Switch Commitments: A Safety Switch for Confidential Transactions

Tim Ruffing and Giulio Malavolta

CISPA, Saarland University
tim.ruffing@mmci.uni-saarland.de, malavolta@cs.uni-saarland.de

Abstract. Cryptographic agility is the ability to switch to larger cryptographic parameters or different algorithms in the case of security doubts. This very desirable property of cryptographic systems is inherently difficult to achieve in cryptocurrencies due to their permanent state in the blockchain: for example, if it turns out that the employed signature scheme is insecure, a switch to a different scheme can only protect the outputs of future transactions but cannot fix transaction outputs already recorded in the blockchain, exposing owners of the corresponding money to risk of theft. This situation is even worse with Confidential Transactions, a recent privacy-enhancing proposal to hide transacted monetary amounts in homomorphic commitments. If an attacker manages to break the computational binding property of a commitment, he can create money out of thin air, jeopardizing the security of the entire currency. The obvious solution is to use statistically or perfectly binding commitment schemes but they come with performance drawbacks due to the need for less efficient range proofs.
In this paper, our aim is to overcome this dilemma. We introduce *switch commitments*, which constitute a cryptographic middle ground between computationally binding and statistically binding commitments. The key property of this novel primitive is the possibility to switch existing commitments, e.g., recorded in the blockchain, from computational bindingness to statistical bindingness if doubts in the underlying hardness assumption arise. This switch trades off efficiency for security. We provide a practical and simple construction of switch commitments by proving that ElGamal commitments with a restricted message space are secure switch commitments. The combination of switch commitments and statistically sound range proofs yields an instantiation Confidential Transactions that can be switched to be resilient against post-quantum attacks.

1 Introduction

The security of Bitcoin relies on cryptographic hardness assumptions, e.g., the hardness of computing discrete logarithms on the `secp256k1` [3] elliptic curve. Advances in solving the discrete logarithm problem can lead to uncertainty about whether currently deployed key sizes or algorithms are still safe.

In this situation, the obvious step is to obsolete current parameters, and switch to larger parameters or even entirely different algorithms in the system.

Since Bitcoin relies on the hardness of the discrete logarithm problem for unforgeability of ECDSA signatures, this just ensures security of future transactions but cannot fix already performed transactions: the current unspent transaction outputs in the blockchain are still protected by the obsolete cryptographic parameters.

While this is a very unfortunate situation, because users' funds are at risk of theft, it is then the responsibility of users to spend these outputs to fresh addresses of their own, thereby creating new unspent outputs protected by new keys and possibly new cryptographic algorithms. (After this step, the attacker can still break old signing keys. However, then consensus will ensure that the old outputs are already spent and thus the signing keys are worthless.) To sum up, individual users may lose their money if they fail to perform this safety measure, but the security of the Bitcoin system as a whole is unaffected.

However, the situation will be much worse in a cryptocurrency with Confidential Transactions (CT) [9,6]. CT is a privacy-enhancing technology thas has been proposed as an extension to Bitcoin. The proposal is currently tested and evaluated in the Elements Alpha sidechain [4]; moreover, it has been successfully deployed in the cryptocurrency Monero [11].

The purpose of CT is to hide the monetary amounts in transactions by replacing plain amounts by commitments to the amounts. Since the commitment scheme used is additively homomorphic, the creator of a transaction can easily prove to the network that a transaction is *balanced*, i.e., the sum of its outputs is not more money than the sum of its inputs. The proof essentially opens the commitment to the homomorphic sum of the inputs minus the outputs to zero, which does not reveal the individual monetary amounts of the inputs and outputs in the transaction. To be sound, a non-interactive zero-knowledge proof is added to each commitment to show that the committed value is in a certain range. These so-called *range proofs* ensure that the computation of the sum does not overflow.

