Homomorphic Trapdoor Commitments to Group Elements

Jens Groth

University College London
j.groth@ucl.ac.uk

Abstract

We present a homomorphic trapdoor commitment to group elements. In contrast, previous homomorphic trapdoor commitment schemes only allow the messages to be exponents. Our commitment scheme is length-reducing, we can make a short commitment to many group elements at once, and it is perfectly hiding and computationally binding. The construction is based on groups with a bilinear map and the binding property follows from the simultaneous triple pairing assumption. While the simultaneous triple pairing assumption is new, we demonstrate that it is implied by the well-known decision linear assumption.

Keywords: Homomorphic trapdoor commitment, bilinear groups, simultaneous triple pairing assumption.

1 Introduction

A non-interactive commitment scheme makes it possible to create a *commitment* c to a secret message m. The commitment *hides* the message, but we may later disclose m and demonstrate that c was a commitment to m by revealing the randomness r used when creating it. Revealing the message and the randomness is called *opening* the commitment. It is essential that once a commitment is made, it is *binding*. Binding means that it is infeasible to find two openings of the same commitment to two different messages.

In this paper, we are interested in public-key commitments with a couple of useful features: First, we want the commitment scheme to have a *trapdoor* property. In normal operation the commitment scheme is binding, however, if we know a secret trapdoor tk associated with the public commitment key ck, then it is possible to create commitments that can be opened to any message. We note that the trapdoor property implies that the commitment hides the message. Second, we want the commitment scheme to be *homomorphic*. Homomorphic means that messages and commitments belong to abelian groups and if we multiply two commitments, we get a new commitment that contains the product of the two messages.

RELATED WORK. There are many examples of homomorphic commitments. Homomorphic cryptosystems such as ElGamal [ElG85], Okamoto-Uchiyama [OU98], Paillier [Pai99], BGN [BGN05] or Linear Encryption [BBS04] can be seen as homomorphic commitment schemes that are perfectly binding and computationally hiding. Commitments based on homomorphic encryption can be converted into computationally binding and perfectly hiding homomorphic commitments, see for instance the mixed commitments of Damgård and Nielsen [DN02] and the commitment schemes used by Groth, Ostrovsky and Sahai [GOS06], Boyen and Waters [BW06], Groth [Gro06] and Groth and Sahai

[GS08]. Even for the perfectly hiding variation of these commitment schemes, the size of a commitment is larger than the size of a message though. This length-increase follows from the fact that the underlying building block is a cryptosystem and a ciphertext must be large enough to accomodate the message.

There are also direct constructions of homomorphic trapdoor commitment schemes such as Guillou and Quisquater commitments [GQ88] and Pedersen commitments [Ped91]. Pedersen commitments are one of the most used commitment schemes in the field of cryptography. The public key consists of two group elements g, h belonging to a group of prime order q and we commit to a message $m \in \mathbb{Z}_q$ by computing $c = g^m h^r$, where $r \in \mathbb{Z}_q$ is a randomly chosen randomizer. Pedersen commitments are perfectly hiding with a trapdoor and if the discrete logarithm problem is hard they are computationally binding. There are many variants of the Pedersen commitment scheme. Fujisaki and Okamoto [FO97] and Damgård and Fujisaki [DF02] for instance suggest a variant where the messages can be arbitrary integers.

There is an important generalization of the Pedersen commitment scheme that makes it possible to commit to many messages at once. The public key consists of $n + 1$ group elements $g_1, \ldots, g_n, h$ and we compute a commitment to $(m_1, \ldots, m_n)$ as $c = h^r \prod_{i=1}^{n} g_i^{m_i}$. This commitment scheme is length-reducing since we only use one group element to commit to n messages, a feature that has been found useful in contexts such as mix-nets/voting, digital credentials, blind signatures and zero-knowledge proofs [FS01, Nef01, Bra00, KZ06, Lip03].

Common for all the homomorphic trapdoor commitment schemes[1] we mentioned above is that they are homomorphic with respect to *addition* in a ring or a field.However, in public-key cryptography it is common to work over groups that are not rings or fields and often it is useful to commit to group elements from such groups. Of course, if we know the discrete logarithms of the group elements we want to commit to, we can use the Pedersen commitment scheme to commit to the discrete logarithms. In general, we cannot expect to know the discrete logarithms of the group elements that we want to commit to though, leaving us with the open problem of constructing homomorphic trapdoor commitments to group elements.

OUR CONTRIBUTION. The contribution of this paper is the construction of a homomorphic trapdoor commitment scheme for group elements. The commitment scheme is perfectly hiding, perfectly trapdoor and computationally binding. We stresss that we can commit to arbitrary group elements and trapdoor-open to arbitrary group elements, even if we do not know the discrete logarithms of these group elements. Moreover, the commitment scheme has the additional advantage of being length-reducing, we can commit to multiple group elements with one short commitment.

Our construction is based on bilinear groups. These are groups G, G_T with a bilinear map $e : G \times G \to G_T$. Both messages and randomizers will be elements from G, whereas the commitments will consist of a couple of group elements in G_T. One advantage of our commitment scheme is that the construction is very simple. The public key consists of $2n + 4$ group elements $(g_1, h_1, \ldots, g_n, h_n, g_r, h_r, g_s, h_s)$ and we commit to $m_1, \ldots, m_n$ by choosing $r, s \in G$ at random and computing the commitment (c, d) as

$$c = e(g_r, r)e(g_s, s) \prod_{i=1}^{n} e(g_i, m_i) \quad \text{and} \quad d = e(h_r, r)e(h_s, s) \prod_{i=1}^{n} e(h_i, m_i).$$

[1] Boyen and Waters [BW06], Groth [Gro06] and Groth and Sahai [GS08] use homomorphic commitments to group elements, but do they do not have a *trapdoor* property that makes it possible to open them to arbitrary group elements. Moreover, those commitments suffer from being length-increasing.

