since $Z_{b'}^y = (g^{z_{b'}})^y$ is equal to the "public" $W'^{\frac{1}{\gamma}}$). The trick is that keys X and Y are declared in such a way that the DHK0 assumption would make the corresponding secret-keys x and y extractable by using input-aware equivocable techniques when given the appropriate coins. Indeed, x would allow to extract b from the commitment and y would allow to equivocate.

Theorem 9. *Under the* $\mathsf{ag\text{-}CDH}^1_{\mathsf{Gen}}$, $\mathsf{DDH}_{\mathsf{Gen}}$, *and* $\mathsf{ag\text{-}DHK0}_{\mathsf{Gen}}$ *assumptions, the protocol* Π_{Gen} *above is an input-aware equivocable bit-commitment.*

Proof (space-constrained sketch). Since the polynomial-time bound and the correctness are trivial, we only have to construct Extract and Equiv.

Sender Input-Awareness. Let S^*_{COM} and S^*_{OPEN} be some malicious commitment and opening algorithms, respectively. We define two algorithms $\mathcal{A}$ and $\mathcal{B}$ as follows. The algorithm $\mathcal{B}$ simulates the experiment $S^*_{COM}(r_S) \leftrightarrow R_{COM}(r_R)$ up to the moment before S^*_{COM} receives X_0, when $\mathcal{B}$ stops. Then, as per dictated by the ag-DHK0$_{\mathsf{Gen}}$ game, $\mathcal{B}$ sets ρ and state according to the experiment he just took part in. That is ρ would be as generated in $S^*_{COM}(r_S) \leftrightarrow R_{COM}(r_r)$ and state would be the current view of S^*_{COM} with its coins limited to its run so far, i.e., limited to a prefix $\overline{r}_s$ of the whole set of coins r_s ($r_s := \overline{r}_s || \overline{\overline{r_s}}$). Then the output (X,X') of S^*_{COM} with input state, augmented with the message X_0 and the coins $\overline{\overline{r_s}}$ defines $\mathcal{A}(\mathsf{state}, X_0; \overline{\overline{r_s}})$. By the ag-DHK0$_{\mathsf{Gen}}$ assumption, there must exist some algorithm $\mathcal{E}(\mathsf{state}, X_0; \overline{\overline{r_s}})$ such that —except for negligible cases— $\mathcal{E}(\mathsf{state}, X_0; \overline{\overline{r_s}})$ outputs x satisfying that $X = g^x$, or R_{COM} rejects (X,X').

Now, let $r_s = \overline{r}_s || \overline{\overline{r_s}}$ be the coins in View_S and $\mathsf{state}, X_0, Z_0, Z_1$ as above be in View_S. We now define $\mathsf{Extract}(\mathsf{View}_S)$ as follows. Let $\rho := S^*_{COM}(r_S)$ and $(X,X') := S^*_{COM}(X_0; r_S)$. Except in negligible cases, $x = \mathcal{E}(1^\lambda, \mathsf{state}, X_0; r_S)$ is such that $X = g^x$. If (U,V) is valid, Extract can compute $Z = VU^{-x}$ and compare Z to Z_0 and to Z_1. If there is no match, then we return $\perp$. Otherwise, we return b as per the match $Z_b = Z$. Note that $\Pr[Z_0 = Z_1]$ is negligible, so there is a unique match.

Now, we need to show the soundness of this procedure, i.e., S^*_{OPEN} cannot open to something different from $b = \mathsf{Extract}(\mathsf{View}_S)$. For this, we show that S^*_{COM} and S^*_{OPEN} could define an adversary for ag-CDH$^1_{\mathsf{Gen}}$. We will use a rewinding technique to define this adversary. (Note that extraction is straight-line. It is only the adversary showing that extraction is sound which is using rewinding.)