The current CT proposal relies on Pedersen commitments on an elliptic curve computed as $c = g^m h^r$, where m is the message, r is a random value, and g and h are public generators of the elliptic curve group. Pedersen commitments are only computationally binding under the assumption that computing discrete logarithms is hard. Thus, if an attacker manages to break one discrete logarithm with current parameters, the balance property of the currency breaks down with catastrophic consequences: Knowledge of $\log_g h$ enables the attacker to open each of his commitments, no matter what amount it is supposed to commit to, to an arbitrary amount of money. That is, the attacker can effectively create an arbitrary amount of money, limited only by the maximum amount of money that can be transferred in a transaction. Even worse, this attack will go unnoticed due to the hiding property of the commitments. As a consequence, if the attacker manages to compute a single discrete logarithm, not only is the individual security of funds threatened, but the entire currency is doomed.

As a consequence, the situation is much worse with CT than without CT, when there is doubt in the hardness of the selected parameters. With CT, the

only safe way out is to introduce new parameters or algorithms and force users to spend unspent transaction outputs using the obsolete parameters *before some hard deadline* T. After time T, such obsolete outputs will not be spendable anymore, i.e., the corresponding funds will expire, effectively destroying money. This is clearly highly undesirable and it is not clear at all if such a change will be accepted by miners.

2 Switch Commitments

The obvious way to overcome all of the aforementioned issues is to use a commitment scheme that is statistically binding, i.e., it is binding even for a computationally unrestricted attacker. For instance, just adding g^r turns a computationally binding Pedersen commitment into a statistically binding ElGamal commitment.[1]

However, this modification requires efficient range proofs particularly suited to the new commitment scheme and, as a consequence, precludes the use of the highly optimized range proofs [13,8] developed for Pedersen commitments.

Instead, we aim for a solution compatible with the efficient range proofs. Our tool to achieve this goal is a novel security notion between computational and statistical bindingness. We introduce *switch commitments*, which are commitments with a *partial* and a *full* verification algorithm and special binding properties as follows.

- The commitment is computationally binding when partially verified.
- The commitment is statistically binding when fully verified.
- The commitment is *everlastingly binding*. This novel property captures the essence of switch commitments. It states that if the commitment is created by a computationally bounded attacker, and can be opened to some message when partially verified, then later even a computationally unbounded attacker can open the commitment to a different message when fully verified.

These properties enable verifiers to use the commitment scheme in a computationally binding or a statistically binding way, depending on the verification algorithm used. In particular, everlasting bindingness ensures that it is possible to start with partial verification and then *switch* to full verification, even for already existing commitments, e.g., commitments stored in the blockchain.

We prove that an ElGamal commitment $(g^m h^r, g^r)$ with a message space of polynomial size is a homomorphic switch commitment where the partial verification algorithm ignores the element g^r and verifies only the Pedersen commitment $g^m h^r$. Since the message space of commitments used in CT is restricted to integers in a fixed range to avoid overflow anyway, this switch commitment scheme is an optimal choice if a trade-off between security and performance is desired.

[1] The ElGamal commitment is actually even perfectly binding. We stick to the more general statistical property in this work.

2.1 Usage in Confidential Transactions

A switch commitment scheme can be used in CT as follows: When performing a transaction now, the network relies only on the partial verification to ensure that the transaction is balanced, i.e., the transaction does not generate money out of thin air. In particular, creators of transactions are forced to prove that they can open the commitments to messages such that no money will be created and the partial verification algorithm accepts the openings. While this means that the balance property holds only computationally, it is sufficient to use range proofs that cover only partial verification, i.e., the creator of the commitment must only demonstrate that he can open the commitment to a value in range when the opening is partially verified. Applied to ElGamal commitments, this effectively means that it suffices for the range proof to cover only the first element, which is a Pedersen commitment. This is more efficient because the most efficient known range proofs systems rely on Pedersen commitments.

In the future, if there is serious doubt about the cryptographic strength of the used commitment scheme or its parameters, a soft-fork can require confidential transactions created after some time T to be fully verified. Then, creators of transactions are forced to prove that they can open commitments only to values such that no money will be created, and that the full verification algorithm will accept this opening. This means that further transactions are required to provide proofs of the balance property with respect to full verification. In other words, no attacker can spend an already existing output with more money than it is supposed to contain, even if this output was created by the attacker before T (when the attacker was assumed to be computationally bounded). These proofs of the balance property require range proofs, which are potentially less efficient than range proofs which only cover partial verification. As a result, this switch to different range proofs trades off efficiency for security.