The commitment scheme is computationally binding assuming the simultaneous triple pairing assumption holds. This assumption says that given two random triples (g_r, g_s, g_t) and (h_r, h_s, h_t) it is computationally infeasible to find group elements r, s, t so

$$e(g_r, r)e(g_s, s)e(g_t, t) = 1 \quad \text{and} \quad e(h_r, r)e(h_s, s)e(h_t, t) = 1.$$

Perhaps surprisingly, we show that the decision linear assumption [BBS04] implies the simultaneous triple pairing assumption.

2 Definitions

NOTATION. Algorithms in our commitment scheme take a security parameter k as input written in unary. For simplicity we will sometimes omit writing the security parameter explicitly, assuming k can be deduced from the other inputs. All our algorithms will be probabilistic polynomial time algorithms. We write $y = A(x; r)$, when A on input x and randomness r outputs y. We write $y \leftarrow A(x)$, for the process of picking randomness r at random and setting $y = A(x; r)$. We also write $y \leftarrow S$ for sampling y uniformly at random from the set S. When defining security, we assume that there is an adversary attacking our scheme. This adversary is modeled as a non-uniform polynomial time stateful algorithm. By stateful, we mean that we do not need to give it the same input twice, it remembers from the last invocation what its state was. This makes the notation a little simpler, since we do not need to explicitly write out the transfer of state from one invocation to the next. Given two functions $f, g : \mathbb{N} \rightarrow [0; 1]$ we write $f(k) \approx g(k)$ when there is negligible difference, *i.e.*, $|f(k) - g(k)| = k^{-\omega(1)}$.

2.1 Commitments

A commitment scheme is a protocol between Alice and Bob that allows Alice to commit to a secret message m. Later Alice may open the commitment and reveal to Bob that she committed to m. Commitment schemes must be binding and hiding. Binding means that Alice cannot change her mind, a commitment can only be opened to one message m. Hiding means that Bob does not learn which message Alice committed to.

In this paper, we will focus on non-interactive commitment schemes. In a non-interactive commitment scheme, Alice computes the commitment herself and sends it to Bob. The opening process is also non-interactive, it simply consists of Alice sending the message and the randomness she used when creating the commitment to Bob. Bob can now run the commitment protocol himself, to check that indeed this was the message Alice had committed to.

A non-interactive commitment scheme consists of three polynomial time algorithms $(\mathcal{G}, K, \mathrm{com})$. $\mathcal{G}$ is a probabilistic setup algorithm that takes as input the security parameter k and outputs some setup information gk. The setup information gk can for instance describe a finite group over which we are working, but it could also just be the security parameter written in unary so there is no loss of generality in including a setup algorithm. We include an explicit algorithm for the setup because when designing cryptographic protocols we often need the commitment scheme to work with an existing finite group. K is a probabilistic algorithm that takes as input the setup gk and generates a commitment key ck and a trapdoor key tk. The commitment key ck specifies a message space $\mathcal{M}_{ck}$, a randomizer space $\mathcal{R}_{ck}$ and a commitment space $\mathcal{C}_{ck}$. We assume it is easy to verify membership of the message space, randomizer space and the commitment space and it is possible to sample randomizers uniformly at random from $\mathcal{R}_{ck}$. The algorithm com takes as input the commitment key ck, a message m from the message space, a randomizer r from the randomizer space and outputs a commitment c in the commitment space.

We are interested in constructing homomorphic trapdoor commitments. By homomorphic, we mean that $\mathcal{M}_{ck}, \mathcal{R}_{ck}, \mathcal{C}_{ck}$ are groups with the property that if we multiply two commitments, then we get a commitment to the product of the messages. By trapdoor we mean that given the secret trapdoor key generated by the key generator, it is possible to open a commitment to any message. For this purpose, we have two additional probabilistic polynomial time algorithms Tcom and Topen. Tcom takes the trapdoor tk as input and outputs an equivocal commitment c and an equivocation key ek. Topen on input ek, c and a message $m \in \mathcal{M}_{ck}$ creates an opening $r \in \mathcal{R}_{ck}$ of the commitment, so $c = \mathrm{com}_{ck}(m; r)$.

Definition 1 (Homomorphic trapdoor commitment scheme) *A homomorphic trapdoor commitment scheme consists of a quintuple of algorithms* $(\mathcal{G}, K, \mathrm{com}, \mathrm{Tcom}, \mathrm{Topen})$ *as described above, such that* $(\mathcal{G}, K, \mathrm{com})$ *is hiding and binding and homomorphic and* $(\mathcal{G}, K, \mathrm{com}, \mathrm{Tcom}, \mathrm{Topen})$ *has a perfect trapdoor property as defined below.*

Definition 2 (Perfect hiding) *The triple* $(\mathcal{G}, K, \mathrm{com})$ *is perfectly hiding if for all stateful adversaries* $\mathcal{A}$ *we have*

$$\begin{aligned} & \Pr\Big[gk \leftarrow \mathcal{G}(1^k); (ck, tk) \leftarrow K(gk); (m_0, m_1) \leftarrow \mathcal{A}(gk, ck); c \leftarrow \mathrm{com}_{ck}(m_0) : \mathcal{A}(c) = 1\Big] \\ = \ & \Pr\Big[gk \leftarrow \mathcal{G}(1^k); (ck, tk) \leftarrow K(gk); (m_0, m_1) \leftarrow \mathcal{A}(gk, ck); c \leftarrow \mathrm{com}_{ck}(m_1) : \mathcal{A}(c) = 1\Big], \end{aligned}$$

where we require that $\mathcal{A}$ *outputs* m_0, m_1 *that belong to* $\mathcal{M}_{ck}$.

