On Secure Multi-Party Computation in Black-Box Groups

Yvo Desmedt[1]*, Josef Pieprzyk[2], Ron Steinfeld[2], and Huaxiong Wang[2,3]

[1] Dept. of Computer Science, University College London, UK
[2] Centre for Advanced Computing – Algorithms and Cryptography (ACAC)
Dept. of Computing, Macquarie University, North Ryde, Australia
[3] Division of Math. Sci., Nanyang Technological University, Singapore
{josef,rons,hwang}@comp.mq.edu.au, hxwang@ntu.edu.sg

Abstract. We study the natural problem of secure n-party computation (in the passive, computationally unbounded attack model) of the n-product function $f_G(x_1, \ldots, x_n) = x_1 \cdot x_2 \cdots x_n$ in an arbitrary finite group $(G, \cdot)$, where the input of party P_i is $x_i \in G$ for $i = 1, \ldots, n$. For flexibility, we are interested in protocols for f_G which require only *black-box* access to the group G (i.e. the only computations performed by players in the protocol are a group operation, a group inverse, or sampling a uniformly random group element).

Our results are as follows. First, on the negative side, we show that if $(G, \cdot)$ is non-abelian and $n \geq 4$, then no $\lceil n/2 \rceil$-private protocol for computing f_G exists. Second, on the positive side, we initiate an approach for construction of black-box protocols for f_G based on k-of-k threshold secret sharing schemes, which are efficiently implementable over any black-box group G. We reduce the problem of constructing such protocols to a combinatorial colouring problem in planar graphs. We then give two constructions for such graph colourings. Our first colouring construction gives a protocol with optimal collusion resistance $t < n/2$, but has exponential communication complexity $O(n\binom{2t+1}{t}^2)$ group elements (this construction easily extends to general adversary structures). Our second probabilistic colouring construction gives a protocol with (close to optimal) collusion resistance $t < n/\mu$ for a graph-related constant $\mu \leq 2.948$, and has efficient communication complexity $O(nt^2)$ group elements. Furthermore, we believe that our results can be improved by further study of the associated combinatorial problems.

Key Words: Multi-Party Computation, Non-Abelian Group, Black-Box, Planar Graph, Graph Colouring.

1 Introduction

Background. Groups form a natural mathematical structure for cryptography. In particular, the most popular public-key encryption schemes today (RSA [17]

* A part of this research was funded by NSF ANI-0087641, EPSRC EP/C538285/1. Yvo Desmedt is BT Chair of Information Security.

and Diffie-Hellman/ElGamal [8, 9]) both operate in abelian groups. However, the discovery of efficient quantum algorithms for breaking these cryptosystems [19] gives increased importance to the construction of alternative cryptosystems in non-abelian groups (such as [13, 15]), where quantum algorithms seem to be much less effective.

Motivated by such emerging cryptographic applications of non-abelian groups, we study the natural problem of secure n-party computation (in the passive, computationally unbounded attack model) of the n-product function $f_G(x_1, \ldots, x_n) = x_1 \cdot x_2 \cdots x_n$ in an arbitrary finite group $(G, \cdot)$, where the input of party P_i is $x_i \in G$ for $i = 1, \ldots, n$. For flexibility, we are interested in protocols for f_G which require only *black-box* access to the group G (i.e. the only computations performed by players in the protocol are a group operation $(x, y) \rightarrow x \cdot y$, a group inverse $x \rightarrow x^{-1}$, or sampling a random group element $x \in_R G$). It is well known that when $(G, \cdot)$ is abelian, a straightforward 2-round black-box protocol exists for f_G which is t-private (secure against t parties) for any $t < n$ and has communication complexity $O(n^2)$ group elements. However, to our knowledge, when $(G, \cdot)$ is non-abelian, no constructions of black-box protocols for f_G have been designed until now. Consequently, to construct a t-private protocol for f_G in some non-abelian group G one currently has to resort to adopting existing non black-box methods, which may lead to efficiency problems (see 'Related Work').

Our Results. Our results are as follows. First, on the negative side, we show that if $(G, \cdot)$ is non-abelian and $n \geq 4$, then no $\lceil n/2 \rceil$-private protocol for computing f_G exists. Second, on the positive side, we initiate an approach for construction of black-box protocols for f_G based only on k-of-k threshold secret sharing schemes (whereas previous non black-box protocols rely on Shamir's t-of-n threshold secret sharing scheme over a ring). We reduce the problem of constructing such protocols to a combinatorial colouring problem in planar graphs. We then give two constructions for such graph colourings. Our first colouring construction gives a protocol with optimal collusion resistance $t < n/2$, but has exponential communication complexity $O(n\binom{2t+1}{t}^2)$ group elements (this construction also easily generalises to general Q^2 adversary structures $\mathcal{A}$ as defined in [11], giving communication complexity $O(n|\mathcal{A}|^2)$ group elements). Our second probabilistic colouring construction gives a protocol with (close to optimal) collusion resistance $t < n/\mu$ for a graph-related constant $\mu \leq 2.948$, and has efficient communication complexity $O(nt^2)$ group elements. Furthermore, we believe that our results can be improved by further study of the associated combinatorial problems. We note that our protocols easily and naturally generalize to other arbitrary functions defined over the group G.

Related Work. There are two known non black-box methods for constructing a t-private protocol for the n-product function f_G for any $t < n/2$. They are both based on Shamir's t-of-n threshold secret sharing scheme [18] and its generalizations.

The first method [3, 4, 10] requires representing f_G as a boolean circuit, and uses Shamir's secret sharing scheme over the field $GF(p)$ for a prime $p > 2t + 1$. This protocol has total communication complexity $O(t^2 \log t \cdot N_{AND}(f_G))$ bits, where $N_{AND}(f_G)$ denotes the number of AND gates in the boolean AND/NOT circuit for computing f_G. Thus this protocol is efficient only for very small groups G, for which $N_{AND}(f_G)$ is manageable.

The second method [5] (see also [2] for earlier work) requires representing f_G as an arithmetic circuit over a finite *ring* R, and accordingly, uses a generalization of Shamir's secret sharing scheme to any finite ring. This protocol has total communication complexity $O(t^2 \log t \cdot N_M(f_G) \cdot \ell(R))$ bits, where $N_M(f_G)$ is the number of multiplication operations in the circuit for f_G over R and $\ell(R) \geq \log |R|$ denotes the number of bits needed for representing elements of R. If we 'embed' group G in the ring $R = R(G)$, so that R inherits the multiplication operation of G, then $N_M(f_G) = n - 1$, and hence the protocol from [5] has total communication complexity $O(nt^2 \log t \cdot \ell(R(G)))$ bits, compared to $O(nt^2 \cdot \ell(G))$ bits for our (second) protocol (assuming $t < n/2.948$), where $\ell(G) \geq \log |G|$ is the representation length of elements of G. Hence, for $t < n/2.948$, the communication complexity of our protocol for f_G is smaller than the one from [5] by a factor $\Theta(\frac{\ell(R(G))}{\ell(G)} \cdot \log t)$ (for $n/2.948 < t < n/2$, the protocol of [5] is still asymptotically the most efficient known proven protocol). Note that, for any finite group G, we can always take $R(G)$ to be the *group algebra* (or group ring) of G over $GF(2)$, which can be viewed as a $|G|$-dimensional vector space over $GF(2)$ consisting of all linear combinations of the elements of G (the basis vectors) with coefficients from $GF(2)$ (the product operation of $R(G)$ is defined by the operation of G extended by linearity and associativity, and the addition operation of $R(G)$ is defined componentwise). However, for this generic choice of $R(G)$ we have $\ell(R(G)) = |G|$, so, assuming $\ell(G) = \log |G|$, our protocol reduces communication complexity by a factor $\Theta(\frac{|G|}{\log |G|} \cdot \log t)$, which is exponentially large in the representation length $\log |G|$. In the worst case, we may have $\ell(R(G)) = \Theta(\ell(G))$ and our protocol may only give a saving factor $O(\log t)$ over the protocol from [5], e.g. this is the case for $G = GL(k, 2)$ (the group of invertible $k \times k$ matrices over $GF(2)$). We remark that this $O(\log t)$ saving factor arises essentially from the fact that Shamir's secret sharing for $2t+1$ shares requires a ring of size greater than $2t + 1$, and hence, for a secret from $GF(2)$, the share length is greater than the secret length by a factor $\Theta(\log t)$ (whereas our approach does not use Shamir's sharing and hence does not suffer from this length expansion). On the other hand, for sharing a secret from $GF(q)$ for 'large' q ($q > 2t + 1$), Shamir's scheme is ideal, so for specific groups such as $G = GL(k, q)$ with $q > 2t + 1$, the communication cost of the protocols from [2, 5] reduces to $O(nt^2 \cdot \ell(R(G)))$.

Organization. The paper is organized as follows. Section 2 contains definitions and results we use. In Section 3 we show that $t < n/2$ is necessary for secure computation of f_G. In Sections 4.2 and 4.3 we show how to construct a t-private protocol for f_G given a 't-Reliable' colouring of a planar graph. Then in Section 4.4, we present two constructions of such t-Reliable colourings. Finally, Section 4.5 summarizes some generalizations and extensions, and Section 5 concludes with some open problems. Some proofs are omitted from this version of the paper due to space limitations – they are available in the full version [6].

2 Preliminaries

We recall the definition of secure multi-party computation in the passive (semi-honest), computationally unbounded attack model, restricted to deterministic symmetric functionalities and perfect emulation [10]. Let $[n]$ denote the set $\{1,\ldots,n\}$.