Efficiency Comparison of Known Range Proof Systems. Assume we would like to prove that the committed value m is in the range $[0, b^n]$. We further assume that we rely on elliptic curves, so group and field elements are of roughly the same size. For Pedersen commitments, the smallest known range proof has been proposed by Back and Maxwell [13] and needs $bn+1$ elements. For ElGamal commitments, the smallest known range proof has been proposed by Andreev [1] and needs $(b+1)n+1$ elements. Consequently, range proofs for Pedersen commitments are more efficient.

Soundness of the Range Proofs. All discussed range proofs for ElGamal commitments are constructed using the Fiat-Shamir transform. They are sound *even if the attacker is able to compute discrete logarithms*, and a recent result of Unruh [15] shows that their soundness holds up in a post-quantum world. That is, an instantiation of CT using ElGamal commitments and one of the aforementioned range proof systems is secure against post-quantum attackers trying to break the balance property.

We note that, even though the soundness of the aforementioned proofs is unconditional in the random oracle model, the soundness only holds against

computationally bounded attackers due to the hash function in the Fiat-Shamir transform. This means that even the usage of switch commitments in CT can only protect against further advances in the discrete logarithm problem but not against a failure of the hash function used in the Fiat-Shamir transform. Consequently, larger parameters are necessary for the post-quantum soundness of the range proofs as compared to classical security.

Hidingness of the Commitments. Note that switch commitments can only be computationally hiding, so the privacy of individual commitments cannot be guaranteed if we assume that the underlying problem is not hard anymore. However, giving up privacy is arguably better than putting the security of the entire currency at risk.

Observe that a soft-fork is a possible way to perform the switch, but it is not strictly necessary. In the time until the soft-fork is deployed (or if the fork cannot be agreed upon), recipients could alternatively just force the new rules by refusing to accept payments via non-statistically secure outputs created after time T (and any of their child transaction outputs in the transaction graph), effectively rendering the funds worthless.

3 Preliminaries

In this section we introduce the notation and the cryptographic primitives that we will use throughout our work. We denote by $\lambda \in \mathbb{N}$ the security parameter and by $\mathsf{poly}(\lambda)$ any function that is bounded by a polynomial in λ. We denote any function that is *negligible* in the security parameter with $\mathsf{negl}(\lambda)$. We say that an algorithm is ppt if its running time is bounded by some function $\mathsf{poly}(\lambda)$. Given a set S, we denote by $x \leftarrow S$ that x is uniformly sampled from S.

3.1 Commitments

A commitment scheme [12] is a two-phase protocol between a sender and a receiver. In the first phase, the sender commits to a message m with a string com. In the second phase, the sender reveals the opening information op and the message m to the receiver, who can check whether com was indeed a valid commitment on m. All algorithms have access to a public random string crs generated by a trusted setup party.

A commitment scheme is *computationally hiding* if commitment itself does not reveal information about the message to a computationally bounded attacker.

Definition 1 (Computationally Hiding). *A commitment scheme with commitment algorithm* Commit *is* computationally hiding *if there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that for all* ppt *attackers* $\mathcal{A}$*, for a randomly sampled* $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$*, and for all pairs of messages* (m_0, m_1)*, we have that*

$$\Pr\left[\mathcal{A}(\mathsf{crs}, \mathsf{com}) = b \mid b \leftarrow \{0,1\}; \mathsf{com} \leftarrow \mathsf{Commit}(\mathsf{crs}, m_b)\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

A commitment scheme is *binding* if no sender is able to output openings $(\mathsf{op}, \mathsf{op}')$ for the same commitment com such that they open it to two different values. We consider binding against computationally bounded and unbounded attackers.