Definition 3 (Computational binding) *The triple* $(\mathcal{G}, K, \mathrm{com})$ *is computationally binding if for all non-uniform polynomial time stateful adversaries* $\mathcal{A}$ *we have*

$$\begin{aligned} \Pr\Big[&gk \leftarrow \mathcal{G}(1^k); (ck, tk) \leftarrow K(gk); (m_0, m_1, r_0, r_1) \leftarrow \mathcal{A}(gk, ck) : \\ & m_0 \neq m_1 \quad \wedge \quad \mathrm{com}_{ck}(m_0; r_0) = \mathrm{com}_{ck}(m_1; r_1)\Big] \approx 0, \end{aligned}$$

where we require that $\mathcal{A}$ *outputs* $m_0, m_1 \in \mathcal{M}_{ck}$ *and* $r_0, r_1 \in \mathcal{R}_{ck}$.

Definition 4 (Perfect trapdoor) *The quintuple* $(\mathcal{G}, K, \mathrm{com}, \mathrm{Tcom}, \mathrm{Topen})$ *has a perfect trapdoor property if for all stateful adversaries* $\mathcal{A}$ *we have*

$$\begin{aligned} & \Pr\Big[gk \leftarrow \mathcal{G}(1^k); (ck, tk) \leftarrow K(gk); m \leftarrow \mathcal{A}(gk, ck); r \leftarrow \mathcal{R}_{ck}; c = \mathrm{com}_{ck}(m; r) : \mathcal{A}(c, r) = 1\Big] \\ = \ & \Pr\Big[gk \leftarrow \mathcal{G}(1^k); (ck, tk) \leftarrow K(gk); m \leftarrow \mathcal{A}(gk, ck); (c, ek) \leftarrow \mathrm{Tcom}_{ck}(tk); \\ & \qquad r \leftarrow \mathrm{Topen}_{ek}(m, r) : \mathcal{A}(c, r) = 1\Big], \end{aligned}$$

where $\mathcal{A}$ *outputs* $m \in \mathcal{M}_{ck}$.

We note that the perfect trapdoor property implies that the commitment scheme is perfectly hiding, since a commitment is perfectly indistinguishable from an equivocal commitment that can be opened to any message.

Definition 5 (Homomorphic) *The commitment scheme* $(\mathcal{G}, K, \mathrm{com})$ *is homomorphic if* K *always outputs ck describing groups* $\mathcal{M}_{ck}, \mathcal{R}_{ck}, \mathcal{C}_{ck}$*, which we will write multiplicatively, such that for all* $m, m' \in \mathcal{M}_{ck}, r, r' \in \mathcal{C}_{ck}$ *we have*

$$\mathrm{com}_{ck}(m; r)\mathrm{com}_{ck}(m; r') = \mathrm{com}_{ck}(mm'; rr').$$

3 Setup

BILINEAR GROUPS. Let $\mathcal{G}$ be a probabilistic polynomial time algorithm that generates $(p, G, G_T, e, g) \leftarrow \mathcal{G}(1^k)$ such that

- p is a k-bit prime
- G, G_T are cyclic groups of order p
- g is a randomly chosen generator of G
- $e : G \times G \rightarrow G_T$ is a non-degenerate bilinear map so
  - $e(g, g)$ generates G_T
  - For all $a, b \in \mathbb{Z}_p$ we have $e(g^a, g^b) = e(g, g)^{ab}$
- Group operations, evaluation of the bilinear map, and membership of G, G_T are all efficiently computable.

SIMULTANEOUS TRIPLE PAIRING ASSUMPTION. The security of our commitment scheme will be based on the simultaneous triple pairing assumption. The simultaneous triple pairing problem is given random elements $g_r, h_r, g_s, h_s, g_t, h_t \in G$ to find a non-trivial triple $(r, s, t) \in G^3$ such that $e(g_r, r)e(g_s, s)e(g_t, t) = 1$ and $e(h_r, r)e(h_s, s)e(h_t, t) = 1$.

Definition 6 (Simultaneous triple pairing assumption) *We say the simultaneous triple pairing assumption holds for the bilinear group generator $\mathcal{G}$ if for all non-uniform polynomial time adversaries $\mathcal{A}$ we have*

$$\begin{aligned}\Pr\Big[&gk = (p, G, G_T, e, g) \leftarrow \mathcal{G}(1^k); g_r, h_r, g_s, h_s, g_t, h_t \leftarrow G;\\ &(r, s, t) \leftarrow \mathcal{A}(gk, g_r, h_r, g_s, h_s, g_t, h_t) : (r, s, t) \in G^3 \setminus \{(1, 1, 1)\}\\ &\wedge \quad e(g_r, r)e(g_s, s)e(g_t, t) = 1 \quad \wedge \quad e(h_r, r)e(h_s, s)e(h_t, t) = 1\Big] \approx 0.\end{aligned}$$

EXAMPLE. Let us give an example taken from Boneh and Franklin [BF03] of a bilinear group where we believe the simultaneous triple pairing assumption holds. Consider the elliptic curve $y^2 = x^3 + 1 \bmod q$, where $q = 2 \bmod 3$ is a prime. It is straightforward to check that a point (x, y) is on the curve. Furthermore, picking $y \in \mathbb{Z}_q$ at random and computing $x = (y^2 - 1)^{\frac{q+1}{3}} \bmod q$ gives us a random point on the curve. The curve has a total of $q + 1$ points, including the point at infinity. When generating such groups, we will pick p as a random k-bit prime. We then let q be the smallest prime so $p|q + 1$ and define G to be the order p subgroup of the curve. The target group is $G_T = \mathbb{F}^*_{q^2}$ and the bilinear map is the modified Weil-pairing.