To define the adversary (using the created ρ) receiving A and B from outside, we first simulate the experiment until we get β. Then, we rewind it but inject $Y = A$ instead of some Y with a known discrete logarithm. We can also compute $Y' = Y^\beta$ thanks to getting β. Similarly, we flip a coin $\tilde{b}$ and inject $Z_{\tilde{b}} = g^{z_{\tilde{b}}}$ with $z_{\tilde{b}}$ random and $Z_{1-\tilde{b}} = B$. Clearly, β is bound to be unchanged. Since View_S has a correct distribution, we can still run $b = \mathsf{Extract}(\mathsf{View}_S)$ and $x = \mathcal{E}(1^\lambda, \mathsf{state}, X_0, r_S)$. If $b \neq \tilde{b}$, this is bad luck and we restart. Since S^*_{COM} sees no information about $\tilde{b}$, bad luck happens with probability $\frac{1}{2}$ and we do not have to restart too much until we are in the lucky $b = \tilde{b}$ case.

Then, the adversary continues to simulate the opening. If $b' = b$, the adversary aborts. Otherwise, the adversary must simulate some genuine (U',V'). We know that $V = g^{z_b} U^x$. The regular receiver would send a random $U' = U^y g^s$ and some $V' = Y^y X^s$ connected to U' with the relation $V' = Y^{z_b}(U')^x$. So, the simulator could just pick U' at random and compute $V' = Y^{z_b}(U')^x$ since he knows $z_b = z_{\tilde{b}}$ and x. He then obtains from S^*_{OPEN} some (W,W'). With a genuine receiver sending s, we obtain γ such that

$$\mathsf{DH}(A,B,W) = \mathsf{DH}(Y, Z_{1-b}, g^\gamma) = Z_{1-b}^{y\gamma}$$

So, to make the receiver accept, the (W,W') pair we must satisfy $\mathsf{DH}(A,B,W) = W'$ even before providing s. Due to the ag-CDH$^1_{\mathsf{Gen}}$ assumption, this happens with negligible probability. So, in the genuine experiment, either the experiment aborts, or $b' = \mathsf{Extract}(\mathsf{View}_S)$, or $W' \neq \mathsf{DH}_K(Y, Z_{1-b'}, W)$, thus making R_{OPEN} aborts.

Receiver Self-Equivocability. Let R^*_{COM} and R^*_{OPEN} be some malicious commitment and opening algorithms. We define two algorithms $\mathcal{A}$ and $\mathcal{B}$ as follows. The algorithm $\mathcal{B}$ simulates the experiment $S_{COM}(r_S) \leftrightarrow R^*_{COM}(r_R)$ until the moment before R^*_{COM} receives Y_0 and then $\mathcal{B}$ stops. As before, $\mathcal{B}$ will produce his needed ρ as in the experiment $S_{COM}(r_S) \leftrightarrow R^*_{COM}(r_R)$ and state as the current view of R^*_{COM}, limiting his coins r_R to $\overline{r_R}$, i.e., to those used so far, where $r_R := \overline{r_R}||\overline{\overline{r_R}}$. Then the output (Y,Y') of R^*_{COM} with input state, augmented with the message Y_0 and the coins $\overline{\overline{r_R}}$ defines $\mathcal{A}(\mathsf{state}, Y_0; \overline{\overline{r_R}})$. Due to the ag-$\mathsf{DHK0}_{\mathsf{Gen}}$ assumption, there must exist some algorithm $\mathcal{E}$ such that, except for negligible cases, $\mathcal{E}(\mathsf{state}, Y_0; \overline{\overline{r_R}})$ produces y satisfying $Y = g^y$, or S_{COM} rejects (Y,Y').

We define all messages as in the $S_{COM}(b; r_S) \leftrightarrow R^*_{COM}(r_R)$ experiment from the view View_S. Note that running $S_{COM}(b; r_S)$ also defines ρ.

We define $\mathsf{Equiv}(b', \mathsf{View}_R; r'_S)$ by sending out b', receiving U', V', computing $y = \mathcal{E}(\mathsf{state}, Y_0; \overline{\overline{r_R}})$ constructed like above, computing $Z^y_{b'}$ and producing the pair (W,W') such that $W' = \mathsf{DH}_K(Y, Z_{b'}, W)$, by $W = g^\gamma$ and $W' = \left(Z^y_{b'}\right)^\gamma$.