Definition 1. *Let $f : (\{0,1\}^*)^n \to \{0,1\}^*$ denote an n-input, single-output function, and let Π be an n-party protocol for computing f. We denote the party input sequence by $\mathbf{x} = (x_1,\ldots,x_n)$, the joint protocol view of parties in subset $I \subseteq [n]$ by $\mathsf{VIEW}_I^{\Pi}(\mathbf{x})$, and the protocol output by $\mathsf{OUT}^{\Pi}(\mathbf{x})$. For $0 < t < n$, we say that Π is a t-private protocol for computing f if there exists a probabilistic polynomial-time algorithm S, such that, for every $I \subset [n]$ with $\#I \leq t$ and every $\mathbf{x} \in (\{0,1\}^*)^n$, the random variables*

$$\langle \mathsf{S}(I, \mathbf{x}_I, f(\mathbf{x})), f(\mathbf{x})\rangle \text{ and } \langle \mathsf{VIEW}_I^{\Pi}(\mathbf{x}), \mathsf{OUT}^{\Pi}(\mathbf{x})\rangle$$

are identically distributed, where $\mathbf{x}_I$ denotes the projection of the n-ary sequence $\mathbf{x}$ on the coordinates in I.

To prove our result we will invoke a combinatorial characterization of 2-input functions for which a 1-private 2-party computation protocol exists, due to Kushilevitz [12]. To state this result, we need the following definitions.

Definition 2. *Let $M = C \times D$ be a matrix, where C is the set of rows and D is the set of columns. Define a binary relation $\sim$ on pairs of rows of M as follows: $x_1, x_2 \in C$ satisfy $x_1 \sim x_2$ if there exists $y \in D$ such that $M_{x_1,y} = M_{x_2,y}$. Let $\equiv$ denote the equivalence relation on the rows of M which is the transitive closure of $\sim$. Similarly, we define $\sim$ and $\equiv$ on the columns of M.*

Definition 3. *A matrix M is called* forbidden *if all its rows are equivalent, all its columns are equivalent, and not all entries of M are equal.*

Definition 4. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,\ldots,m-1\}$ be any 2-input function. A matrix M for f is a $2^n \times 2^n$ matrix with entries in $\{0,\ldots,m-1\}$, where each row x of f corresponds to a value for the first input to f, each column y corresponds to a value for the second input to f, and the entry $M_{x,y}$ contains the value $f(x,y)$.*

Theorem 1 (Kushilevitz [12]). *Let f be a 2-input function and let M be a matrix for f. Then a 1-private 2-party protocol for computing f exists if and only if M does not contain a forbidden submatrix.*

3 Honest Majority is Necessary for n-Product in Non-Abelian Groups

We show that an honest majority $t < n/2$ is necessary for secure computation of the n-product function in non-abelian groups.

Theorem 2. *Let $(G, \cdot)$ denote a finite non-abelian group and let $n \geq 4$. There does not exist a $\lceil \frac{n}{2} \rceil$-private protocol for computing $f_G(x_1, \ldots, x_n) = x_1 \cdot x_2 \cdots x_n$.*

Proof. The proof proceeds by contradiction; we show that if a $\lceil \frac{n}{2} \rceil$-private protocol Π exists for f_G for $n \geq 4$, then we can construct a 1-private 2-party protocol for a 2-input function f'_G whose matrix M' contains a forbidden submatrix, thus contradicting Theorem 1.

Lemma 1. *Suppose there exists a $\lceil \frac{n}{2} \rceil$-private n-party protocol Π for computing the n-input function $f_G : G^n \rightarrow G$ defined by $f_G(x_1, \ldots, x_n) = x_1 \cdots x_n$ for $n \geq 4$. Then we can construct a 1-private 2-party protocol Π' for computing the 2-input function $f'_G : G^2 \times G^2 \rightarrow G$ defined by $f'_G((x'_1, x'_3), (x'_2, x'_4)) = x'_1 \cdot x'_2 \cdot x'_3 \cdot x'_4$.*

Proof. Given party P'_1 with input (x'_1, x'_3) and party P'_2 with input (x'_2, x'_4), the protocol Π' runs as follows. First, if $n \geq 5$, we partition the set $\{5, \ldots, n\}$ into two disjoint subsets S'_1 and S'_2 such that the size of both S'_1 and S'_2 is at most $\lceil \frac{n}{2} \rceil - 2$ (namely, if n is even we take $\#S'_1 = \#S'_2 = n/2 - 2$, and if n is odd we take $\#S'_1 = (n-3)/2$ and $\#S'_2 = (n-5)/2$). Then $\Pi'(P'_1, P'_2)$ consists of running the n-party protocol $\Pi(P_1, \ldots, P_n)$ where:

- P'_1 plays the role of parties $(P_1, P_3, \{P_i\}_{i \in S'_1})$ in Π, and sets those parties inputs to be $x_1 = x'_1$, $x_3 = x'_3$, and $x_i = 1$ for all $i \in S'_1$, respectively.
- P'_2 plays the role of parties $(P_2, P_4, \{P_i\}_{i \in S'_2})$ in Π, and sets those parties inputs to be $x_2 = x'_2$, $x_4 = x'_4$ and $x_i = 1$ for all $i \in S'_2$, respectively.

The 1-privacy of protocol $\Pi'(P'_1, P'_2)$ for computing f'_G follows from the $\lceil \frac{n}{2} \rceil$-privacy of protocol $\Pi(P_1, \ldots, P_n)$ for computing f_G because:

- $f_G(x'_1, x'_2, x'_3, x'_4, 1, \ldots, 1) = f'_G(x'_1, x'_2, x'_3, x'_4) = x'_1 \cdot x'_2 \cdot x'_3 \cdot x'_4$ for all $x'_1, x'_2, x'_3, x'_4 \in G$.
- For each (x'_1, x'_2, x'_3, x'_4), the view of P'_1 (resp. P'_2) in protocol $\Pi'(P'_1, P'_2)$ is identical to the view of a set of at most $\lceil \frac{n}{2} \rceil$ parties in protocol $\Pi(P_1, \ldots, P_n)$ whose inputs are known to P'_1 (resp. P'_2), with special settings of 1 for some inputs. Thus the same view simulator algorithm S of Π can be used to simulate the view in Π'.

This completes the proof. □

Lemma 2. *For any non-abelian group G, the matrix M for the 2-input function $f'_G : G^2 \times G^2 \rightarrow G$ defined by $f'_G((x'_1, x'_3), (x'_2, x'_4)) = x'_1 \cdot x'_2 \cdot x'_3 \cdot x'_4$ contains a 2×2 forbidden submatrix.*

Proof. Observe from Definitions 2 and 3 that any 2×2 matrix with 3 equal elements and a fourth distinct element is a forbidden matrix. Now recall that the rows of matrix M for f'_G are indexed by $(x'_1, x'_3) \in G^2$, the columns of M are indexed by $(x'_2, x'_4) \in G^2$, and the entry of M at row (x'_1, x'_3) and column (x'_2, x'_4) is $M_{(x'_1, x'_3), (x'_2, x'_4)} = x'_1 \cdot x'_2 \cdot x'_3 \cdot x'_4$. Also, since G is non-abelian, there exist a pair of elements a and b in G such that a and b do not commute and $a, b \neq 1$. Consider the 2×2 submatrix of M formed by the intersections of the 2 rows $(1,1)$ and (a,a^{-1}) and the 2 columns $(1,1)$ and (b,b^{-1}) (these row and column pairs are distinct because $a,b \neq 1$). We claim that this submatrix is forbidden. Indeed, three of the submatrix entries are equal because $M_{(1,1),(1,1)} = M_{(a,a^{-1}),(1,1)} = M_{(1,1),(b,b^{-1})} = 1$, and the remaining fourth entry is distinct because $M_{(a,a^{-1}),(b,b^{-1})} = a \cdot b \cdot a^{-1} \cdot b^{-1} = (a \cdot b) \cdot (b \cdot a)^{-1} \neq 1$ since a and b do not commute. This completes the proof. □

Combining Lemma 1 and Lemma 2, we conclude that if a $\lceil \frac{n}{2} \rceil$-private protocol Π exists for f_G for $n \geq 4$, then we obtain a contradiction to Theorem 1. This completes the proof. □

4 Constructions

4.1 Our Approach: Black Box Non-Abelian Group Protocols

Our protocols will treat the group G as a black box in the sense that the only computations performed by players in our protocols will be one of the following three: Multiply (Given $x \in G$ and $y \in G$, compute $x \cdot y$), Inverse (Given $x \in G$, compute x^{-1}), and Random Sampling (Choose a uniformly random $x \in G$). It is easy to see that these three operations are sufficient for implementing a perfect k-of-k threshold secret sharing scheme. We use this k-of-k scheme as a fundamental building block in our protocols. The following proposition is easy to prove.

Proposition 1. *Fix $x \in G$ and integers k and $j \in [k]$, and suppose we create a k-of-k sharing $(s_x(1), s_x(2), \ldots, s_x(k))$ of x by picking the $k-1$ shares $\{s_x(i)\}_{i \in [k] \setminus \{j\}}$ uniformly and independently at random from G, and computing $s_x(j)$ to be the unique element of G such that $x = s_x(1)s_x(2)\cdots s_x(k)$. Then the distribution of the shares $(s_x(1), s_x(2), \ldots, s_x(k))$ is independent of j.*

4.2 Construction of n-Product Protocol from a Shared 2-Product Subprotocol

We begin by reducing the problem of constructing a t-private protocol for the n-product function $f(x_1, \ldots, x_n) = x_1 \cdots x_n$ (where party P_i holds input x_i for $i = 1, \ldots, n$), to the problem of constructing a subprotocol for the *Shared* 2-Product function $f'(x,y) = x \cdot y$, where inputs x, y and output $z = x \cdot y$ are shared among the parties. We define for this subprotocol a so-called *strong* t-privacy definition, which will be needed later to prove the (standard) t-privacy of the full n-product protocol built from subprotocol Π_S. The definition of strong t-privacy requires the adversary's view simulator to simulate *all* output shares except one share not held by the adversary, in addition to simulating the internal subprotocol view of the adversary.