Definition 2 (Computationally and Statistically Binding). *A verification algorithm* Verify *is* computationally binding *if there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that for all* ppt *attackers* $\mathcal{A}$ *and for a randomly sampled* $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$*, we have that*

$$\mathsf{Pr}\left[\begin{array}{l}\mathsf{Verify}(\mathsf{crs},\mathsf{com},\mathsf{op},m)=1\\ \quad\wedge\ \mathsf{Verify}(\mathsf{crs},\mathsf{com},\mathsf{op}',m')=1\\ \quad\wedge\ m\neq m'\end{array}\middle|\ (\mathsf{com},\mathsf{op},m,\mathsf{op}',m')\leftarrow\mathcal{A}(\mathsf{crs})\right]\leq\mathsf{negl}(\lambda).$$

Statistical bindingness *is defined identically except that* $\mathcal{A}$ *is computationally unbounded.*

3.2 Hardness Assumptions

Here we formally describe the computational hardness assumptions that we need for the security of our construction. First, we introduce the discrete logarithm assumption.

Definition 3 (Discrete Logarithm Assumption). *Let* $\mathcal{G}$ *be a multiplicative cyclic group of order* p *proportional to the security parameter* λ *and let* g *be a generator of* $\mathcal{G}$*. We say that the* discrete logarithm *problem is hard if, for a random integer* $x \in \mathbb{Z}_p$ *and for all* ppt *attackers* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\mathsf{Pr}\left[\mathcal{A}(\mathcal{G}, g, g^x) = x\right] \leq \mathsf{negl}(\lambda).$$

Second, we formalize the computational Diffie-Hellman problem and the inverse computational Diffie-Hellman problem. These problems are known to be equivalent [2].

Definition 4 (Computational Diffie-Hellman Assumption). *Let* $\mathcal{G}$ *be a multiplicative cyclic group of order* p *proportional to the security parameter* λ *and let* g *be a generator of* $\mathcal{G}$*. We say that the* computational Diffie-Hellman *problem is hard if, for two random integers* $x, y \in \mathbb{Z}_p$ *and for all* ppt *attackers* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\mathsf{Pr}\left[\mathcal{A}(\mathcal{G}, g, g^x, g^y) = g^{xy}\right] \leq \mathsf{negl}(\lambda).$$

Definition 5 (Inverse Computational Diffie-Hellman Assumption). *Let* $\mathcal{G}$ *be a multiplicative cyclic group of order* p *proportional to the security parameter* λ *and let* g *be a generator of* $\mathcal{G}$*. We say that the* inverse computational Diffie-Hellman *problem is hard if, for a random integer* $x \in \mathbb{Z}_p$ *and for all* ppt *attackers* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\mathsf{Pr}\left[\mathcal{A}(\mathcal{G}, g, g^x) = g^{x^{-1}}\right] \leq \mathsf{negl}(\lambda),$$

where x^{-1} denotes the multiplicative inverse of x.

Finally, we formalize the decisional Diffie-Hellman problem.

Definition 6 (Decisional Diffie-Hellman Assumption). *Let $\mathcal{G}$ be a multiplicative cyclic group of order p proportional to the security parameter λ and let g be a generator of $\mathcal{G}$. We say that the* decisional Diffie-Hellman *problem is hard if, for three random integers $x, y, z \in \mathbb{Z}_p$ and for all* ppt *attackers $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that:*

$$\mathsf{Pr}\left[\mathcal{A}(\mathcal{G}, g, g^x, g^y, h) = b \mid b \leftarrow \{0,1\}; h = \begin{cases} g^{xy} & \text{if } b = 0 \\ g^z & \text{if } b = 1 \end{cases}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

4 Problem Description

The main ingredient of CT [9,6] is homomorphic Pedersen commitments [12]. Given a group $\mathcal{G}$ of prime order p, and two generators g and h, a Pedersen commitment on a message m consists of a single group element computed as $g^m h^r$, for some $r \in \mathbb{Z}_p$ chosen uniformly at random. The opening information is the tuple (m, r) and the verifier can check the validity of a given commitment by simply recomputing it. Commitments are homomorphic due to $g^m h^r \cdot g^{m'} h^{r'} = g^{m+m'} h^{r+r'}$. It is easy to see that the commitment scheme is information-theoretically hiding, and that it is computationally binding under the discrete logarithm assumption.