The view of R includes ρ, X, X', Y_0, U, V, β, b', W, W', γ. In all cases, W, W', γ can be simulated by R with the same distribution, as well as Y_0, β. Since α is produced by R, X' can be simulated as well. Finally, the view reduces to (ρ, X, U, V, b'). Indeed, distinguishing $b = 0$ from $b = 1$ with b' random reduces to the semantic security of the ElGamal cryptosystem. As proven in [5], this reduces to the Decisional Diffie-Hellman (DDH) problem. □

3 UC-Secure (Input-Aware Equivocable) Commitment with a "Mild" Setup

In Subsection 3.1, we introduce the UC functionality called $\mathcal{F}_{atomic}$, which is needed as UC setup for the UC-realization of our (IAEC) commitment. The actual UC-realization of commitment is shown in Subsection 3.2; some discussions about this realization and its relationships with existing lines of UC-realization of commitment are also included.

3.1 UC Setup Functionality for Atomic Exchanges

We will now present a UC functionality that models *one* exchange of messages between two parties, one of which is in complete isolation; hence, the name *atomic* exchange. The restriction to one exchange makes this functionality a specialization of the $\mathcal{F}_{\ell-\text{isolate}}$ of Damgård's *et al.* [15]. Also, differently from [15], the functionality below draws strictly upon the user on which the limited communication is enforced; in that sense, in the functionality below, this user can update its algorithm sent to the functionality several times before the actual computation is made.

The $\mathcal{F}_{atomic}$ Functionality of Atomic Exchanges. Let *poly* be a polynomial. Assume two parties A and B that would like to have an atomic exchange, i.e., A would normally send m to B and, without outside help, B would have to respond with m'. Mainly, this lack of outside help and the *one* exchange are the core of the $\mathcal{F}_{atomic}$ functionality.

Request for Atomicity. The participant B sends a message $(\textbf{atomic}, A, B, M)$ to $\mathcal{F}_{atomic}$, where M denotes description of the Turing machine[4] run by B. The functionality $\mathcal{F}_{atomic}$

[4] We assume that this machine is deterministic.

parses the message and stores (A,B,M). Any other tuple including the same (A,B) is erased.[5] A special case is where the participant B sends the message $(\textbf{atomic},A,B,\perp)$, which counts for an abortion of the atomic session.

Challenge an Atomic Response. The participant A can send the command $(\textbf{challenge},A,B,m)$ to $\mathcal{F}_{atomic}$. In this case, the functionality verifies the existence of a tuple (A,B,M). If the corresponding register is empty or if $M=\perp$, then the functionality sends a reject message to A and to the ideal adversary. Otherwise, the machine proceeds as follows. It runs $M(m)$ for no more than $poly(|m|)$ steps, finally storing the result in m'. Then, it sends $(\textbf{challenge-issued},A,B,m)$ to B and $(\textbf{response},A,B,m')$ to A. The (A,B,M) tuple is then erased.

Again, this functionality models the fact that B does not communicate with another participant in between receiving m and producing his response m', that before "being asked" to compute m in isolation the participant can update his machine and that this computation/communication is supposed to capture one exchange only. As we said in the introduction and in the related-work, this functionality is a specialization of the $\mathcal{F}_{\ell-\text{isolated}}$ in [15], where $\ell=0$, the exchange is reduced to one message per each of the two parties involved and where the machine of the "computing-party" can be updated before the need for the computation is imminent. In that sense, one cannot say clearly if our functionality is weaker or stronger than the $\mathcal{F}_{\ell-\text{isolated}}$ functionality in [15].

Further, we note that this sort of setup is sufficient for bypassing a relay attack of the sort that lead to the impossibility of UC-commitments in the plain model. In the same time, especially for the cases where only two parties are involved (e.g., the aforementioned mutually independent commitments [24]), this sort of setup is suitable to bypass the known malleability problems.