3.1 Security Analysis of the Simultaneous Triple Pairing Assumption

To gain confidence in the simultaneous triple pairing assumption, we will explore its relationship with other cryptographic assumptions. First, we will show that the simultaneous triple pairing assumption follows from a computational hardness assumption called the simultaneous pairing assumption introduced by Groth and Lu [GL07]. Groth and Lu proved that the simultaneous pairing assumption is secure in the generic group model and since the security reduction only uses generic group operations this implies that the simultaneous triple pairing assumption is secure in the generic group model. Second, we will show that the simultaneous triple pairing assumption follows from the decision linear assumption from Boneh, Boyen and Shacham [BBS04].

RELATION TO THE SIMULTANEOUS PAIRING ASSUMPTION. The simultaneous pairing problem is given $g_1 = g^{x_1}, h_1 = g^{x_1^2}, \ldots, g_n = g^{x_n}, h_n = g^{x_n^2}$ for random $x_1, \ldots, x_n \in \mathbb{Z}_p$ find a non-trivial set of elements $\mu_1, \ldots, \mu_n \in G$ such that

$$\prod_{i=1}^{n} e(g_i, \mu_i) = 1 \quad \wedge \quad \prod_{i=1}^{n} e(h_i, \mu_i) = 1.$$

Definition 7 (Simultaneous pairing assumption) *The simultaneous pairing assumption holds for $\mathcal{G}$ if for all non-uniform polynomial time adversaries $\mathcal{A}$ we have*

$$\Pr\Big[gk = (p, G, G_T, e, g) \leftarrow \mathcal{G}(1^k); x_1, \ldots, x_n \leftarrow \mathbb{Z}_p; g_1 = g^{x_1}, h_1 = g^{x_1^2}, \ldots, g_1 = g^{x_n}, h_n = g^{x_n^2};$$
$$(\mu_1, \ldots, \mu_n) \leftarrow \mathcal{A}(gk, g_1, h_1, \ldots, g_n, h_n) :$$
$$\prod_{i=1}^{n} e(g_i, \mu_i) = 1 \wedge \prod_{i=1}^{n} e(h_i, \mu_i) = 1 \wedge \exists i : \mu_i \neq 1\Big] \approx 0.$$

Theorem 8 *If the simultaneous pairing assumption with $n = 3$ holds for $\mathcal{G}$, then the simultaneous triple pairing assumption holds for $\mathcal{G}$.*

Proof. Suppose we have an adversary $\mathcal{A}$ that breaks the simultaneous triple pairing assumption with probability $\epsilon(k)$. We will show how to construct an adversary $\mathcal{B}$ that breaks the simultaneous pairing assumption for $n = 3$ with probability higher than $\epsilon(k) - 6/p$.

Given a random simultaneous pairing problem instance $(gk, g_1, h_1, g_2, h_2, g_3, h_3)$ the adversary $\mathcal{B}$ picks at random $\rho, \sigma, \tau \leftarrow \mathbb{Z}_p^*$ and computes

$$g_r = g_1^{\rho} \qquad h_r = h_1^{\rho} \qquad g_s = g_2^{\sigma} \qquad h_s = h_2^{\sigma} \qquad g_t = g_3^{\tau} \qquad h_t = h_3^{\tau}.$$

If $g_1 = 1, g_2 = 1$ or $g_3 = 1$ it is trivial to solve the simultaneous pairing problem. Provided the discrete logarithms of g_1, g_2, g_3 are non-trivial, *i.e.*, $x_1 \neq 1, x_2 \neq 1, x_3 \neq 1$, we get a random distribution of 6 group elements in $G \setminus \{1\}$, which has statistical distance less than $6/p$ from a random six-tuple of group elements in G. The adversary now runs $\mathcal{A}$ on $(gk, g_r, h_r, g_s, h_s, g_t, h_t)$ and gets an non-trivial simultaneous triple pairing solution (r, s, t) with probability higher than $\epsilon(k) - 6/p$. We have

$$\begin{aligned} e(g_r, r)e(g_s, s)e(g_t, t) &= e(g_1, r^{\rho})e(g_2, s^{\sigma})e(g_3, t^{\tau}) = 1 \\ e(h_r, r)e(h_s, s)e(h_t, t) &= e(h_1, r^{\rho})e(h_2, s^{\sigma})e(h_3, t^{\tau}) = 1, \end{aligned}$$

so $(r^{\rho}, s^{\sigma}, t^{\tau})$ is a non-trivial solution to the simultaneous pairing problem. □

RELATION TO THE DECISION LINEAR ASSUMPTION. The decision linear problem is to decide whether a tuple $(g_1, g_2, g_3, g_1^{\rho}, g_2^{\sigma}, h_3)$ has $h_3 = g_3^{\rho+\sigma}$ or h_3 is random.

