Topology-Hiding Communication from Minimal Assumptions

Marshall Ball[1], Elette Boyle[2], Ran Cohen[3], Lisa Kohl[4], Tal Malkin[5], Pierre Meyer[6], and Tal Moran[7]

[1] Columbia University, marshall@cs.columbia.edu
[2] IDC Herzliya, elette.boyle@idc.ac.il
[3] Northeastern University, rancohen@ccs.neu.edu
[4] Technion, lisa.kohl@cs.technion.ac.il
[5] Columbia University, tal@cs.columbia.edu
[6] IDC Herzliya and École Normale Supérieure de Lyon, pierre.meyer@ens-lyon.fr
[7] Spacemesh, Northeastern University and IDC Herzliya, talm@idc.ac.il

Abstract. *Topology-hiding broadcast* (THB) enables parties communicating over an incomplete network to broadcast messages while hiding the topology from within a given class of graphs. THB is a central tool underlying general *topology-hiding secure computation* (THC) (Moran et al. TCC'15). Although broadcast is a privacy-free task, it was recently shown that THB for certain graph classes necessitates computational assumptions, even in the semi-honest setting, and even given a single corrupted party.

In this work we investigate the minimal assumptions required for topology-hiding communication—both *Broadcast* or *Anonymous Broadcast* (where the broadcaster's identity is hidden). We develop new techniques that yield a variety of necessary and sufficient conditions for the feasibility of THB/THAB in different cryptographic settings: information theoretic, given existence of key agreement, and given existence of oblivious transfer. Our results show that feasibility can depend on various properties of the graph class, such as *connectivity*, and highlight the role of different properties of topology when kept hidden, including *direction*, *distance*, and/or *distance-of-neighbors* to the broadcaster.

An interesting corollary of our results is a dichotomy for THC with a public number of at least three parties, secure against one corruption: information-theoretic feasibility if all graphs are 2-connected; necessity and sufficiency of key agreement otherwise.

1 Introduction

Reliable communication between a set of mutually distrustful parties lies at the core of virtually any distributed protocol, ranging from consensus tasks [19, 15] to secure multiparty computation [21, 11, 5, 9]. Classical protocols from the '80s considered complete communication graphs between the parties, where each pair of parties is connected by a communication channel. However, in many real-life

scenarios the parties are not pairwise connected; this raises the need for distributed interactive computations, and in particular communication protocols, over an incomplete graph. Often, the *network topology itself* may be sensitive information that should not be revealed by the protocol.

Topology-hiding broadcast. With this motivation, Moran et al. [18] formalized the concept of *topology-hiding computation (THC)*. Here, the goal is to allow parties who see only their immediate neighborhood (and possibly know that the graph belongs to some class), to securely compute arbitrary functions without revealing any additional information about the graph topology other than the output (computations on the graphs, e.g., establishing routing tables, are also supported). THC is of theoretical interest, but is also motivated by real-world settings where it is desired to keep the underlying communication graph private. These include social networks, ISP networks, ad hoc (or mesh) networks, vehicle-to-vehicle communications, and possible approaches for contact tracing.

Given the existence of general MPC protocols, achieving THC for arbitrary functions hinges on communicating in a topology-hiding way, rather than on keeping inputs private. In particular, a core bottleneck for achieving general THC is the special case of *topology-hiding broadcast (THB)*, where a designated party (the broadcaster) reliably sends its message to all other parties. Indeed, given an MPC protocol for a function f defined in the broadcast model (where *all* communication is sent via a broadcast channel, possibly encrypted),[8] the parties can replace the broadcast channel by a THB protocol to obtain a THC protocol for the function f.

Although broadcast is a privacy-free task, realizing THB turns out to be challenging, even in the semi-honest setting where all parties follow the protocol. This is in stark contrast to standard (topology-revealing) broadcast, which is trivially achievable in the semi-honest setting, e.g., simply "flooding" the network, forwarding received messages. For general semi-honest corruptions, the best THB constructions follow from a series of works [18, 12, 1, 2, 16], culminating in THB (as well as THC) protocols for all graphs. However, even for THB, all known protocols require structured public-key cryptographic assumptions, such as QR, DDH, or LWE.[9] The use of strong assumptions was justified by Ball et al. [3] who showed that without an honest majority, even THB implies oblivious transfer (OT).[10]

A central paradigm in standard (topology-revealing) secure computation is to exchange *cryptographic* assumptions with an *honest-majority* assumption [5, 9, 20]. A recent work of Ball et al. [4] asked whether such a paradigm can be applied in the topology-hiding realm. The results of [4] demonstrated that answering this

[8] Such protocols exist in the honest-majority setting assuming key agreement, and thus under this assumption, THB implies THC. In the information-theoretic setting THC can be strictly stronger, as we will see.

[9] That is, the *Quadratic Residuosity* assumption, the *Decisional Diffie-Hellman* assumption, and the *Learning With Errors* assumption, respectively.

[10] The lower bound of [3] holds for 4-party 2-secure THB with respect to a small class of 4-node graphs, namely, a square, and a square with any of its edges removed.

question is more subtle than meets the eye, even when considering the basic case of *one semi-honest corruption*. On the one hand, they showed that information-theoretic THB (IT-THB) can be achieved for the graph class of *cycles*, where the protocol hides the ordering of parties within the cycle. On the other hand, they identified that THB for *paths* of $n \geq 4$ nodes (again hiding ordering) implies key agreement.

This work. In a sense, [4] unveiled the tip of the iceberg, revealing a range of questions: Which aspects of the topology can be hidden information theoretically, and which require cryptographic hardness? Is key agreement sufficient for 1-corruption THB, or are there graph classes that require stronger assumptions? In this paper we study the cryptographic power of THB. The main question that we ask is:

What are the minimal cryptographic assumptions required for THB for a given class of graphs?

We focus on a minimal setting, with a small number of parties and a single, or few, semi-honest corruptions, which we denote by t-THB for t corruptions. This makes our lower bounds stronger; and, as we demonstrate, even this simple setting offers a rich multi-layered terrain, and provides insights and implications for more general settings (including THC).

Before proceeding to state our results, we note that prior THB protocols actually achieved the stronger property of *topology-hiding anonymous broadcast* (THAB), where the identity of the broadcaster remains hidden [7, 8]. From the definitions of these primitives, we have that

$$\mathsf{THC} \implies \mathsf{THAB} \implies \mathsf{THB}.$$

Thus, all lower bounds for THB (such as the one from [4] and our own results) apply also for THAB and THC. As we will show, there are classes of graphs where THB is possible information theoretically, but THAB, and thus THC, require strong cryptographic assumptions. Understanding for which topologies the reverse implications hold is addressed here in part, but the full answer remains an interesting open question.

1.1 Our Results

This work makes significant strides in mapping the landscape of THB, THAB, and THC in minimal settings, in the process developing new techniques that may be useful to achieve a full understanding of its complexities. As standard in the THC literature, we consider a synchronous setting, where the protocol proceeds in rounds.[11]

[11] LaVigne et al. [17] recently studied THC in a non-synchronous setting, demonstrating many barriers.

New Lower Bounds and Techniques

- *THB.* We explore which properties of graph topology are "hard" to hide, in the sense of requiring cryptographic assumptions to do so. We show that hiding any one of the properties of *direction*, *distance*, and/or *distance-of-neighbors* to the broadcaster is hard—while revealing all three but nothing else (in fact, only revealing distance-of-neighbors) can always be achieved information theocratically, using the trivial flooding protocol.
- *THAB.* We observe that t-THAB for any graph class containing a graph that is not $(t+1)$-connected[12] implies *key agreement*. We further show that hiding the *number of participants* in certain graph classes implies *infinitely often oblivious transfer*, even for 1-THAB.

Unconditional & KA-Based Upper Bounds

- *Unconditional.* We provide a construction of 1-THAB for *all 2-connected graphs*, whose complexity grows with the number of potential graphs in the class (in particular, it is efficient for constant-size graphs), which achieves *statistical* information-theoretic security.
- *Key Agreement.* Assuming the existence of *key agreement*, we achieve 1-THB for *all graphs*, and 1-THAB for all graphs of ≥ 3 nodes.

Corollaries and Conclusions

- *Dichotomy for 1-THC with ≥ 3 parties.* An interesting corollary of our results is a dichotomy for 1-THC with a fixed and known set of at least three parties[13] (i.e., where all graphs share the same vertex set): if all graphs in a class are 2-connected, the class supports information-theoretic 1-THC; otherwise, key agreement is necessary and sufficient for 1-THC.
- *Dichotomy for 1-THAB with ≥ 3 parties.* A similar result holds for 1-THAB for a dynamic set of parties (i.e., the vertex set of every graph is a *subset* of $[n]$) as long as each graph contains at least three nodes: if all graphs in a class are 2-connected, the class supports information-theoretic 1-THAB; otherwise, key agreement is necessary and sufficient for 1-THAB.
- *Characterization of 1-THB for small graphs.* Our results introduce several new constructions and analysis techniques; as a demonstration of their wider applicability, we provide a characterization of the more complex case of 1-THB for all graph classes on four nodes or fewer. Note that the feasibility boundaries of 1-THB are more complex than 1-THAB since, as we show, certain lower bounds for 1-THAB do not apply to 1-THB.
- *THB without OT.* Our upper bounds constitute the first protocols using machinery "below" oblivious transfer,[14] aside from the specific graph class of cycles of fixed length (that was shown in [4]).