In practice, a possible way to implement such an atomic-exchange functionality is given by distance-bounding protocols [6]. This is one of the actual methods implemented to prevent relay attacks [18]. Namely, to achieve the atomic-exchange, the two concerned parties can use –in an initial/certain part of the communication– a distance-bounding protocol (or a slight modification of such a protocol, which still considers the time-of-flight of the messages in accepting/rejecting them). I.e., the correct answer could have been produced only and solely by the close-by partner, otherwise the distance-bound would be broken.

To easily specify protocols using atomic exchanges, the $(\textbf{challenge},A,B,m)$ query by A it simply denoted "`atomic:` m". It is followed by the message answering $M(m)$ by B, due to an abuse of notation. This implicitly means that B must have committed M to $\mathcal{F}_{atomic}$ before.

3.2 UC-realization of Commitment in the $\mathcal{F}_{atomic}$-hybrid Model

It is easy to see that any input-aware equivocable commitment UC-realizes commitment using $\mathcal{F}_{0\text{-}isolate}$: we just have to run S and R in isolation. Here, we strengthen the result by relying on $\mathcal{F}_{atomic}$ only. The Π_{Gen} protocol, presented in Fig. 1 also requires some messages to be exchanged atomically, i.e., using the $\mathcal{F}_{atomic}$ functionality. This means

[5] Note that –by the above– B can resend this command to $\mathcal{F}_{atomic}$, possibly with a different machine-description M.

that if R wants S to compute X, X' on his own based upon S's view and the fresh receipt of X_0, then they establish an *atomic exchange*: S cooperates in this and sends (several) $(\textbf{atomic}, R, S, \mathsf{algo\_of\_S})$ to $\mathcal{F}_{atomic}$, where $\mathsf{algo\_of\_S}$ computes (X, X') from the (hard-coded) partial view of S and the input X_0. We consider only the last deposited $\mathsf{algo\_of\_S}$. Then, R sends $(\textbf{challenge}, R, S, X_0)$ to $\mathcal{F}_{atomic}$, which will eventually send X_0 to S and X, X' to R, with $(X, X') := \mathsf{algo\_of\_S}(X_0)$ with $\mathsf{algo\_of\_S}$ running up to $poly(|X_0|)$ in time. The same goes for the $Y_0 \mapsto (Y, Y')$ atomic exchange.

We are now going to prove that the Π_{Gen} protocol UC-realizes commitment.

Theorem 10. *Under the* $\mathsf{ag\text{-}CDH}^1_{\mathsf{Gen}}$, $\mathsf{DDH}_{\mathsf{Gen}}$, *and* $\mathsf{ag\text{-}DHK0}_{\mathsf{Gen}}$ *assumptions, in the* $\mathcal{F}_{atomic}$*-hybrid UC model in the presence of static, non-adaptive adversaries, the protocol* Π_{Gen} *UC-realizes* $\mathcal{F}_{\mathsf{COM}}$.

The proof is very similar to the one of Th. 9. We construct an ideal adversary I by using the straight-line extraction of b (when the sender is corrupted) or the straight-line equivocation (when the receiver is corrupted). In the first case, we use the extracted b to commit to it. In the latter case, I simulates the commitment to a dummy bit b to R^*_{COM}, then we use the equivocation once b' is opened by the functionality to simulate the opening to b'.

We note that the constructed I does not require rewinding. However, to prove that I works well, we do rewind algorithms, but this is allowed. The (sketch of) proof is given in Appendix B.

Discussions about the UC-realization of $\mathcal{F}_{COM}$ *by* Π_{Gen}. We underline that, as per Fig. 1, after the initialization phase, the two parties involved are in the position where they share (amongst other things) the tuple (X, Y). This part can be separated and viewed realizing itself a particular key-sharing functionality (call it $\mathcal{G}$) in a $\mathcal{F}_{atomic}$-hybrid UC model. Then, the UC-realization in Th. 10 can be cast as follows: "in the $\mathcal{G}$-hybrid UC model in the presence of static, non-adaptive adversaries, the protocol Π'_{Gen} (i.e., Π_{Gen} without its init phase exchanging X and Y) UC-realizes $\mathcal{F}_{COM}$ (if the $\mathsf{ag\text{-}DHK0}_{\mathsf{Gen}}$ assumption, the $\mathsf{ag\text{-}DDH}_{\mathsf{Gen}}$ and the $\mathsf{ag\text{-}CDH}^1_{\mathsf{Gen}}$ assumption hold)."