Definition 9 (Decision linear assumption) *The decision linear assumption holds for $\mathcal{G}$ if for all non-uniform polynomial time adversaries $\mathcal{A}$ we have:*

$$\begin{aligned} &\Pr\Big[gk = (p, G, G_T, e, g) \leftarrow \mathcal{G}(1^k)\ ;\ g_1, g_2, g_3 \leftarrow G;\ \rho, \sigma \leftarrow \mathbb{Z}_p\ : \\ &\qquad\qquad \mathcal{A}(gk, g_1, g_2, g_3, g_1^\rho, g_2^\sigma, g_3^{\rho+\sigma}) = 1\Big] \\ \approx\ &\Pr\Big[gk = (p, G, G_T, e, g) \leftarrow \mathcal{G}(1^k)\ ;\ g_1, g_2, g_3, h_3 \leftarrow G;\ \rho, \sigma \leftarrow \mathbb{Z}_p\ : \\ &\qquad\qquad \mathcal{A}(gk, g_1, g_2, g_3, g_1^\rho, g_2^\sigma, h_3) = 1\Big]. \end{aligned}$$

Theorem 10 *The simultaneous triple pairing assumption holds for $\mathcal{G}$ if the decision linear assumption holds for $\mathcal{G}$.*

Proof. Let us first make an observation regarding the solution (r, s, t) of a simultaneous triple pairing problem $(g_r, g_s, g_t, h_r, h_s, h_t)$ when $e(g_r, h_s) \neq e(g_s, h_r)$. Taking discrete logarithms this means $x_r y_s \neq x_s y_r$ so the matrix

$$\begin{pmatrix} x_r & x_s \\ y_r & y_s \end{pmatrix}$$

has non-zero determinant. Consequently, (x_r, y_r) and (x_s, y_s) are linearly independent in $\mathbb{Z}_p^2$ and for any g_t, h_t there is a unique solution $(a, b) \in \mathbb{Z}_p^2$ so

$$g_t = g_r^a g_s^b \quad \wedge \quad h_t = h_r^a h_s^b.$$

The solution (r, s, t) satisfies

$$e(g_r, r)e(g_s, s)e(g_t, t) = 1 \quad \wedge \quad e(h_r, r)e(h_s, s)e(h_t, t) = 1,$$

which implies

$$e(g_r, rt^a)e(g_s, st^b) = 1 \quad \wedge \quad e(h_r, rt^a)e(h_s, st^b) = 1.$$

Since (x_r, y_r) and (x_s, y_s) are linearly independent, this means $r = t^{-a}$ and $s = t^{-b}$. The solution is non-trivial if and only if $t \neq 1$.

Suppose now we have an adversary $\mathcal{A}$ that breaks the simultaneous triple pairing assumption with probability $\epsilon(k)$. We will show how to construct an adversary $\mathcal{B}$ that breaks the decision linear assumption with at least probability $\epsilon(k) - 7/p$.

$\mathcal{B}$ is presented with a decision linear problem instance $(gk, g_1, g_2, g_3, h_1, h_2, h_3)$. $\mathcal{B}$ picks at random $\alpha, \beta \leftarrow \mathbb{Z}_p$ and sets $g_t = g_3^2 g_1^\alpha g_2^\beta$ and $h_t = h_3 h_1^\alpha h_2^\beta$. $\mathcal{B}$ then runs $\mathcal{A}$ on the simultaneous triple pairing problem $(gk, g_1, g_2, g_t, h_1, h_2, h_t)$. There is more than $1 - 3/p$ chance that the decision linear problem instance has $g_1 \neq 1, g_2 \neq 1$ and $e(g_1, h_2) \neq e(g_2, h_1)$.

Let us analyze the probability that $\mathcal{A}$ generates a valid solution (r, s, t) to the simultaneous triple pairing problem when $g_1 \neq 1, g_2 \neq 1, e(g_1, h_2) \neq e(g_2, h_1)$. As above, the latter inequality implies that (x_1, x_2) and (y_1, y_2) are linearly independent in $\mathbb{Z}_p^2$ and therefore g_t, h_t are random in G. The probability of $g_1 = 1 \vee g_2 = 1 \vee e(g_1, h_2) = e(g_2, h_1)$ on a random instance of the simultaneous triple pairing problem is less than $3/p$, so we have at last $\epsilon(k) - 3/p$ probability of finding a non-trivial solution (r, s, t) to the simultaneous triple pairing problem created by $\mathcal{B}$.

In total we therefore have more than $\epsilon(k) - 6/p$ probability that the decision linear challenge has $g_1 \neq 1, g_2 \neq 1, e(g_1, h_2) \neq e(g_2, h_1)$ and that we find a non-trivial solution (r, s, t) so

$$e(g_1, r)e(g_2, s)e(g_t, t) = 1 \quad \wedge \quad e(h_1, r)e(h_2, s)e(h_t, t) = 1.$$

By our observation above, the solution satisfies $r = t^{-a}$ and $s = t^{-b}$, where a, b are unknown exponents such that

$$g_t = g_1^a g_2^b \quad \wedge \quad h_t = h_1^a h_2^b,$$

and since it is non-trivial we have $t \neq 1$. Plugging in $g_t = g_3^2 g_1^\alpha g_2^\beta$ and $h_t = h_3 h_1^\alpha h_2^\beta$ gives us

$$e(g_1, rt^\alpha)e(g_2, st^\beta)e(g_3^2, t) = 1 \quad \wedge \quad e(h_1, rt^\alpha)e(h_2, st^\beta)e(h_3, t) = 1$$

so $(r' = t^{\alpha - a}, s' = t^{\beta - b}, t)$ is a solution to the simultaneous triple pairing problem $(gk, g_1, g_2, g_3^2, h_1, h_2, h_3)$ with $t \neq 1$.