Definition 5 (Shared n-Party 2-Product Subprotocol). *A n-Party Shared 2-Product subprotocol Π_S with sharing parameter ℓ and share ownership functions $\mathcal{O}_x, \mathcal{O}_y, \mathcal{O}_z : [\ell] \to [n]$ has the following features:*

- *Input: For $j = 1, \ldots, \ell$, party $P_{\mathcal{O}_x(j)}$ holds jth share $s_x(j) \in G$ of x and party $P_{\mathcal{O}_y(j)}$ holds jth share $s_y(j) \in G$ of y, where $\mathbf{s}_x = (s_x(1), s_x(2), \ldots, s_x(\ell))$ and $\mathbf{s}_y = (s_y(1), s_y(2), \ldots, s_y(\ell))$ denote ℓ-of-ℓ sharing of $x \overset{\text{def}}{=} s_x(1) \cdot s_x(2) \cdots s_x(\ell)$ and $y \overset{\text{def}}{=} s_y(1) \cdot s_y(2) \cdots s_y(\ell)$, respectively.*
- *Output: For $j = 1, \ldots, \ell$, party $P_{\mathcal{O}_z(j)}$ holds jth share $s_z(j)$ of output product $z \overset{\text{def}}{=} s_z(1) \cdots s_z(\ell)$.*
- *Correctness: We say that that Π_S is correct if, for all protocol inputs $\mathbf{s}_x = (s_x(1), s_x(2), \ldots, s_x(\ell))$ and $\mathbf{s}_y = (s_y(1), s_y(2), \ldots, s_y(\ell))$, the output shares $\mathbf{s}_z = (s_z(1), s_z(2), \ldots, s_z(\ell))$ satisfy*

$$z = x \cdot y$$

where $x \overset{\text{def}}{=} s_x(1) \cdot s_x(2) \cdots s_x(\ell)$, $y \overset{\text{def}}{=} s_y(1) \cdot s_y(2) \cdots s_y(\ell)$ and $z \overset{\text{def}}{=} s_z(1) \cdots s_z(\ell)$.

- *Strong t-Privacy: We say that Π_S achieves **strong t-privacy** if there exists a probabilistic simulator algorithm S_{Π_S} such that for all $I \subset [n]$ with $\#I \leq t$, there exist $j^* \in [\ell]$ with $\mathcal{O}_x(j^*) \notin I$ and $\mathcal{O}_z(j^*) \notin I$, and $j^*_y \in [\ell]$ with $\mathcal{O}_y(j^*_y) \notin I$ such that for all protocol inputs $\mathbf{s}_x = (s_x(1), \ldots, s_x(\ell))$ and $\mathbf{s}_y = (s_y(1), \ldots, s_y(\ell))$, the random variables*

$$\langle \mathsf{S}_{\Pi_S}(I, \{\mathbf{s}_x(j)\}_{j \in [n] \setminus \{j^*\}}, \{\mathbf{s}_y(j)\}_{j \in [\ell] \setminus \{j^*_y\}}) \rangle \text{ and }$$
$$\langle \mathsf{VIEW}_I^{\Pi_S}(\mathbf{s}_x, \mathbf{s}_y), \{s_z(j)\}_{j \in [\ell] \setminus \{j^*\}} \rangle$$

*are identically distributed (over the random coins of Π_S). Here $\mathsf{VIEW}_I^{\Pi_S}(\mathbf{s}_x, \mathbf{s}_y)$ denotes the view of I in subprotocol Π_S run with input shares $\mathbf{s}_x, \mathbf{s}_y$, and $s_z(j)$ denotes the jth output share. If $j^*_y = j^*$ for all I, then we say Π_S achieves **symmetric strong t-privacy**.*

Remark 1. The share ownership functions $\mathcal{O}_x, \mathcal{O}_y, \mathcal{O}_z$ specify for each share index $j \in [\ell]$, the indices $\mathcal{O}_x(j), \mathcal{O}_y(j), \mathcal{O}_z(j)$ in $[n]$ of the party which holds the jth input shares $s_x(j)$ and $s_y(j)$ and jth output share $s_z(j)$, respectively.

Remark 2. The adversary view simulator $\mathsf{S}_{\Pi_\mathsf{S}}$ for collusion I is given all input shares except the j^*th x-share $s_x(j^*)$ and j^*_yth y-share $s_y(j^*_y)$ (where $j^*, j^*_y \in [\ell]$, which depend on I, are indices of shares given to players *not* in I), and outputs all output shares except the j^*th share $s_z(j^*)$ of z. Because, for each I, the same value of index j^* is used for both x-input shares and output shares, this allows multiple simulator runs to be composed, using output shares of one subprotocol run as x-input shares in a following subprotocol run, as shown in the security proof of the following construction. If in addition, *symmetric* strong t-privacy is achieved, one can use output shares of one subprotocol run as either x-input *or* y-input shares for the following subprotocol run, allowing for more efficient protocols.

We now explain our construction of an n-Product Protocol $\Pi(T, \Pi_S)$ given a binary computation tree T for f_G with n leaf nodes corresponding to the n

protocol inputs (as illustrated in Fig. 1), and a Shared 2-Product subprotocol Π_S with sharing parameter ℓ and share ownership functions $\mathcal{O}_x, \mathcal{O}_y, \mathcal{O}_z$. The protocol Π begins with each party P_j computing an ℓ-of-ℓ sharing of its input x_j, and distributing out these shares to the n parties according to the share ownership functions $\mathcal{O}_x, \mathcal{O}_y$ of Π_S. Then protocol Π performs each of the internal node 2-product computations of the computation tree T on ℓ-of-ℓ sharings of the internal node's two children nodes by running the shared 2-product subprotocol Π_S, resulting in an ℓ-of-ℓ sharing of the internal node value. Eventually this recursive process gives an ℓ-of-ℓ sharing of the root node value $x_1 \cdots x_n$ of T, which is broadcast to all parties.

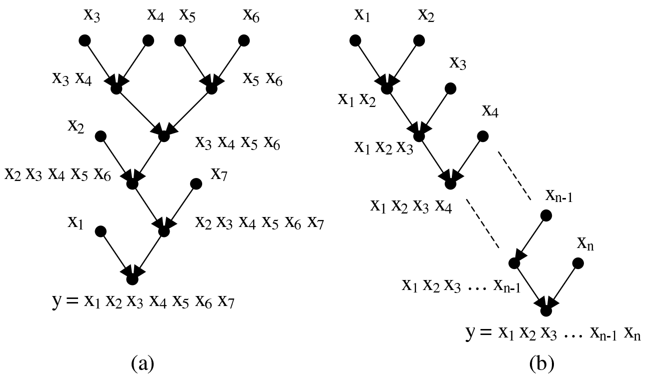

Fig. 1. (a) Example of a binary tree T with $n = 7$ leaves. (b) The *slanted linear tree* T_{slin} with n leaves.

The following Lemma establishes the t-privacy of protocol $\Pi(T, \Pi_S)$, assuming the correctness and strong t-privacy of subprotocol Π_S. Refer to [6] for a proof.

Lemma 3. *For any binary tree T with n leaves, if the n-party Shared 2-Product subprotocol Π_S satisfies correctness and symmetric strong t-privacy (see Definition 5), then protocol $\Pi(T, \Pi_S)$ is an n-party t-private protocol for computing n-Product function $f_G(x_1, \ldots, x_n) = x_1 \cdots x_n$. For the* slanted linear *binary tree T_{slin} shown in Fig 1(b), the above result holds even if Π_S satisfies (ordinary) strong t-privacy (i.e. symmetric strong t-privacy is* not *needed in this case).*

4.3 Construction of a t-Private n-Party Shared 2-Product Subprotocol from a t-Reliable n-Colouring of a Planar Graph

Next, we reduce the problem of constructing a t-Private n-Party Shared 2-Product Subprotocol Π_S to a combinatorial problem defined below of finding a 't-Reliable n-Colouring' of the nodes of a planar graph. We note that our notion of a 't-Reliable n-Colouring' is closely related to a similar notion defined in [7], and shown to be equivalent to the existence of private communication via a network graph in which each node is assigned one of n possible colours and the adversary controls all nodes with colours belonging to a t-colour subset I.

Consider a Planar Directed Acyclic Graph (PDAG) $\mathcal{G}$ having 2ℓ source (input) nodes drawn in a horizontal row at the top, ℓ sink (output) nodes drawn in a horizontal row at the bottom, and $\sigma_{\mathcal{G}}$ nodes overall. We use PDAG $\mathcal{G}$ to represent a blackbox protocol, where the input/output nodes are labelled by the protocol input/output group elements, and the internal graph nodes are labelled by intermediate protocol values. Each internal graph node is also assigned a *colour* specifying the player which computes the internal node value. The graph edges represent group elements sent from one player to another. The computation performed at each node is multiplication of the values on all incoming edges and resharing the product along the outgoing edges using the k-of-k secret sharing scheme in Proposition 1. All computations in the ith round of the 2-Product subprotocol correspond to the ith row (from the top) in the PDAG. Communications between nodes correspond to edges between consecutive rows.

Actually to construct a protocol for any non-abelian group our requirement on graph $\mathcal{G}$ is slightly stronger than planarity and can be precisely defined as follows.