The formulation above renders our result visibly closer to the result in [15]. Namely, if a setup functionality restricting the communication is available, then this leads to some key-establishment, which then leads to the UC-realization of commitment. However, the difference between our approach here and the one in [15] is that secret extraction is integrated based on input-awareness, and we do not need to run a multi-round protocol in isolation: only an elementary challenge-response one. Finally, this indicates that cryptography becomes UC-realizable in a natural way when participants are able to have "close encounters" to exchange public-key material.

ZK is UC-realized in the $\mathcal{F}_{COM}$-hybrid model [10] by mainstream ideas: by repeating t times, in parallel, Blum's protocol for Hamiltonian-Cycles (HC) [4], where the commitments of the provers are calls to $\mathcal{F}_{COM}$. Thus, our one-time one-bit-commitment can be used to UC realize ZK in the same complexity as the Canetti's *et al.*

Damgård and Nielsen UC-realize a commitment UC-functionality called $\mathcal{F}_{HCOM}$ [16], for homomorphic commitment. This functionality is slightly different from the original $\mathcal{F}_{MCOM}$; there the difference stems from the increased efficiency sought and, most importantly, from the way to achieve equivocability and extractability for the ideal adversary I. In the introduction, we recalled the so-called UC-"mixed commitments" [16]

by Damgård and Nielsen, which achieve their equivocability and extractability for the ideal adversary I by basing their commitment on two, disjoint sets of keys: the E-keys (for the perfectly hiding property and equivocability by I), and on the X-keys (for the perfectly binding property and extractability by I). For the simulation to work, only a part of the key (formed of E-keys [16], used for the perfectly hiding property and equivocability by I) is placed in the reference string. The Damgård and Nielsen commitments are inherently based on non-erasure Σ-protocols and their security against lunchtime opening [16], i.e., roughly, an adversary is unable to produce an arbitrary opening for a commitment, even if he sees several fake commitments under E-keys and can adaptively specify how these ones should be opened. One such commitment protocol is based on the p-subgroup assumption [29] and another assumes hardness of the decisional composite residuosity problem [30] used in Paillier's cryptosystem. We believe that our construction can be extended also exploiting the Paillier encryption, to commit to more than one bit. Damgård and Nielsen [16] construct ZK efficiently using their commitments on top of the SAT protocol which proves satisfiability of boolean circuits.

Using our approach, we can further realize a PKI in a natural way. What we need is to establish a link between each participant and a central authority, then UC-realize key registration based on commitment using standard proof-of-knowledge techniques. Based on the PKI, we can realize multiparty computation. Our technique also makes it easier to realize 2-party computation is a light way.

4 Conclusions

In this paper, we formalized two special kinds of Diffie-Hellman assumptions, formalized an input-aware equivocable scheme and exhibits a protocol Π_{Gen} that provably implements the scheme under the aforementioned assumptions. These objects and proofs have been done along traditional lines, i.e., outside of a particular framework like Canetti's UC model.

We presented a UC (setup) functionality called $\mathcal{F}_{atomic}$ (which allows two parties to have a short, "fully isolated" exchange of *just one* message each). We gave the necessary proofs to show that a slight modification of our protocol Π_{Gen} UC-realizes commitments. This is possible without the need of a PKI, i.e., with the mere separation of an initialization phase (using just 2 atomic exchanges) and allows the two parties involved to establish two private, public key-pairs.

Finally, we also herein discussed the relevance and efficiency of our protocol, on a stand-alone basis as well as a protocol realizing other primitives, e.g., ZK.