[12] A graph is k-connected if and only if every pair of nodes is connected by k *vertex-disjoint* paths.

[13] If the class of graphs contains a 2-path, then oblivious transfer is necessary for secure computation [14].

[14] Note that OT is strictly stronger than KA in terms of black-box reductions, since OT implies KA in a black-box way, but the converse does not hold [10].

We next describe these results in more detail.

Lower Bounds We begin by investigating the conditions under which THB and THAB for a graph class $\mathcal{G}$ *necessitate* cryptographic assumptions.

THB: Hiding direction, distance, or distance-of-neighbors. Recall that restricting attention to a class of graphs $\mathcal{G}$ captures that a THB protocol hides *partial* information about a graph topology. For example, if all graphs in $\mathcal{G}$ have property P, then the THB protocol need not hide whether P is satisfied when providing indistinguishability within this class. Our question thus becomes: for which properties of a graph topology is it the case that hiding necessitates cryptography?

Consider as a baseline the trivial "flooding" protocol, which in general is *not* topology hiding. Parties flood the network: on receiving the broadcast message, a party forwards it to all neighbors from which it was not previously received. Indeed, this protocol reveals information; e.g., the round number in which a party first receives the message corresponds directly to its distance from the broadcaster. However, even for this simple protocol, the amount of information revealed is limited. The leakage can be quantified precisely: each party learns exactly the distance from the broadcaster of each of its neighbors,[15] or *"distance-of-neighbors."* In particular, this includes the information of (a) *direction* of the broadcaster (i.e., which neighbors are on a shortest path to the broadcaster), and (b) *distance* to the broadcaster. Since the flooding protocol can be executed unconditionally for any graph class $\mathcal{G}$, it can only be some combination of this leaked distance-of-neighbors information for which hiding requires cryptography.

Examining the lower bound of [4], we observe that it constitutes an example where hiding the *direction* of the broadcaster from a given party necessitates key agreement (KA). This is embodied via the class of two graphs $\mathcal{G}_{\mathsf{4\text{-}path}} = \{(A\text{-}B\text{-}\mathbf{C}\text{-}D), (B\text{-}\mathbf{C}\text{-}D\text{-}A)\}$ on a path, where party C is unaware whether the broadcasting party A lies to its left or right. Indeed, broadcaster direction is central to their lower bound, where KA agents Alice and Bob emulate the THB parties B and D, respectively, and jointly emulate C. Each flips a (private) coin to decide whether to also emulate A on their corresponding side. The two parties can detect cases where both (or neither) party decided to emulate A. In the remaining cases both parties agree on which side the broadcaster appears: this will serve as the secret common key bit.

At a high level, the security of this KA protocol relies on the fact that the eavesdropper's view is essentially that of party C—who, by topology hiding, cannot distinguish the relative direction of A. Thus, one may naturally ask whether hiding the *direction* to the broadcaster captures the essence of the cryptographic power of THB.

[15] If the neighbor sends the message in the first round that the party learns it, then its distance is one less of the party's distance. If the neighbor sends after the party learned it, then its distance equals the party's distance. If the neighbor does not send, then its distance is one more than the party's distance.

Our first result shows that the direction to the broadcaster is not the complete answer. We present a class of graphs $\mathcal{G}_{\text{oriented-5-path}}$ for which any constant-round 1-secure THB implies *infinitely often key agreement*,[16] but for which the direction to the broadcaster is always known. Specifically, we consider the class of 5-path graphs where the broadcaster A is always on the left,[17] i.e.,

$$\mathcal{G}_{\text{oriented-5-path}} = \Big\{(A\text{-}B\text{-}C\text{-}D\text{-}E), (A\text{-}E\text{-}B\text{-}C\text{-}D), (A\text{-}D\text{-}E\text{-}B\text{-}C), (A\text{-}C\text{-}D\text{-}E\text{-}B)\Big\}.$$

Because of this structure, the lower-bound techniques of Ball et al. [4] do not apply. Proving a key-agreement implication for $\mathcal{G}_{\text{oriented-5-path}}$ requires a new, more subtle approach, which we discuss in Section 2. In particular, unlike [4], we must leverage the fact that topology hiding holds for *any* choice of corrupted party. For example, party C cannot distinguish between $(A\text{-}B\text{-}C\text{-}D\text{-}E)$ and $(A\text{-}E\text{-}B\text{-}C\text{-}D)$, and party B cannot distinguish between $(A\text{-}E\text{-}B\text{-}C\text{-}D)$ and $(A\text{-}D\text{-}E\text{-}B\text{-}C)$.

Taking a broader view of this example, we observe that while the *direction* of the broadcaster is public for $\mathcal{G}_{\text{oriented-5-path}}$, the information to be hidden corresponds directly to the *distance* of the given parties to the broadcaster. One may thus once again wonder whether revealing both the *direction and distance* to the broadcaster dictates unconditional THB feasibility.

Our second result reveals that the answer is even more intricate. We demonstrate a class of graphs for which each party publicly knows *both its direction and distance* to the broadcaster, but for which 1-THB still implies key agreement.

Specifically, we consider the class $\mathcal{G}_{\text{triangle}}$ consisting of a triangle, with possibly one of its edges missing (see Figure 1). Interestingly, this is a very basic communication pattern: if a party has two neighbors it does not know if its neighbors are directly connected or not, but a party with one neighbor knows the entire topology. Notably, direction and distance from the broadcaster are both clearly identifiable to each party given just its neighbor set; the only information hidden from a party is *its neighbor's* distance to the broadcaster. We show that this is enough to imply KA (see Section 2 for details).

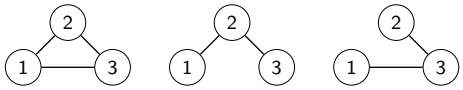


Fig. 1: The class $\mathcal{G}_{\text{triangle}}$.

To summarize, for each strict subset of the properties that are leaked by the flooding protocol (namely, *direction* and/or *distance* to the broadcaster), we

[16] An infinitely often key agreement guarantees correctness and security for infinitely many $\lambda \in \mathbb{N}$ (where λ stands for the security parameter).

[17] In particular, the "left/right" orientation can be deduced locally from each node's neighbor set.

demonstrate a graph class for which hiding only these properties implies public-key cryptographic assumptions. Complementarily, if all three properties (essentially, just the distance-of-neighbors) are known then one can use the flooding protocol to obtain THB information theoretically.

Theorem 1 (THB lower bounds, informal). *We consider THB with 1 semi-honest corruption.*

- *1-secure THB for the graph class $\mathcal{G}_{\text{oriented-5-path}}$ of 5-path graphs for which the broadcasting party is always in the leftmost direction (see above) implies infinitely often key agreement.*
- *1-secure THB for the graph class $\mathcal{G}_{\text{triangle}}$ (Figure 1), for which the broadcasting party is always at a known distance* and *direction, implies key agreement.*

In contrast, for any class $\mathcal{G}$ such that for every party the distance of each of its neighbors to the broadcaster is fixed and known across all graphs, there exists an unconditionally 1-secure THB protocol.

THAB: Key Agreement and Beyond. We next turn to topology-hiding *anonymous* broadcast (THAB). As mentioned above, any lower bound for THB is also a lower bound for THAB; however, we show even stronger results for THAB.

The connection between anonymous communication and cryptographic hardness was previously studied by Ishai et al. [13]. They showed that in a communication network that provides *sender-anonymity* (under relatively strong adversarial observation), key agreement exists unconditionally; i.e., each pair of parties within the system can agree on a secret key. Our setting is slightly different, however, using the lower-bound technique from [4] a similar observation can be made: sender-anonymous communication over a path of three nodes implies the existence of standard Alice-Bob key agreement, where the eavesdropper can see which party sends which message.

This clear-cut impossibility of information-theoretic 1-THAB (in fact, 1-secure anonymous broadcast) on arbitrary incomplete networks stands in contrast to 1-THB, where the determination of when a graph class yields an implication to key agreement was demonstrably complex. Concretely, consider the following (singleton) class $\mathcal{G}_{\{\text{a-b-c}\}}$:

$$\mathcal{G}_{\{\text{a-b-c}\}} = \{(A\text{-}B\text{-}C)\}\,.$$

THB for this class is glaringly trivial (indeed, there is no information to hide because the topology is fixed); however, as discussed, 1-THAB on this class implies key agreement. For completeness, in Section 5.1 we prove this implication as a direct corollary of the key-agreement lower bound of Ball et al. [4], where the "direction" of the broadcaster (either A or C) in this case is hidden from the intermediate party B by anonymity.

At this moment, the reader may pause, ensnared in the underwhelming nature of the above class $\mathcal{G}_{\{\text{a-b-c}\}}$. However, by a standard player-partitioning argument ("projecting" a larger graph down onto the 3-path), the above result yields a much broader statement.