Definition 6 (Admissible PDAG). *We call graph $\mathcal{G}$ an* Admissible PDAG with share parameter ℓ and size parameter m *if it has the following properties:*

- *Nodes of $\mathcal{G}$ are drawn on a square $m \times m$ grid of points (each node of $\mathcal{G}$ is located at a grid point but some grid points may not be occupied by nodes). Rows of the grid are indexed from top to bottom and columns from left to right by the integers $1, 2, \ldots, m$. A node of $\mathcal{G}$ at row i and column j is said to have index (i, j). $\mathcal{G}$ has 2ℓ source (input) nodes on top row 1, and ℓ sink (output) nodes on bottom row m.*
- *Incoming edges of a node on row i only come from nodes on row $i-1$, and outgoing edges of a node on row i only go to nodes on row $i+1$.*
- *For each row i and column j, let $\eta_1^{(i,j)} < \ldots < \eta_{q^{(i,j)}}^{(i,j)}$ denote the ordered column indices of the $q^{(i,j)} > 0$ nodes on level $i+1$ which are connected to node (i, j) by an edge. Then, for each $j = 1, \ldots, m-1$, we have*

$$\eta_{q^{(i,j)}}^{(i,j)} \leq \eta_1^{(i,j+1)}, \tag{1}$$

i.e. the rightmost node on level $i+1$ connected to node (i, j) is to the left of (or equal to) the leftmost node on level $i+1$ connected to node $(i, j+1)$.

We call the left ℓ source nodes on row 1 (indexed $(1,1), \ldots, (1, \ell)$) the 'x-input' nodes and the last ℓ source nodes on row 1 (indexed $(1, \ell+1), \ldots, (1, 2\ell)$) the '$y$-input' nodes. By ith x-input node, we mean the x-input node at position i from the left. We define the ith y-input and ith output node similarly.

Let $C : [m] \times [m] \rightarrow [n]$ be an n-Colouring function that associates to each node (i, j) of $\mathcal{G}$ a colour $C(i, j)$ chosen from a set of n possible colours $[n]$. We now define the notion of a t-Reliable n-Colouring.

Definition 7 (t-Reliable n-Colouring). *We say that $C : [m] \times [m] \rightarrow [n]$ is a t-Reliable n-Colouring for admissible PDAG $\mathcal{G}$ (with share parameter ℓ and size parameter m) if for each t-colour subset $I \subset [n]$, there exist $j^* \in [\ell]$ and $j_y^* \in [\ell]$ such that:*

- *There exists a path $PATH_x$ in $\mathcal{G}$ from the j^*th x-input node to the j^*th output node, such that none of the path node colours are in subset I (we call such a path I-avoiding), and*
- *There exists an I-avoiding path $PATH_y$ in $\mathcal{G}$ from the j_y^*th y-input node to the j^*th output node.*

If $j_y^* = j^*$ for all I, we say that C is a Symmetric *t-Reliable n-Colouring.*

Remark 3. The paths $PATH_x$ and $PATH_y$ in Definition 7 are free to move in *any* direction along each edge of directed graph $\mathcal{G}$, i.e. for this definition we regard $\mathcal{G}$ as an *undirected* graph (throughout the paper we assume that a path is *simple*, i.e. free of cycles; hence each node on the path is only visited once).

An example of an admissible PDAG with I-avoiding paths $PATH_x$ and $PATH_y$ is shown in Fig 2(a). Given an admissible PDAG $\mathcal{G}$ (with share parameter ℓ and size parameter m) and an associated t-Reliable n-Colouring $C : [m] \times [m] \rightarrow [n]$, we construct a t-Private n-Party Shared 2-Product Subprotocol $\Pi_S(\mathcal{G}, C)$.

Shared 2-Product Subprotocol $\Pi_S(\mathcal{G}, C)$

Input: We define the share ownership functions $\mathcal{O}_x, \mathcal{O}_y, \mathcal{O}_z$ of $\Pi_S(\mathcal{G}, C)$ according to the colours assigned by C to the input and output nodes of $\mathcal{G}$ (i.e. $\mathcal{O}_x(j) = C(1, j)$, $\mathcal{O}_y(j) = C(1, \ell + j)$, $\mathcal{O}_z(j) = C(m, j)$ for $j = 1, \ldots, \ell$). For $j = 1, \ldots, \ell$, party $P_{\mathcal{O}_x(j)}$ holds jth share $s_x(j) \in G$ of x and party $P_{\mathcal{O}_y(j)}$ holds jth share $s_y(j) \in G$ of y, where $\mathbf{s}_x = (s_x(1), s_x(2), \ldots, s_x(\ell))$ and $\mathbf{s}_y = (s_y(1), s_y(2), \ldots, s_y(\ell))$ denote ℓ-of-ℓ sharing of $x \overset{\text{def}}{=} s_x(1) \cdot s_x(2) \cdots s_x(\ell)$ and $y \overset{\text{def}}{=} s_y(1) \cdot s_y(2) \cdots s_y(\ell)$, respectively.

For each row $i = 1, \ldots, m$ and column $j = 1, \ldots, m$ of $\mathcal{G}$, party $P_{C(i,j)}$ does the following:

- $P_{C(i,j)}$ computes a label $v^{(i,j)}$ for node (i, j) of $\mathcal{G}$ as follows. If $i = 1$, $P_{C(i,j)}$ defines $v^{(i,j)} = s_x(j)$ for $j \leq \ell$ and $v^{(i,j)} = s_y(j)$ for $\ell + 1 \leq j \leq 2\ell$. If $i > 1$, $P_{C(i,j)}$ computes $v^{(i,j)}$ by multiplying the shares received from nodes at previous row $i - 1$ (labels of edges between a node on row $i - 1$ and node (i, j)), ordered from left to right according to the sender node column index.
- If $i = m$, $P_{C(m,j)}$ sets output share j to be the label $v^{(m,j)}$,
- else, if $i < m$, let $\eta_1^{(i,j)} < \ldots < \eta_{q^{(i,j)}}^{(i,j)}$ denote the ordered column indices of the nodes on level $i + 1$ which are connected to node (i, j) by an edge. $P_{C(i,j)}$ chooses $q^{(i,j)} - 1$ uniformly random elements from G and computes a $q^{(i,j)}$-of-$q^{(i,j)}$ secret sharing $s_1^{(i,j)}, \ldots, s_{q^{(i,j)}}^{(i,j)}$ of label $v^{(i,j)}$ such that:
$$v^{(i,j)} = s_1^{(i,j)} \cdots s_{q^{(i,j)}}^{(i,j)}.$$
- For $k = 1, \ldots, q^{(i,j)}$, $P_{C(i,j)}$ sends share $s_k^{(i,j)}$ to party $P_{C(i+1, \eta_k^{(i,j)})}$ and labels edge from node (i, j) to node $(i + 1, \eta_k^{(i,j)})$ by the share $s_k^{(i,j)}$.

Note that the correctness of Π_S follows from the fact that the product of node values at each row of PDAG $\mathcal{G}$ is preserved and hence equal to $x \cdot y$, thanks to condition (1) in Definition 6.

Lemma 4. *If $\mathcal{G}$ is an admissible PDAG and C is a t-Reliable n-Colouring for $\mathcal{G}$ then $\Pi_S(\mathcal{G}, C)$ achieves strong t-privacy. Moreover, if C is a* Symmetric *t-Reliable n-Colouring, then $\Pi_S(\mathcal{G}, C)$ achieves* Symmetric *strong t-privacy.*

Proof. (Sketch) The full proof of Lemma 4 can be found in [6]. Here we only explain the main idea by considering the case when the I-avoiding paths $PATH_x$ and $PATH_y$ only have downward edges (in [6] we extend the argument to paths with upward edges). Consider $PATH_x$ from the j^*th x-input node to the j^*th output node. At the first node $PATH_x(1)$ on the path, although the node value $v(1) = s_x(j^*)$ is not known to the view simulator S_{Π_S}, we may assume, by Proposition 1, that in the real subprotocol Π_S, when node $PATH_x(1)$ shares out its node label among its q outgoing edges, it sends new random elements (labels) r_i on each of the $q-1$ outgoing edges *not* on $PATH_x$. Thus simulator S_{Π_S} can easily simulate all outgoing edge values of $PATH_x(1)$ which are not on $PATH_x$. The same argument shows that for all kth nodes $PATH_x(k)$ and $PATH_y(k)$ on $PATH_x$ and $PATH_y$ respectively, simulator S_{Π_S} can simulate all values on outgoing edges of $PATH_x(k)$ and $PATH_y(k)$ which are *not* on $PATH_x$ or $PATH_y$ by independent random elements. The values on edges along $PATH_x$ or $PATH_y$ depend on the inputs $s_x(j^*)$ and $s_y(j^*_y)$ which are not known to simulator S_{Π_S}, but since the paths $PATH_x$ and $PATH_y$ are I-avoiding, these values are not in the view of I and need not be simulated by S_{Π_S}. Since S_{Π_S} knows all inputs to Π_S it can compute all other edge values in the Π_S, including all outputs except the j^*th one (which is on $PATH_x$ and $PATH_y$), as required. □

4.4 Constructions of t-Reliable n-Colourings of Planar Graphs

We now present two general constructions of t-Reliable n-Colourings of planar graphs which can be used to build t-Private n-Party protocols for the n-Product function in any finite group as explained in the previous sections. Our first deterministic construction achieves optimal collusion security ($t < n/2$) but has exponential complexity ($\ell = \binom{n}{t}$). Our second probabilistic construction has a slightly suboptimal collusion security ($t < n/2.948$) but has a very efficient linear complexity ($\ell = O(n)$).