References

1. Backes, M., Pfitzmann, B., Waidner, M.: A general composition theorem for secure reactive systems. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 336–354. Springer, Heidelberg (2004)
2. Barak, B., Canetti, R., Nielsen, J.B., Pass, R.: Universally composable protocols with relaxed set-up assumptions. In: Proc. of the 45th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2004, pp. 186–195. IEEE Computer Society, Washington, DC (2004)

3. Beaver, D.: Foundations of secure interactive computing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 377–391. Springer, Heidelberg (1992)
4. Blum, M.: How to prove a theorem so no one else can claim it. In: An Address to the Int. Congress of Mathematicians (August 1986)
5. Boneh, D.: The Decision Diffie-Hellman Problem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 48–63. Springer, Heidelberg (1998)
6. Brands, S., Chaum, D.: Distance-Bounding Protocols (Extended Abstract). In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 344–359. Springer, Heidelberg (1994)
7. Canetti, R.: A Unified Framework for Analyzing Security of Protocols. In: Electronic Colloquium on Computational Complexity (ECCC), vol. 8(16) (2001)
8. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067 (2005), http://eprint.iacr.org/
9. Canetti, R., Dakdouk, R.R.: Towards a theory of extractable functions. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 595–613. Springer, Heidelberg (2009)
10. Canetti, R., Fischlin, M.: Universally Composable Commitments. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 19–41. Springer, Heidelberg (2001)
11. Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally Composable Two-Party and Multi-Party Secure Computation. In: The 34th Annual ACM Symposium on Theory of Computing (STOC 2002), pp. 494–503 (2002)
12. Chandran, N., Goyal, V., Sahai, A.: New Constructions for UC Secure Computation Using Tamper-Proof Hardware. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 545–562. Springer, Heidelberg (2008)
13. Cimato, S., Galdi, C., Persiano, G. (eds.): SCN 2002. LNCS, vol. 2576. Springer, Heidelberg (2003)
14. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
15. Damgård, I., Nielsen, J.B., Wichs, D.: Universally composable multiparty computation with partially isolated parties. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 315–331. Springer, Heidelberg (2009)
16. Damgård, I.B., Nielsen, J.B.: Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 581–596. Springer, Heidelberg (2002)
17. Dent, A.W. The hardness of the DHK problem in the generic group model (2006) a.dent@rhul.ac.uk13277 (received April 24, 2006), (last revised May 9, 2006)
18. Drimer, S., Murdoch, S.J.: Keep your enemies close: distance bounding against smartcard relay attacks. In: Proc. of 16th USENIX Security Symposium on USENIX Security Symposium, SS 2007, pp. 7:1–7:16. USENIX Association, Berkeley (2007)
19. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In: 27th Annual Symposium on Foundations of Computer Science, pp. 174–187 (October 1986)
20. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: Proc. of the Seventeenth Annual ACM Symposium on Theory of Computing, STOC 1985, pp. 291–304. ACM, New York (1985)
21. Herzog, J.C., Liskov, M., Micali, S.: Plaintext awareness via key registration. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 548–564. Springer, Heidelberg (2003)
22. Kalai, Y.T., Lindell, Y., Prabhakaran, M.: Concurrent general composition of secure protocols in the timing model. In: Proc. of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, STOC 2005, pp. 644–653. ACM, New York (2005)

23. Katz, J.: Universally Composable Multi-party Computation Using Tamper-Proof Hardware. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 115–128. Springer, Heidelberg (2007)
24. Liskov, M., Lysyanskaya, A., Micali, S., Reyzin, L., Smith, A.: Mutually independent commitments. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 385–401. Springer, Heidelberg (2001)
25. Mayes, K., Cobourne, S., Markantonakis, K.: Near field technology in challenging environments. In: Smart Card Technology Int., NFC and Contactless, pp. 65–69 (2011)
26. Micali, S., Rogaway, P.: Secure computation (abstract). In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 392–404. Springer, Heidelberg (1992)
27. Welzl, E., Montanari, U., Rolim, J.D.P. (eds.): ICALP 2000. LNCS, vol. 1853. Springer, Heidelberg (2000)
28. Moran, T., Segev, G.: David and Goliath Commitments: UC Computation for Asymmetric Parties Using Tamper-Proof Hardware. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 527–544. Springer, Heidelberg (2008)
29. Okamoto, T., Uchiyama, S.: A new public-key cryptosystem as secure as factoring. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 308–318. Springer, Heidelberg (1998)
30. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
31. Teranishi, I., Ogata, W.: Relationship between standard model plaintext awareness and message hiding. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. E91-A, 244–261 (2008)
32. Ventre, C., Visconti, I.: Message-aware commitment schemes (2008) (unpublished manuscript)