Proposition 1 (THAB lower bound 1, informal, [13, 4]). *Let $\mathcal{G}$ be a class of graphs that contains a graph with at least $(t+2)$ nodes that is not $(t+1)$-connected. Then t-secure THAB for $\mathcal{G}$ implies KA.*

In our final lower-bound result, we demonstrate an even more extreme form of separation between THB and THAB. We consider the graph class $\mathcal{G}_{\text{2-vs-3}}$ that consists of all possible 2-path and 3-path graphs over three parties, i.e.,

$$\mathcal{G}_{\text{2-vs-3}} = \{(A\text{-}B), (A\text{-}C), (B\text{-}C), (A\text{-}B\text{-}C), (B\text{-}C\text{-}A), (C\text{-}A\text{-}B)\}\,.$$

In this class, for example, if A is only connected to B, it does not know whether B has a second neighbor or not. It is easy to see that 1-secure THB exists unconditionally (by the flooding protocol); however, we show that 1-secure THAB implies *infinitely often oblivious transfer*.[18] We emphasize that as opposed to other classes of graphs discussed thus far, the "hardness" of the class $\mathcal{G}_{\text{2-vs-3}}$ is based on hiding the *number of nodes* participating in the protocol. We refer the reader to Sections 2 and 5.2 for further details on the lower bound.

Overall, we obtain the following theorem.

Theorem 2 (THAB lower bound 2, informal). *1-secure THAB for $\mathcal{G}_{\text{2-vs-3}}$ implies infinitely often OT.*

We remark that these results separate THB from THAB for very simple graph classes, where THAB requires computational assumptions whereas unconditional THB exists via the trivial flooding protocol. Later, in Section 1.1 we will show a more interesting separation via the "butterfly" graph, where the existence of information-theoretic THB itself is non-trivial.

Upper Bounds Before stating our results, we recall the state-of-the-art for semi-honest THB and THAB with one corruption. Assuming oblivious transfer (OT), 1-THAB can be obtained for all graphs following the construction approach of Moran et al. [18].[19] Without assuming OT, the only previously known nontrivial[20] construction of THB or THAB is the information-theoretic 1-THAB for the specific graph class of cycles on a known number of nodes in [4].

We consider three settings of upper bounds: (1) with *information-theoretic* security, (2) assuming only *key agreement*, and (3) converting generically from THB to THAB.

[18] An infinitely often OT protocol guarantees correctness and security for infinitely many $\lambda \in \mathbb{N}$ (where λ stands for the security parameter).

[19] The result of [18] was limited to graphs of small diameter to allow an arbitrary number of corruptions. With a single corruption the same construction can support all graphs.

[20] THB exists trivially for any graph class in which each party's neighborhood uniquely identifies the graph topology.

Information-theoretic security. First, we consider protocols for achieving 1-THAB (and THB) in the *information-theoretic* setting, without cryptographic assumptions. Recall that the lower bound in Proposition 1 above rules out the possibility of 1-THAB for any graph class containing a graph that is not 2-connected. We show that conversely, if a class of graphs $\mathcal{G}$ contains only 2-connected graphs, then 1-THAB for $\mathcal{G}$ *is* feasible.

The protocol's communication grows polynomially in the *size* of the class $\mathcal{G}$ and its computation grows polynomially in the size of $\mathcal{G}$ and exponentially in the *maximal degree* of any $G \in \mathcal{G}$. However, our results are meaningful despite this caveat: First, the protocol is efficient when considering a constant number of parties (or appropriate graph classes of polynomial size). Second, since the protocol remains secure against computationally unbounded adversaries, it is still meaningful to consider protocols that are inefficient in the class.

Theorem 3 (1-IT-THAB for 2-connected, informal). *Let $\mathcal{G}$ be a class containing only 2-connected graphs. Then, there exists a statistical information-theoretic 1-THAB for $\mathcal{G}$ whose communication complexity is polynomial in the size of $\mathcal{G}$, and whose computation complexity is polynomial in the size of $\mathcal{G}$ and exponential in the maximal degree of $\mathcal{G}$.*

Combining Proposition 1 and Theorem 3 gives a characterization for information-theoretic 1-THAB: Namely, a protocol exists if and only if all graphs in the class are 2-connected (with the exception of the trivial class containing only the 2-path). For the case of 1-THB such dichotomy does not hold and, as we show, there exist graph classes with 1-connected graphs that still admit information-theoretic 1-THB protocols.

Remark 1 (1-IT-THB for $\mathcal{G}_{\mathsf{butterfly}}$). Consider the 5-node, 1-connected butterfly graph (Figure 2) and let $\mathcal{G}_{\mathsf{butterfly}}$ contain all permutations of the nodes on the graph (where parties' positions are permuted). In Section 6.2, we show that although the simple flooding protocol does *not* directly hide topology, there exists a (perfectly secure) information-theoretic 1-THB protocol for $\mathcal{G}_{\mathsf{butterfly}}$.

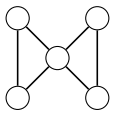


Fig. 2: The butterfly graph.

Upper bounds from KA. Recall that from the lower bounds presented above (see Section 1.1), key agreement is a necessary assumption for 1-THB and 1-THAB for many classes of graphs. This begs the question of when key agreement is also a *sufficient* assumption for 1-THB and 1-THAB. We show that assuming key agreement there exist 1-secure THB for all graphs, and 1-secure THAB for all graphs containing at least 3 nodes.

Theorem 4 (1-THAB and 1-THB from KA, informal).

- *Let $\mathcal{G}$ be a class consisting of graphs with at least three nodes. Assuming key agreement, there exist 1-THAB for $\mathcal{G}$.*
- *Let $\mathcal{G}$ be a class of graphs. Assuming key agreement, there exist 1-THB for $\mathcal{G}$.*

We note that in the first item of Theorem 4, removing the restriction of at least three nodes would require bypassing black-box separation results, due to Theorem 2 that asserts the necessity of (infinitely often) OT for the class $\mathcal{G}_{\text{2-vs-3}}$. On the other hand, by [18], assuming OT there exists 1-THAB for all graphs, essentially closing the gap in this regime.

THC dichotomy. Upon closer inspection, we observe that our upper bounds—both the information-theoretic protocols for 2-connected graphs, as well as the results from KA above—give something even stronger than 1-THAB: they give topology-hiding *secure message transmission*, i.e., emulating pairwise secure point-to-point channels. In this case, assuming that the number of parties is fixed and known across all graphs, we can run the semi-honest "BGW" protocol [5], which only requires pairwise secure channels and works for an honest majority. Thus, together with our lower bounds, we arrive at the following dichotomy for 1-THC:

Corollary 1 (1-secure THC dichotomy, informal). *Consider a class of graphs $\mathcal{G}$ on $n \geq 3$ nodes. Then, the following hold regarding existence of THC for $\mathcal{G}$ secure against 1 semi-honest corruption:*

- *If all graphs $G \in \mathcal{G}$ are 2-connected, then there exists a statistically information-theocratically secure, 1-THC protocol for $\mathcal{G}$, whose communication is polynomial in the size of $\mathcal{G}$ and whose computation is polynomial in the size of $\mathcal{G}$ and exponential in the maximal degree of $\mathcal{G}$.*
- *If there exists $G \in \mathcal{G}$ that is* not *2-connected, then KA is necessary and sufficient for 1-secure THC for $\mathcal{G}$.*

Generically converting THB to THAB. Our results have demonstrated a number of nontrivial separations between THB and THAB, identifying classes of t-connected graphs and computational assumptions which admit t-THB protocols but provably cannot obtain t-THAB. This includes, for example, $\mathcal{G}_{\{\text{a-b-c}\}}$ and $\mathcal{G}_{\text{butterfly}}$ for information theoretic vs. key agreement, as well as $\mathcal{G}_{\text{2-vs-3}}$ for information theoretic vs. oblivious transfer.

Finally, we show that graph connectivity is, indeed, a critical property for determining the relation between THB and THAB on a class of graphs. Specifically, we show that $(t+1)$-connectivity is a sufficient condition for *equivalence* of the two notions against t corruptions.

Theorem 5 (t-THB $\Rightarrow$ t-THAB given $(t+1)$-connectivity, informal). *Let $n \in \mathbb{N}$ and let $\mathcal{G}$ be a class consisting of $(t+1)$-connected graphs over n nodes. If there exists t-THB for $\mathcal{G}$ then there exists t-THAB for $\mathcal{G}$.*

Our reduction builds upon the "Dining Cryptographers" approach for anonymous broadcast due to Chaum [8]. Recall in THAB there exists a unique broadcaster who wishes to convey its input bit $x \in \{0,1\}$ to all parties without revealing its identity (or the topology). To do so, each party first additively secret shares its input—defined to be 0 for any non-broadcaster—across its neighbors, locally sums all received shares to $s_i \in \{0,1\}$, and then acts as broadcaster within the underlying (non-anonymous) THB with input value s_i. After this phase, all parties receive the vector of shares $(s_1, \ldots, s_n)$, which can be summed to yield the original input x. It was shown by [8] that if the graph is $(t+1)$-vertex connected (so as to ensure that the adversary cannot corrupt a vertex cut), then the protocol is anonymous. We observe that the protocol further preserves the *topology hiding* of the underlying THB protocol. Indeed, given $(t+1)$-connectivity, the vector of broadcasted shares $(s_1, \ldots, s_n)$ will be *uniform* conditioned on the necessary sum, independent of the graph structure.