The PDAG. The admissible PDAG $\mathcal{G}_{tri}(\ell', \ell)$ that we consider has sharing parameter ℓ and has $\ell' \times \ell$ nodes. It is shown in Fig. 2(b). The nodes of $\mathcal{G}_{tri}(\ell', \ell)$ are arranged in an $\ell' \times \ell$ node grid. Let (i, j) denote the node at row $i \in [\ell']$ (from the top) and column j (from the left). There are three types of edges in directed graph $\mathcal{G}_{tri}(\ell', \ell)$: (1) Horizontal edge: An edge connecting two adjacent nodes on the same row, directed from right to left (i.e. from node (i, j) to node $(i, j-1)$, for $i \in [\ell']$, $j \in [\ell] \setminus \{1\}$), (2) Vertical edge: An edge connecting two adjacent nodes on the same column, directed from top to bottom (i.e. from node

(i,j) to node $(i+1,j)$, for $i \in [\ell'] \setminus \{\ell'\}$, $j \in [\ell]$), and (3) Diagonal edge: An edge connecting node (i,j) to node $(i+1,j-1)$, for $i \in [\ell'] \setminus \{\ell'\}$, $j \in [\ell] \setminus \{1\}$).

The ℓ nodes on the top row (row 1) of $\mathcal{G}_{tri}$ are the x-input nodes, indexed from left to right. The top ℓ nodes on the rightmost column of $\mathcal{G}_{tri}$ (column ℓ) are the y-input nodes, indexed from top to bottom.

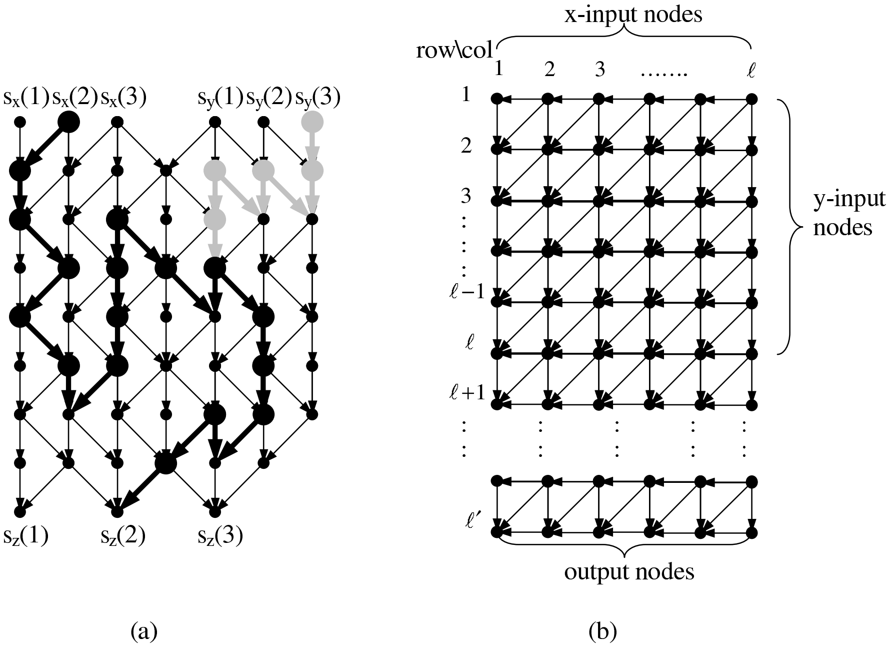

Fig. 2. (a) Example of an admissible PDAG $\mathcal{G}$ with sharing parameter $\ell = 3$ (node colours are not indicated). For a given collusion I, an example I-avoiding path $PATH_x$ is shown in heavy black, and an example I-avoiding path $PATH_y$ (until the meeting with $PATH_x$) is shown in heavy gray. In this example, we have $j^* = 2$ and $j^*_y = 3$. (b) The admissible PDAG $\mathcal{G}_{tri}(\ell', \ell)$.

Remark 4. The reader may notice that the above specification of $\mathcal{G}_{tri}$ does not formally satisfy the convention for drawing an admissible PDAG as defined in Def. 6, due to the horizontal edges and the fact that the y-input nodes are arranged along a column, rather than along the same row as the x-input nodes. However, it is easy to see that $\mathcal{G}_{tri}$ can also be drawn strictly according to Def. 6. Namely by rotating the drawing of $\mathcal{G}_{tri}$ in Fig. 2 by 45 degrees anticlockwise, the horizontal edges become diagonal edges, and x-inputs and y-inputs can be formally put on the same row by adding appropriate 'connecting' nodes of the same colour as the corresponding input nodes of $\mathcal{G}_{tri}$. These are only formal changes in drawing conventions, and there is no change in the protocol itself. In this section we use the drawing convention in Fig. 2 for clarity.

Remark 5. All diagonal edges in the definition of $\mathcal{G}_{tri}$ above are parallel (with a 'positive slope', when using the drawing convention in Fig 2). However, it is clear that the admissible PDAG requirements are still satisfied if we remove from $\mathcal{G}_{tri}$ some 'positive slope' diagonal edges and add some 'negative slope' diagonal edges (connecting a node (i, j) to node $(i + 1, j + 1)$, for some $i \in [\ell'] \setminus \{\ell'\}$, $j \in [\ell] \setminus \{\ell\}$), as long as planarity of $\mathcal{G}$ is preserved (no two diagonal edges intersect). We denote such 'generalised' PDAGs by $\mathcal{G}_{gtri}$.

First Construction C_{comb} ($t < n/2$ and $\ell = \binom{n}{t}$). We now present an explicit construction of a t-Reliable n-Colouring C_{comb} of the square graph $\mathcal{G}_{tri}(\ell, \ell)$. The construction applies for all $n \geq 2t + 1$ (i.e. $t \leq \lfloor \frac{n-1}{2} \rfloor$), and hence (by Section 3) the n-Product protocol constructed from it by the method of Sections 4.2 and 4.3 achieves $\lfloor \frac{n-1}{2} \rfloor$-privacy (which is optimal, as shown in Section 3). Unfortunately, the sharing parameter in this construction $\ell = \binom{n}{t}$, is exponential in t (and therefore the protocol communication cost is also exponential in t).

Colouring C_{comb} for graph $\mathcal{G}_{tri}(\ell, \ell)$ with $\ell = \binom{n}{t}$ and $n \geq 2t + 1$

1. Let $I_1, \ldots, I_\ell$ denote the sequence of all $\ell = \binom{n}{t}$ t-colour subsets of $[n]$ (in some ordering).
2. For each $(i, j) \in [\ell] \times [\ell]$, define the colour $C(i, j)$ of node (i, j) of $\mathcal{G}_{tri}(\ell, \ell)$ to be any colour in the set $S_{i,j} = [n] \setminus (I_i \bigcup I_j)$ (note that since $|I_i| = |I_j| = t$ and $n \geq 2t + 1$, the set $S_{i,j}$ contains at least $n - (|I_i| + |I_j|) \geq n - 2t \geq 1$ colours, so $S_{i,j}$ is never empty).

Lemma 5. *For $n \geq 2t + 1$, the colouring C_{comb} is a Symmetric t-Reliable n-Colouring for graph $\mathcal{G}_{tri}(\ell, \ell)$, with $\ell = \binom{n}{t}$.*

Proof. Given each t-colour subset $I \subseteq [n]$, let j^* denote the index of I in the sequence $I_1, \ldots, I_\ell$ of all t-colour subsets used to construct C_{comb}, i.e $I_{j^*} = I$. By construction of C_{comb}, none of the nodes of $\mathcal{G}_{tri}(\ell, \ell)$ along column j^* have colours in $I_{j^*} = I$. Hence one can take column j^* of $\mathcal{G}_{tri}(\ell, \ell)$ as $PATH_x$. Similarly, we also know that none of the nodes of $\mathcal{G}_{tri}(\ell, \ell)$ along row j^* have colours in $I_{j^*} = I$, so one can take $PATH_y$ to consist of all nodes on row j^* which are on columns $j \geq j^*$, followed by all nodes on column j^* which are on rows $i \geq j^*$. Thus C_{comb} is a Symmetric t-Reliable n-Colouring for graph $\mathcal{G}_{tri}(\ell, \ell)$, as required. □

Remark 6. The colouring C_{comb} remains a Symmetric t-Reliable n-Colouring even if we remove all diagonal edges from $\mathcal{G}_{tri}(\ell, \ell)$ (since the paths $PATH_x$ and $PATH_y$ only contain vertical and horizontal edges).

Combining Lemma 5 (applied to a subset of $n' = 2t + 1 \leq n$ colours from $[n]$) with Lemmas 3 and 4, we have

Corollary 1. *For any $t < n/2$, there exists a black-box t-private protocol for f_G with communication complexity $O(n\binom{2t+1}{t}^2)$ group elements.*

Second Construction C_{rand} ($t < n/2.948$ and $\ell = O(n)$). It is natural to ask whether the exponentially large sharing parameter $\ell = \binom{n}{t}$ can be reduced. Our second construction C_{rand} shows that this is certainly the case when $t < n/2.948$, achieving a linear sharing parameter $\ell = O(n)$.

As a first step towards our second construction, we relax the properties required from C in Definition 7 to slightly simpler requirements for the square graph $\mathcal{G}_{tri}(\ell, \ell)$ (i.e. $\ell' = \ell$), as follows.