Loosely speaking, a confidential transaction contains a collection of commitments, whose messages add up to zero, and a publicly verifiable proof that this is the case, which is essentially just opening of the homomorphic sum commitment to zero. Additionally, each commitment comes with a range proof that demonstrates that the committed integer value lies within a certain range $[0, d]$, where d is some fixed value that determines the maximum number of currency units allowed in a single transaction output.[2] We remark that, for the specific case of Pedersen commitments, there exist efficient computationally sound range proofs based on borromean ring signatures [8] and optimizations [13].

4.1 Attacker Model

We consider an attacker whose goal is to break the binding property of a commitment by computing a commitment c over a certain value m for a confidential transaction and later on perform a transaction opening c to some $m' \neq m$ (or just proving that he knows how to open c to $m' \neq m$). Clearly this implies that the attacker was able to create money if $m' > m$.

[2] In fact, the value supported by CT is expressed by a floating point number, with the exponent being public and only the mantissa hidden in the commitment [9,6]. We ignore the public exponent in our description, because it does not affect our treatment. The valid range of values for the mantissa is $[0, 2^{32} - 1]$, i.e., $d = 2^{32} - 1$ satoshis (currency units).

If we consider an attacker that is computationally bounded at the time of the generation of a commitment, but later on *unbounded*, then it is easy to see that the current implementation of confidential transactions is no longer secure: An attacker could honestly compute a commitment to some small value m as $c = g^m h^r$ and then later on open it to any value $m' > m$ by computing $x = \log_g h$ and $r' = (m - m')/x + r$. It is easy to see that (m', r') is a valid opening for c.

Such a scenario may appear artificial at first glance, but one must consider that system parameters are chosen based on an estimation of the progress of the field, and therefore it is possible that unexpected developments of algorithms or new technologies render current choices for key lengths obsolete. Among others, the advent of quantum computers would imply an immediate breakdown of all systems based on discrete logarithm-related assumptions. Therefore we believe that considering an attacker that is computationally bounded only *during* the execution of the protocol constitutes a problem of practical relevance. We note that a similar model has already been considered for privacy properties in the context of electronic voting [10], multi-party computation [14], and encryption in the bounded storage model [7].

4.2 Switch Commitments

Here we extend the notion of a commitment scheme to support the *switching* functionality and we formally introduce the security definitions for our primitive.

Definition 7 (Switch Commitment Scheme). *A* switch commitment scheme $(\mathsf{Commit}, \mathsf{Verify}^{\mathsf{part}}, \mathsf{Verify}^{\mathsf{full}})$ *consists of four* ppt *algorithms as follows:*

- $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$*: Given the security parameter* λ*, the setup algorithm* Setup *outputs a public random string* crs*.*
- $(\mathsf{com}, \mathsf{op}) \leftarrow \mathsf{Commit}(\mathsf{crs}, m)$*: Given the public random string* crs*, and a message* m*, the commitment algorithm* Commit *outputs a commitment* com *and opening information* op*.*
- $b \leftarrow \mathsf{Verify}^{\mathsf{part}}(\mathsf{crs}, \mathsf{com}, \mathsf{op}, m)$*: Given the public random string* crs*, a message* m*, a commitment* com *and opening information* op*, the partial verification algorithm* $\mathsf{Verify}^{\mathsf{part}}$ *outputs* 1 *iff* op *is a valid partial opening for commitment* com *on message* m*.*
- $b \leftarrow \mathsf{Verify}^{\mathsf{full}}(\mathsf{crs}, \mathsf{com}, \mathsf{op}, m)$*: Given the public random string* crs*, a message* m*, a commitment* com *and opening information* op*, the full verification algorithm* $\mathsf{Verify}^{\mathsf{full}}$ *outputs* 1 *iff* op *is a valid full opening for commitment* com *on message* m*.*

A switch commitment essentially defines two commitment schemes, namely a scheme with the partial verification algorithm and a scheme with the full verification algorithm. We require that both schemes fulfill standard security notions.

Definition 8 (Standard Security Properties). *For security of a switch commitment scheme* $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify}^{\mathsf{part}}, \mathsf{Verify}^{\mathsf{full}})$*, we require that*

– *the commitment algorithm* Commit *is computationally hiding,*
– *the verification algorithm* $\mathsf{Verify}^{\mathsf{part}}$ *is computationally binding, and*
– *the verification algorithm* $\mathsf{Verify}^{\mathsf{full}}$ *is statistically binding.*

Following our the attacker model as described in Section 4.1, we further require that even an unbounded attacker cannot open an old commitment (from the time when the attacker was still bounded) to a different message than it was created for. The novel security property is crucial for the intended application. In the following we formally define the notion of *everlasting bindingness* for a switch commitment scheme.