Summary and Characterization of Graphs with up to Four Nodes We summarize our combined contributions in Table 1, together with relevant prior results.

	1-THB		1-THAB	
	sufficient	**necessary**	**sufficient**	**necessary**
IT	$\mathcal{G}_{\text{cycle}}$ [4] $\mathcal{G}_{\text{butterfly}}$ (Remark 1) 2-connected (Thm 3)	–	2-connected (Thm 3)	–
KA	All graphs (Thm 4)	$\mathcal{G}_{\text{4-path}}$ [4] $\mathcal{G}_{\text{oriented-5-path}}$ (Thm 1) $\mathcal{G}_{\text{triangle}}$ (Thm 1)	All graphs (≥ 3 nodes) (Thm 4)	Not 2-connected (≥ 3 nodes) (Prop 1)
OT	All graphs [18]	–	All graphs [18]	$\mathcal{G}_{\text{2-vs-3}}$ (Thm 2)

Table 1: Summary of Upper and Lower Bound Results. Read as "[row label] is necessary/sufficient for [column label]." E.g. the IT setting suffices to construct 1-THAB for any 2-connected family of graphs, whereas KA is needed to construct 1-THB for $\mathcal{G}_{\text{triangle}}$.

In addition, and as a demonstration of the power and applicability of the techniques developed, in the full version of this work we provide a characterization of the feasibility of 1-THB and 1-THAB for all graph classes on up to 4 parties. The characterization uses a partition of the 4-node graphs into multiple classes, each of which can be handled by a separate technique.

Organization of the Paper We proceed in Section 2 to provide an overview of the core new techniques toward proving our main results. In Section 3 we provide

the necessary definitions and preliminaries. In Section 4 and Section 5 we present an abbreviated version of our THB and THAB lower bounds, respectively. And, in Section 6 and Section 7 we include a short version of our information-theoretic and KA-based upper bounds. We refer the reader to the full version of this paper for detailed treatment of these results, as well as corollaries and implications to characterization of 1-THB, 1-THAB, and 1-THC.

2 Technical Overview

We next highlight a selection of our new analysis and protocol-construction techniques, described in Sections 2.1 to 2.4. We will describe two analysis techniques that are used in our lower bounds: "*phantom jump*" and "*artificial over-extension*." In addition, we will describe two protocol-design techniques that are used in our upper bounds: "*censored brute force*" and "*dead-end channels*."

2.1 Analysis Technique: "Phantom Jump"

The "phantom jump" technique is a means for proving indistinguishability of the transcript of messages sent across a given edge A-B in THB executions on two different graphs, via a sequence of intermediate indistinguishability steps, each appealing to THB security for a different graph pair. In applications, the initial and final graphs will have a party "jump" from one side of the graph to the other, which will be used within the key-agreement implication analysis.

This technique is used within some of our key-agreement lower bounds. We focus here on a specific example for the class $\mathcal{G}_{\mathsf{triangle}}$ (of a triangle graph with a potential edge missing). We point the reader to more elaborate examples on 4-node graph classes in the full version.

We start by recalling how a 1-THB protocol π for $\mathcal{G}_{\mathsf{4\text{-}path}} = \{(A\text{-}B\text{-}C\text{-}D), (B\text{-}C\text{-}D\text{-}A)\}$ was used to construct key agreement in [4]. The idea is for Alice to choose two long random strings r_1 and r_2 and send them to Bob in the clear. Next, Alice and Bob continue in phases as follows:

- In each phase Alice and Bob locally toss coins x_{Alice} and x_{Bob}, respectively.
- They proceed to run two executions of π in which Alice always emulates B and C and Bob emulates D. In addition, if $x_{\mathsf{Alice}} = 0$ then Alice emulates A (as a neighbor of B) broadcasting r_1 in the first run; otherwise she emulates A broadcasting r_2 in the second run. Similarly, if $x_{\mathsf{Bob}} = 1$ then Bob emulates A (as a neighbor of D) broadcasting r_1 in the first run; otherwise he emulates A broadcasting r_2 in the second run.
- If parties B and D output r_1 in the first run and r_2 in the second, Alice and Bob output their bits x_{Alice} and x_{Bob}, respectively; otherwise, they execute another phase.

Clearly, if $x_{\mathsf{Alice}} = x_{\mathsf{Bob}}$ in some iteration then Alice and Bob will output the same coin, and by the assumed security of π, the eavesdropper Eve will not be able to learn who emulated A in the first run and who in the second. If $x_{\mathsf{Alice}} \neq x_{\mathsf{Bob}}$,

then in at least one of the runs nobody emulates the broadcaster A, so with overwhelming probability Alice and Bob will detect this case.

We now show how to adjust this argument to $\mathcal{G}_{\mathsf{triangle}}$. Constructing the KA protocol is rather similar, where Alice always emulates B and Bob always emulates C, and each party emulates the broadcaster A based on their local coins x_{Alice} and x_{Bob} (see Figure 3). Proving correctness follows exactly as in the argument from [4]; however, proving security is more involved. Indeed, in $\mathcal{G}_{\mathsf{4\text{-}path}}$ the view of Eve corresponds to a partial view of the intermediate node C who is never a neighbor of A, and so by the security of π, never learns its direction to A. When considering $\mathcal{G}_{\mathsf{triangle}}$, the view of Eve consists of the communication between B and C, and one of them must be a neighbor of A.

This is where the new phantom-jump technique comes into play. As opposed to [4], we do not construct a reduction from Eve to the security of the THB protocol; rather, we use a direct indistinguishability argument. Notice that the KA construction required the use of only two graphs (A-B-C) and (B-C-A). The third graph (the triangle) is needed for the proof.

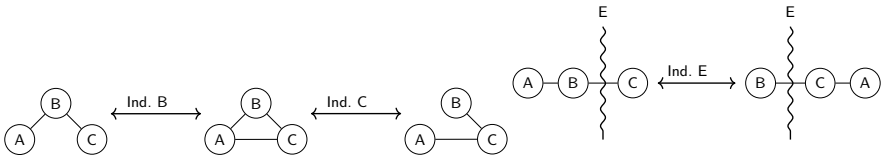


Fig. 3: 1-THB on $\mathcal{G}_{\mathsf{triangle}}$ implies KA.

As depicted in Figure 3, the view of Eve consists of the communication between B and C. By THB security B cannot distinguish between the 3-path (A-B-C) and the complete triangle; in particular, the distribution of the messages on the channel between B and C is indistinguishable in both cases. Similarly, by THB security C cannot distinguish between the 3-path (B-C-A) and the complete triangle; in particular, the distribution of the messages on the channel between B and C is indistinguishable in both cases. By a simple hybrid argument it follows that the messages between B and C are indistinguishable when communicating in (A-B-C) and when communicating in (B-C-A). It follows that the distinguishing advantage of Eve is negligible.

2.2 Analysis Technique: "Artificial Over-Extension"

The artificial over-extension technique is used for proving two of our lower bounds. First, Theorem 1 where 1-THB for $\mathcal{G}_{\mathsf{oriented\text{-}5\text{-}path}}$ is used to construct infinitely often KA (see also Section 4.1); and second, Theorem 2 where 1-THAB for $\mathcal{G}_{\mathsf{2\text{-}vs\text{-}3}}$ is used to construct infinitely often OT (see Section 5.2). In the following, we focus on the latter.

Recall that in the class $\mathcal{G}_{\text{2-vs-3}}$ a party (say A) that has a single neighbor (say B) does not know whether B has another neighbor C or not. This uncertainty is the source of the cryptographic hardness we present; indeed, if the parties know that an honest majority cannot be assumed (i.e., there are only two parties) then 1-THAB is trivial, whereas if an honest majority can be assumed (i.e., there are three parties) then 1-THAB exists assuming KA (by Theorem 4). We also note that without anonymity, 1-THB trivially exists in $\mathcal{G}_{\text{2-vs-3}}$ (via the flooding protocol).

We start with an intermediate goal, that of constructing oblivious transfer from a *two-round* 1-THAB protocol π for the graph class $\mathcal{G}_{\text{2-vs-3}}$,[21] and later explain how the novel "artificial over-extension" technique allows us to extend this construction to arbitrary constant-round protocols. Note that using this technique we can only construct *infinitely often* OT, and extending the implication to a full-blown OT is left as an interesting open question.

OT from two-round 1-THAB. Given a two-round 1-THAB protocol π we construct a secure two-party protocol for Boolean AND (which in turn implies OT [14]).