Definition 8 (Weakly t-Reliable n-Colouring). *We say that $C : [\ell] \times [\ell] \rightarrow [n]$ is a* Weakly *t-Reliable n-Colouring for graph $\mathcal{G}_{tri}(\ell, \ell)$ if for each t-colour subset $I \subset [n]$:*

- *There exists an I-avoiding path P_x in $\mathcal{G}$ from a node on the top row (row 1) to a node on the bottom row (row ℓ). We call such a path an I-avoiding* top-bottom *path.*
- *There exists an I-avoiding path P_y in $\mathcal{G}$ from a node on the rightmost column (column ℓ) to a node on the leftmost column (column 1). We call such a path an I-avoiding* right-left *path.*

Note that in the above definition of Weak t-Reliability, the index of the starting node of path P_x in the top row need not be the same as the index of the exit node of P_x in the bottom row (whereas in the definition of t-Reliability, $PATH_x$ must exit at the same position along the output row as the position in the top row where $PATH_x$ begins).

The following lemma shows that finding a Weakly t-Reliable n-Colouring for the square graph $\mathcal{G}_{tri}(\ell, \ell)$ is sufficient for constructing a (standard) t-Reliable n-Colouring for a rectangular graph $\mathcal{G}_{gtri}(2\ell - 1, \ell)$. The idea is to add $\ell - 1$ additional rows to $\mathcal{G}_{tri}(\ell, \ell)$ by appending a 'mirror image' (reflected about the last row) of itself, as shown in Fig. 3 (refer to [6] for the detailed proof).

Lemma 6. *Let $C : [\ell] \times [\ell] \rightarrow [n]$ be a Weakly t-Reliable n-Colouring (see Def. 8) for square admissible PDAG $\mathcal{G}_{tri}(\ell, \ell)$. Then we can construct a (standard) t-Reliable n-Colouring (see Def. 7) for a rectangular admissible PDAG $\mathcal{G}_{gtri}(2\ell - 1, \ell)$.*

For our second colouring construction, we use the 'probabilistic method' [1], namely we choose the colour of each node in the square graph $\mathcal{G}_{tri}(\ell, \ell)$ independently and uniformly at random from $[n]$. Although there is a finite error probability p that such a random n-Colouring will not be Weakly t-Reliable, we show that if $n/t > 2.948$ and we use a sufficiently large (but only *linear* in n) sharing parameter $\ell = O(n)$, then the error probability p can be made arbitrarily small. Moreover, p decreases exponentially fast with ℓ, so p can be easily made negligible.

Colouring C_{rand} for graph $\mathcal{G}_{tri}(\ell, \ell)$ with $\ell = O(n)$ and $n \geq 2.948t$

For each $(i, j) \in [\ell] \times [\ell]$, choose the colour $C(i, j)$ of node (i, j) of $\mathcal{G}_{tri}(\ell, \ell)$ independently and uniformly at random from $[n]$.

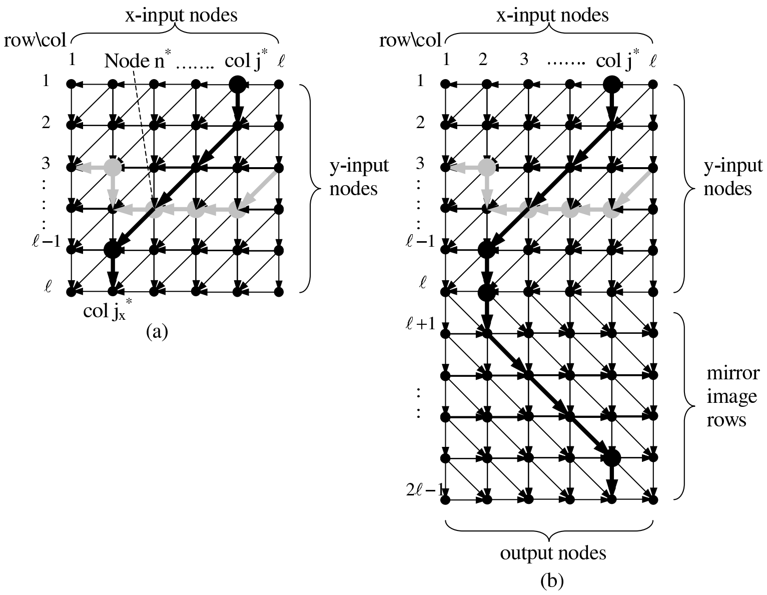

Fig. 3. (a) Example paths in square PDAG $\mathcal{G}_{tri}(\ell, \ell)$ for a given Weakly t-Reliable n-Colouring (P_x in heavy black, P_y in heavy gray). (b) Corresponding paths in rectangular PDAG $\mathcal{G}_{gtri}(2\ell - 1, \ell)$.

To analyse this construction, we will make use of the following counting Lemma. Here, for any right-left path in $\mathcal{G}_{tri}(\ell, \ell)$, we define its *length* as the number of nodes on the path. We say a path is *minimal* if removing any node from the path disconnects the path.

Lemma 7. *The number $N_P(k, \ell)$ of minimal right-left paths of length k in graph $\mathcal{G}_{tri}(\ell, \ell)$ is upper bounded as*

$$N_P(k, \ell) \leq c(\mu) \cdot \ell \cdot \mu^k,$$

for some constants $\mu, c(\mu)$, with $\mu \leq 2.948$. We call the minimal possible value for μ the connective constant *of $\mathcal{G}_{tri}(\ell, \ell)$.*

Proof. For a minimal right-left path, there are ℓ possible starting nodes on the rightmost column. We may assume without loss of generality that the first edge of the path is not a vertical edge. For the ith starting node on the rightmost column, there are at most 2 possibilities for the first path edge: a horizontal edge, or a diagonal edge. For $j \geq 1$, let $N_i(j)$ denote the number of minimal paths in $\mathcal{G}_{tri}(\ell, \ell)$ of length j starting at the ith node on the rightmost column. Note that the paths counted in $N_i(j)$ are not necessarily right-left paths, i.e. the last node in the path need not be on the leftmost column.

We use induction on j to show $N_i(j) \leq 3^{j-1}$ for $j \geq 2$. We have already shown above the basis step $N_i(2) = 2 < 3$. For the induction step, suppose that $N_i(j) \leq 3^{j-1}$ for some $j \geq 2$. We show that $N_i(j+1) \leq 3^j$.

Consider each path P of length j. We claim that there are at most 3 possible choices for adding a $(j+1)$th node $P(j+1)$ to P to create a minimal path P' of length $j+1$. Let $P(j-1)$ and $P(j)$ denote the $(j-1)$th node and jth node of P, respectively.

Suppose first that $P(j)$ is is a boundary node of $\mathcal{G}_{tri}(\ell, \ell)$ (i.e. it is on row 1 or row ℓ or column 1 or column ℓ). Then $P(j)$ has degree at most 4, and one of the 4 nodes adjacent to $P(j)$ is $P(j-1)$, so there are at most 3 possible choices for $P(j+1)$, as required.

Now suppose that $P(j)$ is an internal node of $\mathcal{G}_{tri}(\ell, \ell)$. Then $P(j)$ has degree 6, and one of the 6 nodes adjacent to $P(j)$ is $P(j-1)$. Hence there are at most 5 possibilities for $P(j+1)$. But it is easy to verify that 2 of those 5 adjacent nodes of $P(j)$ must also be adjacent to $P(j-1)$. Hence, neither of these 2 nodes can be chosen as $P(j+1)$ since the resulting path P' will not be minimal (indeed, if $P(j+1)$ is chosen adjacent to $P(j-1)$ then internal node $P(j)$ could be removed from P' without disconnecting it). So there are at most 3 possibilities for $P(j+1)$ to keep P' minimal.

We conclude that any minimal path P of length j can be extended in at most 3 ways to a minimal path P' of length $j+1$. It follows that $N_i(j+1) \leq 3N_i(j) \leq 3^j$, which completes the inductive step. Since there are ℓ possible starting nodes on the rightmost column, we get $N_P(k, \ell) \leq \ell \cdot 3^k$, which proves $\mu \leq 3$.

We now show how to improve the connective constant upper bound to $\mu \leq 2.948$. This improvement is based on the fact that the bound $\mu \leq 3$ only takes into account a '1 edge history' of the path to restrict the number of possible 'next' nodes by ruling out those which destroy the path minimality due to 3 node cycles. By taking into account m-edge history for larger $m > 1$, we can improve the bound by also ruling out m'-cycles for $m' > 3$. Here we examine the case of $m = 4$ edge history, ruling out $m' = 6$ node cycles, as well as $m' = 3$ node cycles (see [6] for some results with even larger m).

Consider the 6 node cycle C_6 in graph $\mathcal{G}_{tri}(\ell, \ell)$ shown in Fig. 4(a). For any minimal path P of length $j \geq 4$ whose last 4 edges match a sequence of 4 successive edges along C_6 (in either clockwise or anticlockwise sense, such as the 4 edges between nodes $P(j-4), P(j-3), P(j-2), P(j-1), P(j)$ in Fig. 4(a)), we have at most 2 possibilities (labelled n_1, n_2 in Fig. 4(a)) for choosing a $(j+1)$th node $P(j+1)$ to extend P to a minimal path P' of length $j+1$. This is because by minimality, only 3 possiblities are allowed for $P(j+1)$ to rule out 3-node cycles in P' (as shown above), and out of those 3 nodes, one (labelled n^* in Fig 4(a)) can be eliminated to rule out the 6-cycle C_6 from being contained in P'. This reduction from 3 to 2 possibilities for $P(j+1)$ when the last 4 edges of P match a sequence from C_6 will give us the improved upper bound on μ.