Definition 9 (Everlastingly Binding). *A switch commitment scheme* $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify}^{\mathsf{part}}, \mathsf{Verify}^{\mathsf{full}})$ *is* everlastingly binding *if there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that for all attackers* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$*, where* $\mathcal{A}_0$ *is* ppt *(and* $\mathcal{A}_1$ *is not computationally bounded), and for a randomly sampled* $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$*, we have that*

$$\mathsf{Pr}\left[\begin{array}{l} \mathsf{Verify}^{\mathsf{part}}(\mathsf{crs}, \mathsf{com}, \mathsf{op}, m) = 1 \\ \quad \wedge\ \mathsf{Verify}^{\mathsf{full}}(\mathsf{crs}, \mathsf{com}, \mathsf{op}', m') = 1 \\ \quad \wedge\ m \neq m' \\ \hline (\mathsf{com}, m, \mathsf{op}, \mathit{state}) \leftarrow \mathcal{A}_0(\mathsf{crs}); \\ (m', \mathsf{op}') \leftarrow \mathcal{A}_1(\mathsf{crs}, \mathit{state}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

5 Construction

In the following we describe our construction for a switch commitment scheme with efficient range proof. Our scheme is essentially a combination of a Pedersen and ElGamal commitment scheme with restricted message space. The commitment algorithm outputs an ElGamal commitment $(g^x h^r, g^r)$ and the full verification algorithm recomputes the commitment to verify it. However, the partial verification algorithm verifies only the Pedersen commitment $g^x h^r$. This makes it possible to use efficient range proofs optimized for Pedersen commitments.

It is crucial for the security of our construction that the message space is restricted to a size polynomial in the security parameter and the verification algorithm rejects messages not in the space. In the proof of everlasting bindingness, the reduction guesses a message in a commitment, and thus the reduction incurs a loss proportional to the size of the message space. Slightly increased parameters are necessary to compensate for this loss of security.

Note that that the message space of the commitments used in CT is already is restricted to integers in the range $[0, d]$ for a fixed non-negative integer d that is a parameter of the system and determines the maximum value of a transaction.

With the application in CT in mind, we describe the scheme for concreteness with this message space. We however stress that any other restriction of the message space is possible, as long as the message space has polynomial size in the security parameter.

- $\mathsf{Setup}(1^\lambda, d)$: Initialize a multiplicative cyclic group $\mathcal{G}$ of order p, for some prime p of size proportional to λ. Sample random g and h in $\mathcal{G}$ and output $\mathsf{crs} = (\mathcal{G}, g, h, d)$.
- $\mathsf{Commit}(\mathsf{crs}, m)$: Parse crs as $(\mathcal{G}, g, h, d)$ and sample $r \in \mathbb{Z}_p$. Return $\mathsf{com} = (g^m h^r, g^r)$, and $\mathsf{op} = r$.
- $\mathsf{Verify}^{\mathsf{part}}(\mathsf{crs}, \mathsf{com}, \mathsf{op}, m)$: Parse crs as $(\mathcal{G}, g, h, d)$, com as (c, ℓ), and op as r. If $c = g^m h^r$ and $m \leq d$, then return 1. Return 0 otherwise.
- $\mathsf{Verify}^{\mathsf{full}}(\mathsf{crs}, \mathsf{com}, \mathsf{op}, m)$: Parse crs as $(\mathcal{G}, g, h, d)$, com as (c, ℓ), and op as r. If $c = g^m h^r$, $\ell = g^r$, and $m \leq d$, then return 1. Return 0 otherwise.

Avoiding Trusted Setup. We have chosen a description in the standard model to stress that the construction does not require random oracles. However, it is possible to avoid a trusted setup in the random oracle model by setting $h = H(g)$, for a hash function H. This is essentially what has been proposed in the draft of CT.