In the protocol, Alice and Bob will emulate an execution of the 1-THAB protocol on a path, where each extends the length of the path (by emulating an extra party) if their input is 1. More concretely, Alice simulates a single node B if her input is 0, and two nodes A-B if her input is 1. Similarly, Bob simulates a single node C if his input is 0 and two nodes C-A if his input is 1 (see Figure 4). Next, Alice chooses a message $m \xleftarrow{R} \{0,1\}^\lambda$ at random, sends it to Bob in the clear, and initiates an execution of π on message m on the graph with her left-most node (either B or A) as broadcaster. At the conclusion of π, Bob identifies whether his right-most emulated party (either C or A) correctly outputs m. If so, then Bob outputs 0; if not, he outputs 1.

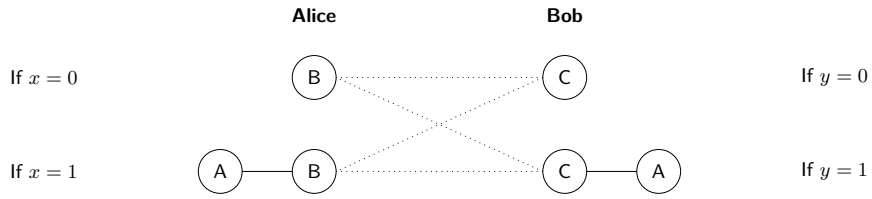


Fig. 4: Boolean AND from two-round 1-THAB for {B-C, A-B-C, B-C-A}

We show that this protocol securely computes AND of Alice and Bob's inputs.

[21] In fact, for this step we will only need for the smaller graph class {B-C, A-B-C, B-C-A} $\subset \mathcal{G}_{\text{2-vs-3}}$.

– For *security*, we exploit the fact that the only case where there is something to hide (namely, if a party holds input 0) is where the respective party has control over just a *single* node in π. Security therefore follows from the fact that π is a THAB protocol with security against one corruption. For example, the views of a corrupt Alice emulating B within executions over graphs $\mathsf{B\text{-}C}$ (Bob has input 0) and $\mathsf{B\text{-}C\text{-}A}$ (Bob has input 1) are indistinguishable.
  Note here that for security it is crucial that π is an *anonymous* broadcast protocol, because in case $x = 0$, Alice broadcasts from node B and in case $x = 1$ from node A. (In fact, as noted above, 1-THB can be achieved trivially on $\mathcal{G}_{\mathsf{2\text{-}vs\text{-}3}}$.)
– For *correctness*, first note that when at least one party has input 0, the corresponding graph is an element of $\{\mathsf{B\text{-}C}, \mathsf{A\text{-}B\text{-}C}, \mathsf{B\text{-}C\text{-}A}\} \subset \mathcal{G}_{\mathsf{2\text{-}vs\text{-}3}}$, in which case proper delivery of m to Bob's right-most node is guaranteed by correctness of π. On the other hand, when $x \wedge y = 1$ (i.e., both Alice and Bob emulate node A) the parties effectively emulate π over an "invalid" length-4 path $\mathsf{A\text{-}B\text{-}C\text{-}A}$. While behavior of π within such execution is unclear, since π runs in only 2 rounds, the message m simply cannot reach the right-most node emulated by Bob at distance 3. Thus, Bob will correctly output 1.

Infinitely often OT from constant-round 1-THAB. Note that correctness of the construction above crucially relies on efficiently detecting an execution of π on the graph $\mathsf{A\text{-}B\text{-}C\text{-}A}$, leveraging its insufficient round complexity. However, this argument is no longer guaranteed when π completes in more than two rounds. This is where the "artificial over-extension" technique comes into play.

The insight is that *either* an execution of π on graph $\mathsf{A\text{-}B\text{-}C\text{-}A}$ can indeed be efficiently detected, in which case the protocol above extends (and we are done), *or* π actually provides a stronger form of topology hiding that we can further leverage. Namely, if neither Alice nor Bob can identify when π is executed on $\mathsf{A\text{-}B\text{-}C\text{-}A}$ as opposed to a legal graph, then in particular π provides 1-THAB for the larger graph class $\mathcal{G}_{\mathsf{2\text{-}vs\text{-}3}}{}' := \mathcal{G}_{\mathsf{2\text{-}vs\text{-}3}} \cup \{\mathsf{A\text{-}B\text{-}C\text{-}A}\}$.

In this case, we can take a similar approach to above, but with the graphs $\{\mathsf{A\text{-}B}, \mathsf{C\text{-}A\text{-}B}, \mathsf{A\text{-}B\text{-}C\text{-}A}\} \subset \mathcal{G}_{\mathsf{2\text{-}vs\text{-}3}}{}'$, with Alice emulating A or $\mathsf{C\text{-}A}$, and Bob emulating B or $\mathsf{B\text{-}C\text{-}A}$, and hope that π identifiably breaks down on the "over-extended" path $\mathsf{C\text{-}A\text{-}B\text{-}C\text{-}A}$ of length 5. If not, this argument repeats, until—via this artificial over-extension technique—ultimately we reach a graph class $\mathcal{G}$ for which:

– π *is* 1-THAB on $\mathcal{G}$, including $\{\mathsf{Y\text{-}Z}, \mathsf{X\text{-}Y\text{-}Z}, \mathsf{Y\text{-}Z\text{-}}\boldsymbol{P}\} \subset \mathcal{G}$
– π is *not* 1-THAB on $\mathcal{G} \cup \{\mathsf{X\text{-}Y\text{-}Z\text{-}}\boldsymbol{P}\}$,

where $\mathsf{X}, \mathsf{Y}, \mathsf{Z} \in \{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$, and $\boldsymbol{P}$ is a path of length upper-bounded by the round complexity of π. Once we do, then the original secure-AND protocol approach will succeed, modulo some differences described below, with Alice emulating Y or $\mathsf{X\text{-}Y}$, and Bob emulating Z or $\mathsf{Z\text{-}}\boldsymbol{P}$.

To argue that eventually we find a path $\mathsf{X\text{-}Y\text{-}Z\text{-}}\boldsymbol{P}$ for which π identifiably breaks down, we again appeal to its bounded round complexity, i.e., π must fail identifiably (with probability 1) once the length of the path exceeds the

round complexity. The limitation of constant rounds is a subtle side effect of the corresponding hybrid argument, to argue that there must be some step where we jump sufficiently from indistinguishable to efficiently identifiable.

Consider the resulting secure-AND protocol, once an appropriate $\mathsf{X}, \mathsf{Y}, \mathsf{Z}, \boldsymbol{P}$ are found. The only modification from the simpler two-round version is how to detect the (over-extended) case $x \wedge y = 1$. When π was two rounds, identifying this event was immediate: Bob's right-most party simply will not receive the delivered message. Here, this is not necessarily the case, as the identifiable "breakdown" of π may occur before the length of $\mathsf{X}\text{-}\mathsf{Y}\text{-}\mathsf{Z}\text{-}\boldsymbol{P}$ exceeds π's round complexity. Thus, instead, the parties will run the distinguisher that—roughly speaking—exists from the fact that π is not 1-THAB on $\mathcal{G} \cup \{\mathsf{X}\text{-}\mathsf{Y}\text{-}\mathsf{Z}\text{-}\boldsymbol{P}\}$. This is the reason why our final protocol guarantees correctness only for infinitely many $\lambda \in \mathbb{N}$: All we can say is that *either* the protocol π is 1-THAB on $\mathcal{G} \cup \{\mathsf{X}\text{-}\mathsf{Y}\text{-}\mathsf{Z}\text{-}\boldsymbol{P}\}$ and we can continue with the extension argument, *or* π is not 1-THAB, i.e., there exists a distinguisher that efficiently detects the "too-long" path $\mathsf{X}\text{-}\mathsf{Y}\text{-}\mathsf{Z}\text{-}\boldsymbol{P}$ with noticeable advantage for *infinitely many* $\lambda \in \mathbb{N}$. Finally, in order to boost correctness towards negligible correctness error (for infinitely many λ), Alice and Bob simply run the protocol π and the distinguisher sufficiently many times, each time on input of a fresh message m, and take a corresponding majority vote.

2.3 Protocol Design: "Censored Brute Force"

This technique enables constructing unconditionally secure pairwise channels between each pair of parties which further guarantees sender anonymity. Such anonymous and private channels are used for proving Theorem 3 and Corollary 1, by constructing 1-THAB and 1-THC with information-theoretic security for any class $\mathcal{G}$ for which all graphs are 2-connected (see Section 6.1 for more details). Recall that the communication complexity of the resulting protocols is polynomial in the size of $\mathcal{G}$ (which could be superpolynomial) and the computation complexity is polynomial in the size of $\mathcal{G}$ and exponential in the maximal degree of $\mathcal{G}$.

The high-level idea is twofold: For any *single* 2-connected graph G, we show how to unconditionally perform sender-anonymous point-to-point communication on G with an ability for any party to (anonymously) "censor" the communication, i.e., yielding delivery of random garbage instead of the intended message. Then, for a given class of 2-connected graphs $\mathcal{G}$, the parties will simultaneously execute (in parallel) a separate anonymous-communication protocol for *every graph* $G \in \mathcal{G}$; for each such G-execution, a party will *censor* the execution if its true neighborhood is inconsistent with its neighborhood in G. As such, the only protocol execution that remains uncensored will be the one corresponding to the *correct* execution graph G (and the identity of which G this corresponds to can be made hidden to the receiving party). We elaborate on these two aims below.