There are two types of challenges we can meet in the decision linear game with $g_1 \neq 1, g_2 \neq 1, e(g_1, h_2) \neq e(g_2, h_1)$. If the challenge is a linear tuple $(g_1, g_2, g_3, g_1^\rho, g_2^\sigma, g_3^{\rho+\sigma})$ we have

$$g_3^2 = g_1^{x_3/x_1} g_2^{x_3/x_2} \quad \wedge \quad h_3 = h_1^{x_3/x_1} h_2^{x_3/x_2}.$$

This means the solution (r', s', t) satisfies $r' = t^{-x_3/x_1}$ and $s' = t^{-x_3/x_2}$ with $t \neq 1$. We then have the following equalities

$$e(g_1, r')e(g_3, t) = 1 \quad \wedge \quad e(g_2, s')e(g_3, t) = 1$$

that can be tested by $\mathcal{B}$ after running $\mathcal{A}$ to get (r, s, t). In this case, $\mathcal{B}$ will output 1.

On the other hand, if the challenge is chosen randomly as $(g_1, g_2, g_3, g_1^\rho, g_2^\sigma, h_3)$, we only have $1/p$ chance of $h_3 = h_1^{x_3/x_1} h_2^{x_3/x_2}$ being true. If $h_3 \neq h_1^{x_3/x_1} h_2^{x_3/x_2}$ we have $r' \neq t^{-x_3/x_1}$ or $s' \neq t^{-x_3/x_2}$. Since $t \neq 1$ this means

$$e(g_1, r')e(g_3, t) \neq 1 \quad \vee \quad e(g_2, s')e(g_3, t) \neq 1.$$

In this case, $\mathcal{B}$ will output 0.

To conclude the description of $\mathcal{B}$ we let it output 1, whenever $g_1 = 1, g_2 = 1, e(g_1 h_2) = e(g_2, h_1)$ or $\mathcal{A}$ does not produce a valid solution (r, s, t). In case $\mathcal{B}$ is presented with a linear tuple, it will therefore always output 1. In case, $\mathcal{B}$ is presented with a random tuple instead, it outputs 0 with probability at least $\epsilon(k) - 7/p$. □

4 Homomorphic Trapdoor Commitment to Group Elements

We will now present the homomorphic trapdoor commitment scheme. The setup algorithm generates a bilinear group (p, G, G_T, e, g). The commitment scheme permits committing to n group elements from G. More precisely, we have message space $\mathcal{M}_{ck} = G^n$, randomizer space $\mathcal{R}_{ck} = G^2$ and commitment space $\mathcal{C}_{ck} = G_T^2$, where each of them are interpreted as a group using entry-wise multiplication.

Setup: On input 1^k return $gk = (p, G, G_T, e, g) \leftarrow \mathcal{G}(1^k)$.

Key generator: On input gk pick at random $x_1, y_1, \ldots, x_n, y_n, x_r, y_r, x_s, y_s \leftarrow \mathbb{Z}_p$ such that $x_r y_s \neq x_s y_r$ and define

$$g_1 = g^{x_1} \quad h_1 = g^{y_1} \cdots g_n = g^{x_n} \quad h_n = g^{y_n} \quad g_r = g^{x_r} \quad h_r = g^{y_r} \quad g_s = g^{x_s} \quad h_s = g^{y_s}.$$

Since $x_r y_s \neq x_s y_r$ we can compute

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} = \begin{pmatrix} x_r & x_s \\ y_r & y_s \end{pmatrix}^{-1}.$$

The commitment key is $ck = (gk, g_1, h_1, \ldots, g_n, h_n, g_r, h_r, g_s, h_s)$ and the trapdoor key is $tk = (gk, x_1, y_1, \ldots, x_n, y_n, x_r, x_s, y_r, y_s, \alpha, \beta, \gamma, \delta)$.

Commitment: Using commitment key ck on input message $(m_1, \ldots, m_n) \in G^n$ pick randomizer $(r, s) \leftarrow G^2$. The commitment is $(c, d) \in G_T^2$ given by

$$c = e(g_r, r)e(g_s, s)\prod_{i=1}^{n} e(g_i, m_i) \quad \wedge \quad d = e(h_r, r)e(h_s, s)\prod_{i=1}^{n} e(h_i, m_i).$$

Trapdoor commitment: Using commitment key ck and trapdoor key tk we generate an equivocal commitment $(c, d) \in G_T^2$ by picking $(r, s) \leftarrow G^2$ and computing

$$c = e(g_r, r)e(g_s, s) \quad \wedge \quad d = e(h_r, r)e(h_s, s).$$

The corresponding equivocation key is $ek = (tk, r, s)$.

Trapdoor opening: To trapdoor open an equivocal commitment $(c, d) \in G_T^2$ to a message $(m_1, \ldots, m_n) \in G^n$ using the equivocation key ek, we compute

$$a = r^{x_r} s^{x_s} \prod_{i=1}^{n} m_i^{-x_i} \quad \wedge \quad b = r^{y_r} s^{y_s} \prod_{i=1}^{n} m_i^{-y_i}.$$

We then compute

$$r' = a^{\alpha} b^{\beta} \quad \wedge \quad s' = a^{\gamma} b^{\delta}.$$

We return the opening (r', s') of (c, d) to message $(m_1, \ldots, m_n)$.