Homomorphic Property. Since the commitment algorithm is identical to the one of ElGamal commitments, the commitments are homomorphic due to $g^m h^r \cdot g^{m'} h^{r'} = g^{m+m'} h^{r+r'}$ and $g^r \cdot g^{r'} = g^{r+r'}$.

5.1 Security Analysis

Here, we formally argue about the security of the construction described above.

Claim 1 (Standard Security Properties) *The construction fulfills the standard security properties. In particular, commitments are computationally hiding under the decisional Diffie-Hellman assumption, the commitment scheme with the partial verification algorithm is computationally binding under the discrete logarithm assumption, and the scheme with the full verification algorithm is statistically binding.*

Proof. The construction is computationally hiding under the decisional Diffie-Hellman assumption, because the commitment algorithm is identical to the one for ElGamal commitments. For binding, recall that ElGamal commitments are perfectly (and thus statistically) binding, and that Pedersen commitments are computationally binding under the discrete logarithm assumption. We refer the reader to ElGamal [5] and Pedersen [12] for detailed discussions. □

Theorem 2 (Everlastingly Binding). *The construction is everlastingly binding under the computational Diffie-Hellman assumption.*

Proof. We prove that the construction is everlastingly binding under the inverse computational Diffie-Hellman assumption, which is known to be equivalent to the (standard) computational Diffie-Hellman assumption [2]. Assume towards

contradiction that there exists an attacker $(\mathcal{A}_0, \mathcal{A}_1)$ such that $\mathcal{A}_0$ is ppt and

$$\mathsf{Pr}\left[\begin{array}{l} \mathsf{Verify}^{\mathsf{part}}(\mathsf{crs}, \mathsf{com}, \mathsf{op}, m) = 1 \\ \quad \wedge\ \mathsf{Verify}^{\mathsf{full}}(\mathsf{crs}, \mathsf{com}, \mathsf{op}', m') = 1 \\ \quad \wedge\ m \neq m' \\ \hline (\mathsf{com}, m, \mathsf{op}, state) \leftarrow \mathcal{A}_0(\mathsf{crs}); \\ (m', \mathsf{op}') \leftarrow \mathcal{A}_1(\mathsf{crs}, state) \end{array}\right] \geq \epsilon(\lambda).$$

for some non-negligible function $\epsilon(\lambda)$. We construct the following reduction $\mathcal{R}$ against the inverse computational Diffie-Hellman assumption.

$\mathcal{R}(1^\lambda, \mathcal{G}, g, h)$: On input the group description $\mathcal{G}$, the generator g and a random element h, the reduction sets $\mathsf{crs} = (\mathcal{G}, g, h, d)$ for a fixed d. Then it runs $\mathcal{A}_0$ in input crs, which outputs $(\mathsf{com} = (c, \ell), m = w, \mathsf{op} = v)$, at some point of the execution. Finally, the reduction samples a random $d' \leq d$ and returns to the challenger

$$I = \left(\frac{\ell}{g^v}\right)^{(w-d')^{-1}}.$$

The reduction is efficient since it only executes $\mathcal{A}_0$, which is ppt; note that the reduction never executes $\mathcal{A}_1$. Let us denote w as m^{part} and v as r^{part}. By assumption, $\mathcal{A}_1$ will be able to open the commitment to some value m^{full} such that $m^{\mathsf{full}} \neq m^{\mathsf{part}}$ and $m^{\mathsf{full}} \leq d$ with probability at least $\epsilon(\lambda)$. Assume that the reduction guesses such a value m^{full} when selecting d' (note that this happens with probability at least $1/d$); then we have that $d' = m^{\mathsf{full}}$. Now we observe that

$$I = \left(\frac{\ell}{g^v}\right)^{(w-d')^{-1}} = \left(\frac{g^{r^{\mathsf{full}}}}{g^{r^{\mathsf{part}}}}\right)^{(m^{\mathsf{part}}-m^{\mathsf{full}})^{-1}}.$$