A "Unconventional" Commitments: A Comparison

The notion of input-aware commitments (IAC) was studied before under the name of extractable commitments [13]. This was carried mainly in the CRS model, in [27,13], as part of zero-knowledge proofs. Unlike the scheme to follow, these commitments did not contain an explicit notion of equivocability, in the standard lines, i.e., outside the UC framework. Thus, we sometimes refer to them as *IAC* (input-aware commitments) as opposed to *IAEC* (input-aware equivocable commitments).

Canetti *et al.* [9] applied known commitment-constructions from injective one-way functions and from pseudorandom generators to get extractable commitments (i.e., IAC) when the underlying primitives used are extractable. We dissociate ourselves from this method and rely instead on hardness assumptions[6].

In the above sense, we use a stronger knowledge guarantee, which brings us closer to an (unpublished) result by Ventre and Visconti [32] in which they construct extractable commitments (i.e., IAC) from plaintext-aware encryption schemes, using certain hardness assumptions. However, our construction is not from PAW encryption directly, yet it bears similar assumptions to such encryption schemes [14], but it is also equivocable, i.e., it is an IAEC.

[6] Extractable functions abstract away from specific e.g., number-theoretic assumptions like the knowledge of exponents and are cast in a complexity-theoretic setting.

Further, we mention that primitives similar to input-aware equivocable commitments have been explored before by Damgård and Nielsen (i.e., mixed commitments) inside the UC framework, UC-realizing an ideal functionality $\mathcal{F}_{HCOM}$ of homomorphic commitment [16] in the CRS-hybrid UC model. Here, the formalization is different, the protocol more specific, the scheme is initially cast upon traditional lines. We only eventually show that we UC-realize the normal, ideal functionality of commitment, i.e., not the homomorphic version, using not a CRS, but a different setup. Namely, we show that our specialized commitment protocol is UC-realizable in the UC hybrid model with the $\mathcal{F}_{atomic}$ setup. More precisely, we will show that the thus-wise realized protocol UC-emulates the ideal functionality of commitment $\mathcal{F}_{COM}$ (not $\mathcal{F}_{HCOM}$). The protocol from Damgård and Nielsen [16] is sometimes extractable, sometimes equivocable, but not both. This depends on what the simulator needs in UC-security. (See more technical details on page 134.) In the plain model, Damgård and Nielsen's commitment is therefore not extractable nor equivocable. This is essentially different from the protocol advanced herein. Indeed, one of the ideas in this paper also lies in introducing new techniques of extractability of the "real" committed bit by the ideal adversary. Our protocol enjoys both extractability and equivocability, at the same time, even outside of the UC framework.

When compared to constructions from Damgård *et al.* [15], one advantage of our input-aware equivocable commitment is that it integrates the secret key extraction and becomes feasible with $\mathcal{F}_{atomic}$ efficiently. (In [15], the entire prover protocol of a WI ℓ-PoK scheme must be run in isolation.)

Another notion to thwart relay attack in commitment protocols is the notion of mutually independent commitments [24].

B Proof of Th. 10

Proof (sketch). Given a real-world adversary $\mathcal{A}$ in the UC model with atomic-exchange setup, we construct a UC ideal adversary I as follows.

A. We first treat the case where only S is corrupted by $\mathcal{A}$ and it is denoted as S^*. I simulates S^*, $\mathcal{F}_{atomic}$ and R_{COM} internally, and I lets S^* interact with $\mathcal{Z}$ externally (so that $\mathcal{Z}$ cannot distinguish I's run from the real-world experiment).