Theorem 11 $(\mathcal{G}, K, \mathrm{com}, \mathrm{Tcom}, \mathrm{Topen})$ *described above is a homomorphic trapdoor commitment scheme to* n *group elements. It has the perfect trapdoor property and assuming the simultaneous triple pairing assumption holds for* $\mathcal{G}$ *the commitment scheme is computationally binding.*

Proof. Let us first prove the commitment scheme is homomorphic. The message space is G^n, the randomizer space is G^2 and the commitment space is G_T^2, which with entry-wise multiplication all are finite abelian groups. Given a commitment key $ck = (gk, g_1, h_1, \ldots, g_n, h_n, g_r, h_r, g_s, h_s)$ it is straightforward to check the homomorphic property. For all $(m_1, \ldots, m_n), (m'_1, \ldots, m'_n) \in G^n$ and all $(r, s), (r', s') \in G^2$ we have

$$\begin{aligned} e(g_r, r)e(g_s, s)\prod_{i=1}^{n} e(g_i, m_i) \cdot e(g_r, r')e(g_s, s')\prod_{i=1}^{n} e(g_i, m'_i) &= e(g_r, rr')e(g_s, ss')\prod_{i=1}^{n} e(g_i, m_i m'_i) \\ e(h_r, r)e(h_s, s)\prod_{i=1}^{n} e(h_i, m_i) \cdot e(h_r, r')e(h_s, s')\prod_{i=1}^{n} e(h_i, m'_i) &= e(h_r, rr')e(h_s, ss')\prod_{i=1}^{n} e(h_i, m_i m'_i) \end{aligned}$$

Next, we will prove that the commitment scheme has the perfect trapdoor property. By construction, $x_r y_s \neq x_s y_r$ so (x_r, y_r) and (x_s, y_s) are linearly independent in $\mathbb{Z}_p^2$. We can deduce from this that both real commitments and trapdoor commitments are distributed uniformly at random in G_T^2, because of their $e(g_r, r)e(g_s, s)$ and $e(h_r, r)e(h_s, s)$ factors where r, s are chosen randomly from G. The linear independence of (x_r, y_r) and (x_s, y_s) also implies that for any pair $(c, d) \in G_T^2$ and a set of messages $(m_1, \ldots, m_n) \in G^n$ there is a unique randomizer $(r, s) \in G^2$ so

$$c = e(g_r, r)e(g_s, s)\prod_{i=1}^{n} e(g_i, m_i) \quad \wedge \quad d = e(h_r, r)e(h_s, s)\prod_{i=1}^{n} e(h_i, m_i).$$

To conclude the proof for the perfect trapdoor property, we therefore just need to show that the trapdoor opening algorithm gives the correct opening (r', s') of the commitment. Since

$$\begin{pmatrix} x_r & x_s \\ y_r & y_s \end{pmatrix} \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

we have

$$\begin{aligned} e(g_r, r')e(g_s, s') &= e(g^{x_r}, a^{\alpha}b^{\beta})e(g^{x_s}, a^{\gamma}b^{\delta}) = e(g, a^{x_r\alpha + x_s\gamma})e(g, b^{x_r\beta + x_s\delta}) = e(g, a) \\ e(h_r, r')e(h_s, s') &= e(g^{y_r}, a^{\alpha}b^{\beta})e(g^{y_s}, a^{\gamma}b^{\delta}) = e(g, a^{y_r\alpha + y_s\gamma})e(g, b^{y_r\beta + y_s\delta}) = e(g, b). \end{aligned}$$

By plugging in $a = r^{x_r}s^{x_s}\prod_{i=1}^{n} m_i^{-x_i}$ and $b = r^{y_r}s^{y_s}\prod_{i=1}^{n} m_i^{-y_i}$ we get

$$\begin{aligned} e(g_r, r')e(g_s, s')\prod_{i=1}^{n} e(g_i, m_i) &= e(g, r^{x_r}s^{x_s})\prod_{i=1}^{n} e(g, m_i^{x_i - x_i}) = e(g_r, r)e(g_s, s) = c \\ e(h_r, r')e(h_s, s')\prod_{i=1}^{n} e(h_i, m_i) &= e(g, r^{y_r}s^{y_s})\prod_{i=1}^{n} e(g, m_i^{y_i - y_i}) = e(h_r, r)e(h_s, s) = d, \end{aligned}$$

as we wanted.

Finally, we will prove that the commitment scheme is computationally binding if the simultaneous triple pairing assumption holds for $\mathcal{G}$. More precisely, we will show that if $\mathcal{A}$ has probability $\epsilon(k)$ of breaking the binding property, then there is an algorithm $\mathcal{B}$ that breaks the simultaneous triple pairing assumption with at least $\epsilon(k) - 1/p$ chance.

Let $(gk, g_r, g_s, g_t, h_r, h_s, h_t)$ be a random simultaneous triple pairing challenge for $\mathcal{B}$. In case $e(g_r, h_s) = e(g_s, h_r)$, we have a solution to the simultaneous triple pairing problem given by

$$e(g_r, g_s)e(g_s, g_r^{-1})e(g_t, 1) = 1 \wedge e(h_r, g_s)e(h_s, g_r^{-1})e(h_t, 1) = 1,$$

which is non-trivial unless $g_s = g_r = 1$. If $g_s = g_r = 1$, then we have a solution $(h_s, h_r^{-1}, 1)$ to the simultaneous triple pairing problem, unless $h_s = h_r = 1$ as well. But in case $g_r = g_s - h_r = h_s = 1$, we have the non-trivial solution $(g, g, 1)$ to the simultaneous triple pairing problem.

Let us now consider the case $e(g_r, h_s) \neq e(g_s, h_r)$. In this case, the discrete logarithms satisfy $x_r y_s \neq x_s y_r$ just as when we generate a commitment key. We pick at random $\rho_1, \sigma_1, \tau_1, \ldots, \rho_n, \sigma_n, \tau_n \leftarrow \mathbb{Z}_p$ and define $g_1, h_1, \ldots, g_n, h_n$ by

$$g_i = g_r^{\rho_i} g_s^{\sigma_i} g_t^{\tau_i} \quad \wedge \quad h_i = h_r^{\rho_i} h_s^{\sigma_i} h_t^{\tau_i}.$$

Since (x_r, y_r) and (x_s, y_s) are linearly independent in $\mathbb{Z}_p^2$ all these group elements are randomly distributed in G. This means $ck = (gk, g_1, h_1, \ldots, g_n, h_n, g_r, h_r, g_s, h_s)$ has the same distribution as commitment keys generated by K.