To analyse this improvement, let $S(j)$ denote the set of all minimal paths P in $\mathcal{G}_{tri}(\ell, \ell)$ of length j starting at the ith node on the rightmost column of $\mathcal{G}_{tri}(\ell, \ell)$. We partition $S(j)$ into 4 disjoint subsets $S_1(j), \ldots, S_4(j)$ according to the number of matches of the 4 last edges of P with a sequence of successive edges on C_6, namely:

- $S_4(j)$ denotes the subset of paths in $S(j)$ whose 4 last edges match a sequence of 4 successive edges along C_6 (in either clockwise or anticlockwise sense).
- For $k = 3, 2, 1$, $S_k(j)$ denotes the subset of paths in $S(j)$ which are not in $S_{k+1}(j)$, but whose k last edges match a sequence of k successive edges along C_6 (in either clockwise or anticlockwise sense).

For $j \geq 5$ and $k \in \{1, 2, 3, 4\}$, we say that a minimal path P of length j is in state k if $P \in S_k(j)$. We can now construct a finite state machine M whose state transition function specifies for each minimal path P of length j in state k, the possible 'next' state k' of a minimal path P' of length $j + 1$ formed by adding a $(j + 1)$th node to P. The state transition diagram of M is shown in Fig 4(b), where a label b on a transition from state k to k' indicates that there are b possibilities for the $(j + 1)$th node which lead to this state transition. For example, as shown in Fig 4(a), if P is in state 4, then there are 2 possibilities for node $P(j + 1)$: one (node labelled n_1) leads to a transition to state 1 (since no two successive edges in C_6 are in the same column), the other (node labelled n_2) leads to a transition to state 2 (since no three successive edges in C_6 are in the order 'horizontal, vertical, horizontal'). It is easy to verify that the same transition rule from state 4 holds for all paths P in state 4 (i.e. regardless of the particular sequence of 4 successive edges along C_6 which form the last 4 edges of P). The transition rules for the other three states are also easy to verify.

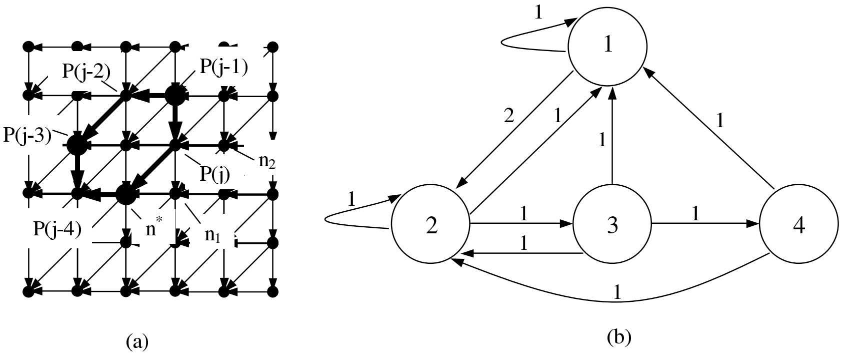

Fig. 4. (a) The 6 node cycle C_6 in $\mathcal{G}_{tri}(\ell, \ell)$ is shown in heavy black. (b) The state transition diagram of finite state machine M.

For $j \geq 5$ and $k \in \{1, 2, 3, 4\}$ let $N_k(j)$ denote the number of minimal paths (starting at ith node of the rightmost column of $\mathcal{G}_{tri}(\ell, \ell)$) of length j in state k. From the labelled state transition diagram of M in Fig 4(b), we immediately obtain the following recursive bound:

$$\begin{pmatrix} N_1(j+1) \\ N_2(j+1) \\ N_3(j+1) \\ N_4(j+1) \end{pmatrix} \leq A_M \cdot \begin{pmatrix} N_1(j) \\ N_2(j) \\ N_3(j) \\ N_4(j) \end{pmatrix}, \text{ where } A_M = \begin{pmatrix} 1\,1\,1\,1 \\ 2\,1\,1\,1 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \end{pmatrix}. \tag{2}$$

It follows from (2) that the vector $\mathbf{N}(j) \overset{\text{def}}{=} [N_1(j)\ N_2(j)\ N_3(j)\ N_4(j)]^T$ satisfies

$$\mathbf{N}(j) \leq A_M^{j-5}\mathbf{N}(5) \tag{3}$$

for $j \geq 5$. The matrix A_M can be diagonalised into the form $A_M = Q \cdot D \cdot Q^{-1}$, where Q is a 4×4 invertible matrix having the eigenvectors of A_M as its columns, and D is a 4×4 diagonal matrix having the 4 eigenvalues $\lambda_1, \ldots, \lambda_4$ of A_M on the diagonal. Note that $A_M^{j-5} = Q \cdot D^{j-5} \cdot Q^{-1}$, and D^{j-5} is a diagonal matrix with diagonal elements λ_k^{j-5} for $k = 1, \ldots, 4$. Plugging into (3) and adding up the components of $\mathbf{N}(j)$, we get the following upper bound on the number $N_P(j) = N_1(j) + \cdots N_4(j)$ of minimal paths of length j, starting at the ith node in the rightmost column of $\mathcal{G}_{tri}(\ell, \ell)$:

$$N_P(j) \leq c_1\lambda_1^{j-5} + c_2\lambda_2^{j-5} + c_3\lambda_3^{j-5} + c_4\lambda_4^{j-5}, \tag{4}$$

where the constants $c_1, \ldots, c_4$ are determined from (3) by $\mathbf{N}(5)$ and the eigenvector matrix Q. It follows that $N_P(j) = O(\lambda^j)$, where $\lambda \overset{\text{def}}{=} \max_k |\lambda_k|$ is the largest eigenvalue magnitude of A_M. Numerical computation shows that $\lambda \leq 2.948$, and hence (considering the ℓ possible starting nodes on the rightmost column of $\mathcal{G}_{tri}(\ell, \ell)$), the claimed bound $N_P(k, \ell) \leq c(\mu) \cdot \ell \cdot \mu^k$ with $\mu = \lambda \leq 2.948$ follows, for some constant $c(\mu)$. □

Remark 7. Our terminology *connective constant* for μ comes from similar (although not identical) constants defined in combinatorial studies of the 'self avoiding walk' in a lattice [14, 16]. However, the particular connective constant μ which arises in our work seems to not have been previously studied.

Remark 8. We have done some preliminary numerical eigenvalue computations using MATLAB with larger values of the 'edge history' parameter m on the path, extending our method for proving Lemma 7 (refer to [6] for more details). Using $m = 8$ we obtained the improved bound $\mu \leq 2.913$, although we are not yet certain about the accuracy of these MATLAB computations. We believe the efficient techniques from [14, 16] can be useful to further improve our numerical computed upper bound on μ by using even larger values of the 'edge history' on the path. Also, our method of bounding μ does not take into account the restriction that the paths of length k are right-left paths, so further improvements might result by taking this restriction into account.

Now we are ready to prove the following result.

Theorem 3. *Let $\mu, c(\mu)$ denote the connective constants of $\mathcal{G}_{tri}(\ell, \ell)$ (see Lemma 7). For any real constant $R > \mu$, if $t \leq n/R$, there exists a Weakly t-Reliable n-Colouring for graph $\mathcal{G}_{tri}(\ell, \ell)$ for some $\ell = O(n)$. Moreover, for any constant $\delta > 0$, the probability p that the random n-Colouring C_{rand} is not Weakly t-Reliable is upper bounded by δ if we choose*

$$\ell \geq b \cdot \frac{\log \binom{n}{t}}{\log(R/\mu)},$$

for a constant b satisfying

$$b - \left(\frac{3}{\log R}\right) \log b \geq 1 + \frac{\log\left(2c(\mu)\delta^{-1}\left(\frac{\log R}{\log(R/\mu)}\right)^3\right)}{\log R}. \tag{5}$$

Proof. Fix a t-colour subset I. We upper bound the probability $p(I)$, that if all ℓ^2 node colours of $\mathcal{G}_{tri}(\ell, \ell)$ are chosen uniformly and independently at random from $[n]$, the colouring C_{rand} is not Weakly t-Reliable, i.e. either an I-avoiding top-bottom path P_x doesn't exist, or an I-avoiding right-left path P_y doesn't exist.

Suppose that for a given colouring C, an I-avoiding top-bottom path P_x doesn't exist. This implies that the set $S(C)$ of graph nodes with colours in I must form a *top-bottom cutset*, which is defined as follows.

Definition 9 (Cutset/Minimal Cutset). *A set of nodes S in $\mathcal{G}_{tri}(\ell, \ell)$ is called a* top-bottom cutset *(resp.* right-left cutset*) if all top-bottom paths (resp. right-left paths) in $\mathcal{G}_{tri}(\ell, \ell)$ pass via a node in S. A cutset S is called* minimal *if removing any node from S destroys the cutset property.*

Note that the top-bottom cutset $S(C)$ must contain a minimal top-bottom cutset. The following intuitively obvious lemma shows that in order to count the minimal top-bottom cutsets of $\mathcal{G}_{tri}(\ell, \ell)$ it is enough to look at all minimal right-left paths in $\mathcal{G}_{tri}(\ell, \ell)$. Its formal proof can be found in [6].

Lemma 8 (Minimal Cutsets are Minimal Paths). *A set of nodes S in $\mathcal{G}_{tri}(\ell, \ell)$ is a minimal top-bottom cutset (resp. right-left cutset) if and only if it is a minimal right-left path (resp. top-bottom path).*

By Lemma 8, we conclude that if an I-avoiding top-bottom path doesn't exist for a colouring C then $S(C)$ contains a minimal right-left path $P_{c,x}$. Since $P_{c,x}$ is a subset of $S(C)$, its nodes only have colours in I. So, over the random choice of colouring C_{rand}, the probability that an I-avoiding top-bottom path doesn't exist is equal to the probability $p_x(I)$ that there exists a minimal right-left path $P_{c,x}$ whose node colours are all in t-collusion I.