The simulation by I together with $\mathcal{Z}$ defines an algorithm $\mathcal{B}$, which stops before $\mathcal{F}_{atomic}$ receives X_0 from R_{COM} (as per the games defining the DDH and DHK0 assumptions). The algorithm $\mathcal{B}$ defines ρ and state, the latter being the current view of S^*. Like before, in state, we restrict to the coins $\overline{r_{\mathcal{A}}}$ that S^* has used so far. Let the unused coins by S^* be denoted $\overline{\overline{r_{\mathcal{A}}}}$. The next step of the simulation defines from state the last algorithm that S^* would have sent to $\mathcal{F}_{atomic}$ such that $\mathcal{A}(\mathsf{state}, X_0; \overline{\overline{r_{\mathcal{A}}}})$ would produce (X, X'), using solely on the view of $\mathcal{A}$ since in fact X, X' should be the output m' of $\mathcal{F}_{atomic}$. By the assumptions we use, we now have another algorithm $\mathcal{E}(\mathsf{state}, X_0; \overline{\overline{r_{\mathcal{A}}}})$ that yields x such that $X = g^x$ or R_{COM} aborts[7]. Thus, our constructed I can simply run $\mathcal{E}(\mathsf{state}, X_0; \overline{\overline{r_{\mathcal{A}}}})$ by using the view of S^*. As I goes on in the simulation of R_{COM}, it can extract the committed bit b from (U, V) thanks to x and send this bit to $\mathcal{F}_{COM}$. As in Th. 9, we can show that the opening to $1 - b$ would contradict the assumptions.

B. When R is corrupted by $\mathcal{A}$, we denote it as R^*. The simulation works as follows. I simulates R^*, $\mathcal{F}_{atomic}$ and $S_{COM}(b_0)$ (for an arbitrary bit b_0) internally, and I lets R^*

[7] If M aborts in real life, we assume it outputs a special value such that the protocol itself finishes.

interact with $\mathcal{Z}$ externally (so that $\mathcal{Z}$ cannot distinguish I's run from the real-world experiment).

The simulation by I together with $\mathcal{Z}$ defines an algorithm $\mathcal{B}$, which runs until the moment before $\mathcal{F}_{atomic}$ receives Y_0 from S_{COM} and then $\mathcal{B}$ stops. As before, $\mathcal{B}$ will produce ρ and state as the current view of R^*, limiting his coins $r_{\mathcal{A}}$ to $\overline{r_{\mathcal{A}}}$, i.e., to those used so far, where $r_{\mathcal{A}} := \overline{r_{\mathcal{A}}}||\overline{\overline{r_{\mathcal{A}}}}$. Then the output (Y, Y') of $\mathcal{F}_{atomic}$ (on the algorithm sent to it by R^*) can be seen as the output of $\mathcal{A}$ with input state. Augmented with the message Y_0 and the coins $\overline{\overline{r_{\mathcal{A}}}}$, it defines $\mathcal{A}(\mathsf{state}, Y_0; \overline{\overline{r_{\mathcal{A}}}})$. Due to the $\mathsf{ag\text{-}DHK0}_{\mathsf{Gen}}$ assumption, there must exist some algorithm $\mathcal{E}$ such that, except for negligible cases, $\mathcal{E}(\mathsf{state}, Y_0; \overline{\overline{r_{\mathcal{A}}}})$ produces y satisfying $Y = g^y$, or S_{COM} rejects (Y, Y'). Note that as before, the pair (Y, Y') is produced by using solely on the view of R^* (since the message Y_0 is tagged as atomic). So, our constructed I can again simply run $\mathcal{E}(\mathsf{state}, Y_0; \overline{\overline{r_{\mathcal{A}}}})$. Then, the adversary I can either simulate S_{OPEN} (if $b{=}b_0$) or, otherwise, simulate Equiv using y.

The argument of the indistinguishability between the two worlds (the real one and the simulated one by I) follows the exact same arguments as those in the proof of Th. 9. □