We give this ck to $\mathcal{A}$ and with probability $\epsilon(k)$ it produces two different messages $(m_1, \ldots, m_n), (m'_1, \ldots, m'_n)$ and randomizers $(r, s), (r', s')$ so

$$\mathrm{com}_{ck}(m_1, \ldots, m_n; r, s) = \mathrm{com}_{ck}(m'_1, \ldots, m'_n; r', s').$$

Define $\mu_1 = m'_1 m_1^{-1}, \ldots, \mu_n = m'_n m_n^{-1}$ and $r'' = r'r^{-1}, s'' = s's^{-1}$. By the homomorphic property of the commitment scheme we have $\mathrm{com}_{ck}(\mu_1, \ldots, \mu_n; r'', s'') = (1, 1)$. This gives us

$$\begin{aligned} e(g_r, r'')e(g_s, s'')\prod_{i=1}^{n} e(g_i, \mu_i) = e(g_r, r''\prod_{i=1}^{n}\mu_i^{\rho_i})e(g_s, s''\prod_{i=1}^{n}\mu_i^{\sigma_i})e(g_t, \prod_{i=1}^{n}\mu_i^{\tau_i}) &= 1 \\ e(h_r, r'')e(h_s, s'')\prod_{i=1}^{n} e(h_i, \mu_i) = e(h_r, r''\prod_{i=1}^{n}\mu_i^{\rho_i})e(h_s, s''\prod_{i=1}^{n}\mu_i^{\sigma_i})e(h_t, \prod_{i=1}^{n}\mu_i^{\tau_i}) &= 1. \end{aligned}$$

Since $(m_1, \ldots, m_n)$ and $(m'_1, \ldots, m'_n)$ are different, there is at least one $\mu_i \neq 1$. Recall $g_i = g_r^{\rho_i} g_s^{\sigma_i} g_t^{\tau_i}$ and $h_i = h_r^{\rho_i} h_s^{\sigma_i} h_t^{\tau_i}$ for random $\rho_i, \sigma_i, \tau_i \leftarrow \mathbb{Z}_p$. Since (x_r, y_r) and (x_s, y_s) are linearly independent in $\mathbb{Z}_p^2$ there is for any τ'_i a unique pair $(\rho'_i, \sigma'_i) \in \mathbb{Z}_p^2$ that would yield g_i, h_i. This means from $\mathcal{A}$'s perspective τ_i is a perfectly hidden random value in $\mathbb{Z}_p$. The probability that $\prod_{i=1}^n \mu_i^{\tau_i} = 1$ is therefore at most $1/p$.

So if $e(g_r, h_s) = e(g_s, h_r)$ the adversary $\mathcal{B}$ breaks the simultaneous triple pairing problem with probability 1. If $e(g_r, h_s) \neq e(g_s, h_r)$ the adversary $\mathcal{B}$ has at least $\epsilon(k) - 1/p$ chance of $(r'' \prod_{i=1}^n \mu_i^{\rho_i}, s'' \prod_{i=1}^n \mu_i^{\sigma_i}, \prod_{i=1}^n \mu_i^{\tau_i})$ being a non-trivial solution to the simultaneous triple pairing problem. □

References

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55, 2004.

[BF03] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341, 2005.

[Bra00] Stefan Brands. *Rethinking Public Key Infrastructures and Digital Certificates; Building in Privacy*. MIT Press, 2000.

[BW06] Xavier Boyen and Brent Waters. Compact group signatures without random oracles. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 427–444, 2006.

[DF02] Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *ASIACRYPT*, volume 2501 of *Lecture Notes in Computer Science*, pages 125–142, 2002.

[DN02] Ivan Damgård and Jesper Buus Nielsen. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 581–596, 2002. Full paper available at `http://www.brics.dk/RS/01/41/index.html`.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[FO97] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 16–30, 1997.

[FS01] Jun Furukawa and Kazue Sako. An efficient scheme for proving a shuffle. In *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 368–387, 2001.

[GL07] Jens Groth and Steve Lu. A non-interactive shuffle with pairing based verifiability. In *ASIACRYPT*, volume 4833 of *Lecture Notes in Computer Science*, pages 51–67, 2007.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 97–111, 2006.

[GQ88] Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both trasmission and memory. In *EUROCRYPT*, volume 330 of *Lecture Notes in Computer Science*, pages 123–128, 1988.

[Gro06] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *ASIACRYPT*, volume 4248 of *Lecture Notes in Computer Science*, pages 444–459, 2006. Full paper available at `http://www.brics.dk/∼jg/NIZKGroupSignFull.pdf`.

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432, 2008. Full paper available at `http://eprint.iacr.org/2007/155`.

[KZ06] Aggelos Kiayias and Hong-Sheng Zhou. Concurrent blind signatures without random oracles. In *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 49–62, 2006.

[Lip03] Helger Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In *ASIACRYPT*, volume 2894 of *Lecture Notes in Computer Science*, pages 398–415, 2003.

[Nef01] C. Andrew Neff. A verifiable secret shuffle and its application to e-voting. In *ACM CCS*, pages 116–125, 2001.

[OU98] Tatsuaki Okamoto and Shigenori Uchiyama. A new public-key cryptosystem as secure as factoring. In *EUROCRYPT*, volume 1403 of *Lecture Notes in Computer Science*, pages 308–318, 1998.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite residuosity classes. In *EUROCRYPT*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–239, 1999.

[Ped91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140, 1991.