Let $N_P(k, \ell)$ denote the total number of minimal right-left paths in $\mathcal{G}_{tri}(\ell, \ell)$ of length k. Since node colours are chosen independently and uniformly in $[n]$, each such path has probability $(t/n)^k$ to have all its node colours in I. It is clear that $\ell \leq k \leq \ell^2$. So, summing over all possible path lengths, we get the following upper bound: $p_x(I) \leq \sum_{k=\ell}^{\ell^2} N_P(k, \ell)(t/n)^k$. By symmetry, a similar argument gives the same upper bound on the probability $p_y(I)$ that a right-left I-avoiding path P_y does not exist. So we get the following upper bound on the probability $p(I)$ that either I-avoiding top-bottom path doesn't exist or an I-avoiding right-left path doesn't exist for each fixed t-subset I: $p(I) \leq 2\sum_{k=\ell}^{\ell^2} N_P(k,\ell)(t/n)^k$. Finally, taking a union bound over all $\binom{n}{t}$ possible t-colour subsets I, we get an upper bound on the probability p that the colouring C_{rand} is not Weakly t-Reliable of the form $p \leq 2\sum_{k=\ell}^{\ell^2} N_P(k,\ell)(t/n)^k\binom{n}{t}$. Using the bound on $N_P(k,\ell)$ from Lemma 7, we get

$$p \leq 2c(\mu)\ell^3(\mu t/n)^\ell \binom{n}{t}. \tag{6}$$

Since $n/t \geq R > \mu$, it is clear that this upper bound on p is less than 1 for sufficiently large ℓ. In fact, it suffices to take $\ell = O(\log(\binom{n}{t})/\log(n/(\mu t))) = O(n)$, as claimed. Now suppose we fix $\delta > 0$ and we want to find a lower bound on ℓ such that the error probability $p \leq \delta$. From (6) and using $n/t \geq R$ we see that $p \leq \delta$ is satisfied as long as

$$\ell \log(R/\mu) - 3\log(\ell) \geq \log(2c(\mu)N\delta^{-1}), \tag{7}$$

where $N = \binom{n}{t}$. Take $\ell = b\log(N)/\log(R/\mu)$. Plugging this choice of ℓ into (7), and using the fact that $N \geq \binom{\lceil R \rceil}{1} \geq R$ for all $n \geq R$ (since $N = \binom{n}{n/R}$ increases monotonically with n), we conclude that (7) is satisfied if the constant b is sufficiently large such that (5) holds. This completes the proof. □

Combining Theorem 3 (applied with $n' = R \cdot t \leq n$ colours from $[n]$ for constant $R > \mu$) with Lemmas 3, 4, 6 and 7, we have

Corollary 2. *For any constant $R > 2.948$, if $t \leq n/R$, there exists a black box t-private protocol for f_G with communication complexity $O(nt^2)$ group elements. Moreover, for any $\delta > 0$, we can construct a probabilistic algorithm, with run-time polynomial in n and $\log(\delta^{-1})$, which outputs a protocol Π for f_G such that the communication complexity of Π is $O(nt^2\log^2(\delta^{-1}))$ group elements and the probability that Π is not t-private is at most δ.*

Remark 9. Our computational experiments indicate that $t > n/2.948$ can be achieved with moderate values of ℓ – for example, for $n = 24$, $t = 11$ (i.e. $t \approx n/2.182$), we found a t-Reliable n-Colouring of $\mathcal{G}_{tri}(\ell,\ell)$ with $\ell = 350$, which is much smaller than $\binom{n}{t} \approx 2.5 \cdot 10^6$.

4.5 Generalisations and Other Results

General functions over G. Some applications may require n-party computation of more general functions over G (using only the group operation) instead of f_G. The most general such function is of the form $f'_G(x_1,\ldots,x_m) = x_1\ldots,x_m$, where $m \geq n$ and each of the n parties holds one or more x_i's. Our reduction from Section 4.2 (and hence all our protocols) trivially extends to this most general case in the natural way.

General adversary structures. One may also consider more general adversary structures in place of the t-threshold structure. With the exception of our second construction in Section 4.4, all other results in the paper trivially generalise to the case of a Q^2 adversary structure $\mathcal{A}$, in which no pairwise union of collusions in $\mathcal{A}$ covers all n parties [11]. In particular, the generalisation of the first construction in Section 4.4 has communication complexity $O(n|\mathcal{A}|^2)$ group elements.

More efficient protocols for small t. For the cases $t \in \{1, 2\}$, we have managed to design explicit t-private black-box protocols for f_G with linear communication complexity ($O(n)$ group elements) and optimal collusion resistance. These protocols and their analysis can be found in [6]. We have also implemented a computer program for finding t-Reliable n-Colourings of a given graph, with which one can easily construct efficient protocols for small values of n, t (avoiding the error probability δ of Theorem 3).

5 Conclusions

We showed how to design black-box t-private protocols for computing the n-product function over any finite group by reducing the problem to a combinatorial graph colouring problem, using tools from communication security [7]. Our work raises some interesting combinatorial questions. For example, for our PDAG $\mathcal{G}_{tri}(\ell, \ell)$, what is the shape of the 'tradeoff' curve $R_{max}(\ell)$ relating the maximal achievable (using a suitable colouring) secure collusion resistance $R_{max} = t/n$ to the graph size ℓ? (we showed that $R_{max}(\ell) \geq 1/2.948$ for $\ell = O(t)$ and $R_{max}(\ell) = 1/2$ for $\ell \geq \binom{2t+1}{t}$). More generally, what is the largest collusion resistance achievable with an admissible PDAG of size polynomial in n, and what kind of PDAG achieves this optimum? There are also interesting cryptographic questions. First, can our black-box protocols be efficiently strengthened to yield black-box protocols secure against *active* adversaries? Second, can the communication complexity $O(nt^2)$ of our t-private protocols be reduced further? Third, does there exist an efficient (run-time polynomial in n) *deterministic* algorithm to generate a Weakly t-Reliable n-Colouring of $\mathcal{G}_{tri}(\ell, \ell)$ (or some other admissible PDAG) given n, t as input?

Acknowledgements. This research was supported by ARC research grants DP0451484, DP0558773, DP0663452 and DP0665035. Ron Steinfeld's research was supported in part by a Macquarie University Research Fellowship (MURF). Huaxiong Wang's research was supported in part by the Singapore Ministry of Education grant (T206B2204). Yvo Desmedt is grateful for the research visits to Macquarie University. The authors also thank Chris Charnes and Scott Contini for helpful discussions about this work.

References

1. N. Alon and J. Spencer. *The Probabilistic Method.* Wiley-Interscience, 2000.
2. J. Bar-Ilan and D. Beaver. Non-Cryptographic Fault-Tolerant Computing in a Constant Number of Rounds of Interaction. In *Symposium on Principles Of Distributed Computing (PODC)*, pages 201–209, New York, 1989. ACM.

3. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation. In *Proc. 20-th STOC*, pages 1–10. ACM, 1988.
4. D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *Proceedings of the twentieth annual ACM Symp. Theory of Computing, STOC*, pages 11–19, May 2–4, 1988.
5. R. Cramer, S. Fehr, Y. Ishai, and E. Kushilevitz. Efficient Multi-Party Computation Over Rings. In *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 596–613, Berlin, 2003. Springer-Verlag.
6. Y. Desmedt, J. Pieprzyk, R. Steinfeld, and H. Wang. On Secure Multi-Party Computation in Black-Box Groups. Full version of this paper, 2007. Available at `http://www.comp.mq.edu.au/∼rons/`.
7. Y. Desmedt, Y. Wang, and M. Burmester. A Complete Characterization of Tolerable Adversary Structures for Secure Point-to-Point Transmissions. In *ISAAC 2005*, volume 3827 of *LNCS*, pages 277–287, Berlin, 2005. Springer-Verlag.
8. W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Trans. on Information Theory*, 22:644–654, 1976.
9. T. ElGamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Tran. Info. Theory*, IT-31(4):469–472, 1985.
10. O. Goldreich. *Foundations of Cryptography, Volume II.* Cambridge University Press, Cambridge, 2004.
11. M. Hirt and U. Maurer. Complete Characterization of Adversaries Tolerable in Secure Multi-Party Computation (Extended Abstract). In *Symposium on Principles Of Distributed Computing (PODC)*, pages 25–34, New York, 1997. ACM.
12. E. Kushilevitz. Privacy and Communication Complexity. *SIAM J. on Discrete Mathematics*, 5(2):273–284, 1992.
13. S. Magliveras, D. Stinson, and T. van Trung. New approaches to Designing Public Key Cryptosystems using One-Way Functions and Trapdoors in Finite Groups. *Journal of Cryptology*, 15:285–297, 2002.
14. J. Noonan. New Upper Bounds for the Connective Constants of Self-Avoiding Walks. *Journal of Statistical Physics*, 91(5/6):871–888, 1998.
15. S. Paeng, K. Ha, J. Kim, S. Chee, and C. Park. New Public Key Cryptosystem Using Finite Non Abelian Groups. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 470–485, Berlin, 2001. Springer-Verlag.
16. A. Pönitz and P. Tittmann. Improved Upper Bounds for Self-Avoiding Walks in $\mathbb{Z}^d$. *The Electronic Journal of Combinatorics*, 7, 2000.
17. R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–128, 1978.
18. A. Shamir. How To Share a Secret. *Communications of the ACM*, 22:612–613, 1979.
19. P. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comp.*, 26(5):1484–1509, 1997.