Communicating anonymously in a 2-connected graph. More concretely, suppose we have a *single* 2-connected graph G on vertex set $[n]$, and fix some designated

source and target nodes $\sigma \neq \tau \in [n]$. Let $H_{\sigma\tau}$ denote an arbitrary *$\sigma\tau$-orientation* of G,[22] i.e., a directed acyclic graph with unique sink τ and unique source σ formed by assigning a direction to each edge in G. Moreover, label all nodes $1, 2, \ldots, n$ according to a topologically consistent ordering of $H_{\sigma\tau}$ (beginning with σ and ending at τ). We consider the numbering/orientation of any graph G to be a public parameter, computed according to some deterministic procedure (see full version).

Now suppose node u wishes to send a message m to the target node τ anonymously and securely on the graph G. In the first round, the source σ (i.e., the node labeled 1) prepares additive shares of 0 (or of m if $\sigma = u$) for each of its outgoing edges in $H_{\sigma\tau}$. In round 2, the source σ sends the corresponding share to its neighbor node labeled 2, who then prepares secret shares of what it received ($+m$ if it is u) for each outgoing edge. More generally, in round $i < n$ all nodes with an edge to the i^{th} node send their shares to the i^{th} node. The i^{th} node, having received shares on all incoming edges, then sums up what it receives (adds m if it is u) and prepares additive shares of the result for each of its outgoing edges. In round n, all nodes with edges to τ (the target node) send their shares to τ and τ outputs their summation.

Correctness follows from the homomorphic properties of additive secret sharing. To see why this protocol is secure (namely, that it hides u and m), note that the 2-connectivity of G implies that there are at least 2 vertex-disjoint $\sigma\tau$-paths in $H_{\sigma\tau}$. Thus, the messages any intermediate party (corresponding to $2, \ldots, n-1$) receives are uniformly random because that node is in some sense always missing at least one share (corresponding to a disjoint $\sigma\tau$-path); the source σ does not receive anything at all, and the view of the target τ is simply a random sharing of its output m.

This protocol enjoys some other useful properties. Most notably, any non-sink node can covertly "censor" communication by simply preparing (and sending) shares of a uniformly random message, instead of preparing shares corresponding to what they received (as per the protocol). The view of every other party is identically distributed, with the exception of τ who now receives secret shares of a uniformly random message in the final round.

Compiling to hide topology. Now, let $\mathcal{G}$ be a class of 2-connected graphs on vertex set $[n]$. Loosely speaking, the parties will simulate the above protocol for every possible graph in $\mathcal{G}$ simultaneously. Each node will covertly censor every protocol corresponding to a graph that is *not* locally consistent with their local neighborhood (sending random messages at the appropriate times). As a result, exactly one protocol (corresponding to the "real" graph) will give the correct output message and all others will give uniformly random output.

To be slightly more concrete, all nodes will execute the protocol above for each graph in the class in parallel. To keep track of which message is which, for every node but τ we will label the messages with the graph/protocol that the

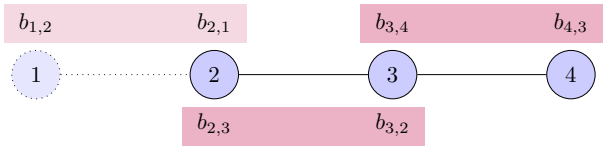
[22] The standard notation in the literature is *st*-orientation; to avoid confusion with the notation t that stands for the corruption threshold, we use $\sigma\tau$-orientation instead.

message corresponds to. If an edge is missing from the real graph, but present in a graph corresponding to one of the simulated protocols, the corresponding message cannot be sent. However, the receiving node knows not to expect a message either. From this and the uniformly random nature of non-terminal messages in the above protocol, nothing is leaked locally by labeling the simulations. However, sending labeled messages to τ would clearly identify the "real" topology. So instead, all parties will send all final protocol messages in randomly permuted order. To enable τ identifying the real output, the sender will append a long checksum to the message. The target τ will try all message combinations (this is the reason for the exponential dependency in the maximal degree) and output the unique one with a correct checksum (or abort if more than one message has a valid checksum).

2.4 Protocol Design Technique: "Dead-End Channels"

The Dead-End Channels technique is used to obtain 1-THAB for all graphs of at least 3 nodes (and 1-THB for all graphs), assuming existence of *key agreement*. Recall that before the present work, such results were only known assuming oblivious transfer [18].

The high-level idea of our 1-THAB protocol, as in Moran et al. [18], is to broadcast the message via *flooding*, but in a way that hides from the parties at which round they received the broadcast message. This can be achieved by passing the message between *virtual parties*, each consisting of two real parties that hold additive secret shares of the message (depicted, e.g., as purple bars for each neighboring pair of parties below). Only in the final round will the parties exchange their secret shares and recover the message.



The challenge thus becomes passing the messages between virtual parties. In [18] this is solved by using oblivious transfer (OT) to run an MPC protocol realizing the virtual party, and allowing every adjacent pair of virtual parties to securely compute the *OR* of their messages.

In our setting, we do not have the ability to perform secure computations pairwise between parties without OT. Instead, we leverage the fact that given at least three nodes we are guaranteed an honest majority, and can therefore (once the parties establish secure channels using the key agreement protocol) build on techniques from information-theoretic secure computation to appropriately pass along the message.

However, this itself is not so straightforward. For example, in the image above, the neighboring parties 2-3-4 would wish to jointly emulate a three-party secure computation to perform the secure transfer from 2-3 to 3-4. But, the issue

is that parties cannot reveal whether they truly have neighbors with which to jointly compute: for example, party 2 above must then emulate a nonexistent neighbor 1 to hide its true degree. Thus grouping parties in three, including possibly a simulated neighbor, would allow the adversary to gain control over a majority. (On the other hand, building on secure computation including four our more neighbored parties, the same party could appear several times in the protocol and therefore potentially learn about the connectivity of its neighbors.)

Our approach builds on the following idea: We will give one party within each group of three the role of a *dealer* to deal *OT correlations*, which can be used to establish a secure OT channel between two other parties. This alone is not sufficient, as one of the parties could be simulated by the dealer (in the case that the dealer has degree one), and therefore allows the dealer to gain full control over the OT channel, and in particular learn the honest parties' inputs. To prevent this, we observe that — again using OT correlations — one can establish *dead-end channels* (i.e., information sent via such a channel cannot be read by anyone apart from the sender) if and only if the receiver is a simulated party. Therefore, even if the dealer simulates one of the parties, it does not learn anything about the honest parties' inputs. Note that it is crucial that dead-end channels are indistinguishable from secure channels from the view of the sender. Further, a key observation is that using OT correlations to establish dead-end channels does not leak anything about the topology, even if the dealer of the OT correlations has degree one. This is the case, because the only thing the dealer could potentially learn from the other party is whether its degree is one — but if the dealer has degree one it already knows that the degree of its neighbor must be at least two (as we are guaranteed a connected graph with a strict honest majority).

3 Preliminaries

Notations. For $n \in \mathbb{N}$ let $[n] = \{1, \cdots, n\}$. In our protocols we sometimes denote by B an upper bound on the number of participating parties, by n the number of actually participating parties, and by t an upper bound on the number of corrupted parties. The security parameter is denoted by λ.

Graph notations and properties. A graph $G = (V, E)$ is a set V of vertices and a set E of edges, each of which is an unordered pair $\{v, w\}$ of distinct vertices. A graph is *directed* if its edges are instead ordered pairs (v, w) of distinct vertices. An *oriented* graph is a directed graph having no symmetric pair of directed edges, and an *orientation* of an undirected graph is an assignation of a direction to each of its edges so as to make it oriented. A graph is *k-connected* if it has more than k vertices and remains connected whenever fewer than k vertices are removed. A graph class $\mathcal{G}$ is k-connected if every graph $G \in \mathcal{G}$ is k-connected. Throughout this paper we only consider *connected* graphs, even if we do not systematically make this explicit. The *(open) neighborhood* of a vertex v in an undirected graph G, denoted $\mathcal{N}_G(v)$, is the set of vertices sharing an edge with v in G. The *closed neighborhood* of v in G is in turn defined by $\mathcal{N}_G[v] \coloneqq \mathcal{N}_G(v) \cup \{v\}$.

UC framework. We work in the UC framework of Canetti [6]. Unless stated otherwise, we will consider computationally unbounded, static, and semi-honest adversaries and environments.

Topology-hiding computation (THC). We recall the definition of topology-hiding computation from [18, 4]. The real-world protocol is defined in a model where all communication is transmitted via the functionality $\mathcal{F}_{\mathsf{graph}}^{\mathcal{G}}$ (described in Figure 5). The functionality is parametrized by a family of graphs $\mathcal{G}$, representing all possible network topologies (aka communication graphs) that the protocol supports. We implicitly assume that every node in a graph is associated with a specific *party identifier*, pid. To simplify the notation, we will consider that P_v in the protocol is associated with node v in the graph.