Since $g^{m^{\mathsf{full}}} h^{r^{\mathsf{full}}} = c$ (by the winning conditions of the game) or equivalently

$$g^{r^{\mathsf{full}}} = \left(\frac{c}{g^{m^{\mathsf{full}}}}\right)^{x^{-1}},$$

we have

$$I = \left(\frac{\left(\frac{c}{g^{m^{\mathsf{full}}}}\right)^{x^{-1}}}{g^{r^{\mathsf{part}}}}\right)^{(m^{\mathsf{part}}-m^{\mathsf{full}})^{-1}}.$$

Since also $g^{m^{\mathsf{part}}} h^{r^{\mathsf{part}}} = c$, it holds that

$$I = \left(\frac{\left(\frac{g^{m^{\mathsf{part}}} h^{r^{\mathsf{part}}}}{g^{m^{\mathsf{full}}}}\right)^{x^{-1}}}{g^{r^{\mathsf{part}}}}\right)^{(m^{\mathsf{part}}-m^{\mathsf{full}})^{-1}} = \left(\frac{\left(\frac{g^{m^{\mathsf{part}}+xr^{\mathsf{part}}}}{g^{m^{\mathsf{full}}}}\right)^{x^{-1}}}{g^{r^{\mathsf{part}}}}\right)^{(m^{\mathsf{part}}-m^{\mathsf{full}})^{-1}}$$

$$= \left(\frac{g^{(m^{\mathsf{part}}-m^{\mathsf{full}})x^{-1}+r^{\mathsf{part}}}}{g^{r^{\mathsf{part}}}}\right)^{(m^{\mathsf{part}}-m^{\mathsf{full}})^{-1}} = g^{x^{-1}}.$$

As argued above, this happens with probability at least $\frac{\epsilon(\lambda)}{d}$, which is non-negligible. This represents a contradiction to the computational inverse Diffie-Hellman assumption and concludes the proof. □

Acknowledgements. We thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the German Ministry for Education and Research (BMBF) through funding for the Center for IT-Security, Privacy and Accountability (CISPA) and the German Universities Excellence Initiative.

References

1. Andreev, O.: Confidential assets (2017), https://github.com/chain/chain/blob/confidential-spec/docs/protocol/specifications/ca.md#value-range-proof. Archived at http://www.webcitation.org/6qUEe3dKc
2. Bao, F., Deng, R.H., Zhu, H.: Variations of Diffie-Hellman problem. In: ICICS'03. Springer
3. Certicom Research: Sec 1: Elliptic curve cryptography, http://www.secg.org/download/aid-780/sec1-v2.pdf
4. Elements Project: Alpha sidechain, https://www.elementsproject.org/sidechains/alpha/
5. Elgamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: CRYPTO'84. Springer
6. Gibson, A.: An investigation into confidential transactions (2016), http://diyhpl.us/~bryan/papers2/bitcoin/An%20investigation%20into%20Confidential%20Transactions%20-%20Adam%20Gibson%20-%202016.pdf. Archived at http://www.webcitation.org/6qUF8XYmP
7. Harnik, D., Naor, M.: On everlasting security in the hybrid bounded storage model. In: ICALP'06
8. Maxwell, G., Poelstra, A.: Borromean ring signatures (2015), https://github.com/Blockstream/borromean_paper/raw/master/borromean_draft_0.01_9ade1e49.pdf. Archived at http://www.webcitation.org/6qUFVS2Ux
9. Maxwell, G.: Confidential transactions (2015), https://people.xiph.org/~greg/confidential_values.txt. Archived at http://www.webcitation.org/6qUFGwJah
10. Moran, T., Naor, M.: Receipt-free universally-verifiable voting with everlasting privacy. In: CRYPTO'06
11. Noether, S., Mackenzie, A.: Ring confidential transactions. Ledger (2016), http://www.ledgerjournal.org/ojs/index.php/ledger/article/view/34
12. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: CRYPTO'91. Springer
13. Poelstra, A., Back, A., Friedenbach, M., Maxwell, G., Wuille, P.: Confidential assets. In: BITCOIN'17. Springer
14. Unruh, D.: Everlasting multi-party computation. In: CRYPTO'13. Springer
15. Unruh, D.: Post-quantum security of fiat-shamir. Cryptology ePrint Archive, Report 2017/398 (2017), https://eprint.iacr.org/2017/398