Initially, before the protocol begins, $\mathcal{F}_{\mathsf{graph}}^{\mathcal{G}}$ receives the network communication graph G from a special graph party P_{graph}, makes sure that $G \in \mathcal{G}$, and provides to each party P_v with $v \in V$ its local neighbor-set. Next, during the protocol's execution, whenever party P_v wishes to send a message m to party P_w, it sends (v, w, m) to the functionality; the functionality verifies that the edge (v, w) is indeed in the graph, and if so delivers (v, w, m) to P_w.

Note that if all the graphs in $\mathcal{G}$ have exactly n nodes, then the exact number of participants is known to all and need not be kept hidden. In this case, defining the ideal functionality and constructing protocols becomes a simpler task. However, if there exist graphs in $\mathcal{G}$ that contain a *different* number of nodes, then the model must support functionalities and protocols that only know an *upper bound* B on the number of participants. In the latter case, the actual number of participating parties n must be kept hidden.

Given a class of graphs $\mathcal{G}$ with an upper bound B on the number of parties, we define a protocol π with respect to $\mathcal{G}$ as a set of B PPT interactive Turing machines (ITMs) $(P_1, \ldots, P_B)$ (the parties), where any subset of them may be activated with (potentially empty) inputs. Only the parties that have been activated participate in the protocol, send messages to one another (via $\mathcal{F}_{\mathsf{graph}}^{\mathcal{G}}$), and produce output.

An ideal-model computation of a functionality $\mathcal{F}$ is augmented to provide the corrupted parties with the information that is leaked about the graph; namely, every corrupted (dummy) party should learn its neighbor-set. Note that the functionality $\mathcal{F}$ can be completely agnostic about the actual graph that is used, and even about the family $\mathcal{G}$. To augment $\mathcal{F}$ in a generic way, we define the wrapper-functionality $\mathcal{W}_{\mathsf{graph\text{-}info}}^{\mathcal{G}}(\mathcal{F})$, that runs internally a copy of the functionality $\mathcal{F}$. The wrapper $\mathcal{W}_{\mathsf{graph\text{-}info}}^{\mathcal{G}}(\cdot)$ acts as a shell that is responsible to provide the relevant leakage to the corrupted parties; the original functionality $\mathcal{F}$ is the core that is responsible for the actual ideal computation.

More specifically, the wrapper receives the graph $G = (V, E)$ from the graph party P_{graph}, makes sure that $G \in \mathcal{G}$, and sends a special initialization message containing G to $\mathcal{F}$. (If the functionality $\mathcal{F}$ does not depend on the communication graph, it can ignore this message.) The wrapper then proceeds to process messages as follows: Upon receiving an initialization message from a party P_v responds with its neighbor set $\mathcal{N}_G(v)$ (just like $\mathcal{F}_{\mathsf{graph}}^{\mathcal{G}}$). All other input messages

The functionality $\mathcal{F}_{\mathsf{graph}}^{\mathcal{G}}$

The functionality $\mathcal{F}_{\mathsf{graph}}^{\mathcal{G}}$ is parametrized by a family of graphs $\mathcal{G}$; let B denote the maximal number of nodes in $G \in \mathcal{G}$. The functionality proceeds with a special graph party P_{graph} and with a subset of the parties $P_1, \ldots, P_B$ (to be defined by the graph received from P_{graph}) as follows.

Initialization Phase:

Input: $\mathcal{F}_{\mathsf{graph}}^{\mathcal{G}}$ waits to receive the graph $G = (V, E)$ from P_{graph}. If $G \notin \mathcal{G}$, abort.

Output: Upon receiving an initialization message from P_v, verify that $v \in V$, and if so send $\mathcal{N}_G(v)$ to P_v.

Communication Phase:

Input: $\mathcal{F}_{\mathsf{graph}}^{\mathcal{G}}$ receives from a party P_v a destination/data pair (w, m) where $w \in \mathcal{N}_G(v)$ and m is the message P_v wants to send to P_w. (If $v, w \notin V$, or if w is not a neighbor of v, $\mathcal{F}_{\mathsf{graph}}^{\mathcal{G}}$ ignores this input.)

Output: $\mathcal{F}_{\mathsf{graph}}^{\mathcal{G}}$ gives output (v, m) to P_w indicating that P_v sent the message m to P_w.

Fig. 5: The communication graph functionality

from a party P_v are forwarded to $\mathcal{F}$ and every message from $\mathcal{F}$ to a party P_v is delivered to its recipient.

Note that formally, the set of all possible parties V^* is fixed in advance. To represent a graph $G' = (V', E')$ where $V' \subset V^*$ is a subset of the parties, we use the graph $G = (V^*, E')$, where all vertices $v \in V^* \setminus V'$ have degree 0.

Definition 1 (Topology-hiding computation). *We say that a protocol π securely realizes a functionality $\mathcal{F}$ in a topology-hiding manner with respect to $\mathcal{G}$ tolerating a semi-honest adversary corrupting t parties if π securely realizes $\mathcal{W}_{\mathsf{graph\text{-}info}}^{\mathcal{G}}(\mathcal{F})$ in the $\mathcal{F}_{\mathsf{graph}}^{\mathcal{G}}$-hybrid model tolerating a semi-honest adversary corrupting t parties.*

Broadcast and anonymous broadcast. In this work we will focus on topology-hiding computation of two central functionalities. The first is the *broadcast* functionality (see Figure 6), where a designated and publicly known party, named *the broadcaster*, starts with an input value m. Our broadcast functionality guarantees that every party that is connected to the broadcaster in the communication graph receives the message m as output. In this paper, we assume the communication graphs are always connected. However, the broadcaster may not be participating, in which case it is represented as a degree-0 node in the communication graph (and all the participating nodes are in a separate connected component.)

Parties that are not connected to the broadcaster receive a message that is supplied by the adversary (we can consider stronger versions of broadcast, but this simplifies the proofs).

We denote the broadcast functionality where the broadcaster is P_i by $\mathcal{F}_{\mathsf{bc}}(P_i)$.

The functionality $\mathcal{F}_{\mathsf{bc}}(P_i)$

The broadcast functionality $\mathcal{F}_{\mathsf{bc}}(P_i)$ is parametrized by the broadcaster P_i and proceeds as follows.

Initialization: The functionality receives the communication graph G from the wrapper $\mathcal{W}_{\mathsf{graph\text{-}info}}$.

Input: Record the input message $m \in \{0,1\}$ sent by the broadcaster P_i.

Output: Send the output m to every party that is in the same connected component as P_i in G. For every other party in G, the output delivered to that party is supplied by the adversary.

Fig. 6: The broadcast functionality

Definition 2 (t-THB). *Let $\mathcal{G}$ be a family of graphs and let t be an integer. A protocol π is a t-THB protocol with respect to $\mathcal{G}$ if $\pi(P_v)$ securely realizes $\mathcal{F}_{\mathsf{bc}}(P_v)$ in a topology-hiding manner with respect to $\mathcal{G}$, for every P_v, tolerating a semi-honest adversary corrupting t parties.*

The second is the *anonymous-broadcast* functionality (see Figure 7). This functionality is similar to broadcast with the exception that the broadcaster is not known and its identity is kept hidden even after the computation completes. Namely, the environment will activate exactly one of the parties with an input value, informing this party that it is the broadcaster. We denote the anonymous broadcast functionality $\mathcal{F}_{\mathsf{anon\text{-}bc}}$.

The functionality $\mathcal{F}_{\mathsf{anon\text{-}bc}}$

The anonymous-broadcast functionality $\mathcal{F}_{\mathsf{anon\text{-}bc}}$ proceeds as follows.

Initialization: The functionality receives the communication graph G from the wrapper $\mathcal{W}_{\mathsf{graph\text{-}info}}$.

Input: Upon receiving an input message $m \in \{0,1\}$ from one of the parties P_i, record it.

Output: If exactly one input message m from party P_i was received, Send the output m to every party that is in the same connected component as P_i in G. For every other party in G, the output delivered to that party is supplied by the adversary.
If more than one input was received, send G and all received inputs to the adversary, and for every party in G, the output delivered to that party is supplied by the adversary (i.e., there is no security guarantee if more than one input was received.)

Fig. 7: The anonymous-broadcast functionality

Definition 3 (t-THAB). *Let $\mathcal{G}$ be a family of graphs and let t be an integer. A protocol π is a t-THAB protocol with respect to $\mathcal{G}$ if π securely realizes $\mathcal{F}_{\mathsf{anon\text{-}bc}}$ in a topology-hiding manner with respect to $\mathcal{G}$, tolerating a semi-honest adversary corrupting t parties.*

4 THB Lower Bounds

In this section we demonstrate that achieving broadcast while hiding certain graph properties necessitates cryptographic assumptions.

4.1 Hiding Distance Requires io-KA (The Oriented 5-Path)

In this section, we show that hiding the distance from the broadcaster, in constant rounds, requires infinitely-often Key Agreement (io-KA). In particular, we will show that any constant-round protocol for the class $\mathcal{G}_{\text{oriented-5-path}}$ (Figure 8), implies io-KA.

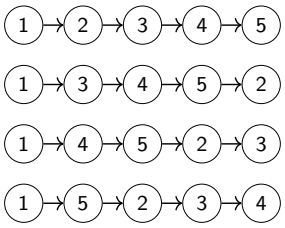


Fig. 8: The class $\mathcal{G}_{\text{oriented-5-path}}$ of oriented paths, rooted in P_1. Communication is bidirectional, arrows simply indicate that nodes can deduce the broadcaster's direction.

In this class the nodes ②, ③, ④, ⑤ always know the direction of the broadcaster, ① (it's in the direction of their lowest-valued neighbor, mod 5), but cannot distinguish (from their local neighborhood) whether they are distance 2 or 3 from the broadcaster. E.g. ③ cannot distinguish between ①-②-③-④-⑤ and ①-⑤-②-③-④, as in both cases its local neighborhood is ②-③-④. Note that if just distance is leaked to this class, the trivial flooding protocol is secure.

Intriguingly, the resulting key agreement construction is not fully-black-box, nor is it even explicit. Further, our result critically requires the $\mathcal{G}_{\text{oriented-5-path}}$-THB to be efficient in *round* complexity. We remark that such a limitation is inherent, as we demonstrate that $\mathcal{G}_{\text{oriented-5-path}}$ *unconditionally* admits an ϵ-secure topology-hiding broadcast protocol that works in $O(1/\epsilon)$ rounds, for any $\epsilon > 0$.[23] In contrast, the key agreement construction of Section 4.2 *is* fully black-box and rules out the existence of such an upper bound for the class $\mathcal{G}_{\text{triangle}}$. It remains open whether an ϵ-secure $\mathcal{G}_{\text{oriented-5-path}}$-THB in $< 1/\epsilon$ rounds requires io-KA, or more generally whether negligible security in polynomial rounds requires io-KA.

Theorem 6. *If there exists a constant-round 1-THB protocol for the class $\mathcal{G}_{\text{oriented-5-path}}$, then infinitely-often key-agreement also exists.*

[23] In fact, the upper bound holds for a large body of graph classes, where only distance need be hidden.

The proof introduces an argument called '*artificial over-extension*' (Section 2) which involves using the 1-THB on longer and longer graphs (outside of the scope of its correctness and security guarantees) until the protocol breaks in an identifiable way. Details are deferred to the full version of this paper.

4.2 Hiding Neighbor Distances Requires KA (The Triangle)

Consider the class $\mathcal{G}_{\mathsf{triangle}} = \{G^0_{\mathsf{tr}}, G^1_{\mathsf{tr}}, G^2_{\mathsf{tr}}\}$ as represented in Figure 9, which we (abusively) call '*the Triangle*'. The players are P_1, P_2, P_3 with the broadcaster always being P_1; P_2 and P_3 are connected, and P_2 and/or P_3 is connected to P_1.

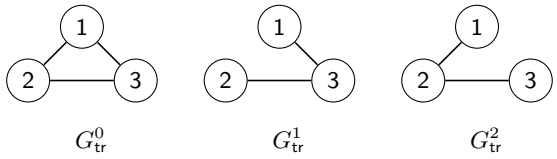


Fig. 9: The class $\mathcal{G}_{\mathsf{triangle}}$.

The *secret of the topology* can be summarized as follows: if one of the two non-broadcasting parties P_2 or P_3 is connected to the broadcaster, it does not know if the other is as well. Note that preserving the secret of the topology of $\mathcal{G}_{\mathsf{triangle}}$ can also be reformulated as 'hiding the neighbor distances' from the parties. Indeed, for P_2 (resp. P_3) knowing the topology means knowing if P_3 (resp. P_2) is at distance one or two from the broadcaster.

Theorem 7 (Broadcast on 'The Triangle' requires KA). *If there exists a 1-THB(1) protocol for the class $\mathcal{G}_{\mathsf{triangle}}$ then there exists a key-agreement protocol.*

In order to prove this theorem, we explicitly construct a key-agreement scheme from a 1-THB(1) protocol π on $\mathcal{G}_{\mathsf{triangle}}$. The construction of this KA protocol follows very closely the proof of Theorem 3.1 (and the associated Protocol 3.2) in Ball et al. [4], but we use a novel technique (the *phantom jump* argument to reduce the security of the key-agreement scheme to the topology-hiding properties of π. See description in Section 2.1; the details of the proof are deferred to the full version of this paper.

5 THAB Lower Bounds

5.1 Low Vertex Connectivity Requires KA

In this section we show how t-THAB on a class which contains even a single graph with at least $t+2$ vertices and which is not $(t+1)$-connected implies Key-Agreement. It should be noted that this is a relatively weak result on its own,

as testified by the fact that even non topology-hiding anonymous broadcast on such a class already implies KA, but we present it here for completeness' sake and because it matches the upper bound of Theorem 11.

Proposition 2. *t-THAB on a class containing a graph with at least $(t+2)$ vertices which is not $(t+1)$-vertex-connected implies KA.*

The result is similar in spirit to that of Ishai et al. [13]—who showed how *anonymity* can be leveraged to obtain *privacy*—. We show it using techniques from [4] however, which are more directly applicable as their setting is the same as ours. The proof is deferred to the full version of this paper.

5.2 Uncertain Honest Majority Requires io-OT (The 2-vs-3 Paths)

In the previous section we showed that, for a large number of graph classes, key-agreement is *necessary* to achieve 1-THAB. A natural follow-up question is to ask whether key agreement is sufficient to achieve 1-THAB *on all graphs* or not. We answer this question negatively by showing that constant-round 1-THAB on the class of paths of length two and three implies *infinitely often oblivious transfer*.

This is similar to the result of Ball et al. [3], who showed no honest majority can imply oblivious transfer in the 2-corruption setting. Note though that our result requires inherently different techniques, as in the one-corruption setting there exists only one graph with no honest majority, namely the path of length 2. 1-THAB on this graph only is trivial, as the only party that is not the broadcaster *knows* that the other party must be the broadcaster. But, adding the path of length 3 (where in fact there is always a honest majority), we can prove an implication to infinitely-often oblivious transfer (io-OT). From a certain point of view, we show that one cannot hide *how far* information travels, unless *always* guaranteed an honest majority.

Theorem 8. *Let π be a constant-round 1-THAB protocol for $\mathcal{G}_{\text{2-vs-3}}$. Then, there exists a uniform infinitely-often OT protocol secure in the presence of a semi-honest adversary.*

In order to prove this theorem, we show that a constant-round 1-THAB for $\mathcal{G}_{\text{2-vs-3}}$ can be used to build a secure two-party infinitely often AND functionality, which in turn implies infinitely often OT. The proof closely follows that of Theorem 6, which introduced the "artificial over-extension" technique, and is described in Section 2.2. The details are deferred to the full version of this paper.

6 Information-Theoretic Upper Bounds

In this section, we present our information-theoretic constructions: in Section 6.1, 1-IT-THAB for 2-connected graphs, and in Section 6.2, 1-IT-THB for the 1-connected butterfly graph.

6.1 2-Connectivity is Sufficient for 1-IT-THAB

On an intuitive level, $(t+1)$-vertex-connectivity could be a sufficient condition to perform t-IT-THAB since messages exchanged between distant parties in the graph can be secret-shared among $t+1$ vertex-disjoint paths. This way, privacy of communication can be ensured (since, with only t corruptions, an adversary cannot recover all the shares). The core challenge, however, is *how* to have the parties route message shares consistently on general, unstructured graphs, in a topology-hiding fashion (in particular, message routing can only be done locally). We prove this intuition to be true for $t = 1$, and provide a way for parties to route secret-shares in 2-connected graphs.

Theorem 9. *Let $n \in \mathbb{N}$, let $\mathcal{G}$ be a class of 2-connected graphs of vertex set size at most n, let d_{max} is the maximal degree of any graph in $\mathcal{G}$, and let $\delta > 0$. Then, there exists a protocol that securely realizes $\mathcal{F}_{\text{anon-bc}}$ with security δ in a topology-hiding manner with respect to $\mathcal{G}$, tolerating a single semi-honest corruption.*

Moreover, the protocol completes within n rounds with total communication complexity $O(n^2 d_{max} \cdot |\mathcal{G}| \cdot (\ell + \log(n/\delta) + d_{max} \cdot \log|\mathcal{G}|))$ and computation complexity $O(|\mathcal{G}|^{d_{max}})$ per node.

The proof revolves around a technique we call "*censored brute-force*" (see Section 2.3). Full details are deferred to the full version of this paper.

6.2 2-Connectivity is Not Necessary for 1-IT-THB (Butterfly Graph)

Section 5.2 shows a separation between 1-THAB and 1-THB, with the class $\mathcal{G}_{\text{2-vs-3}}$: 1-THAB implies infinitely often OT, yet 1-THB is possible information-theoretically by flooding. In order to understand if the separation is really meaningful or due to an edge case of the definition of THB, we ask whether there is a class which separates the two functionalities and for which 1-THB is not trivial (i.e., cannot be achieved by simple flooding). To this end, we prove there exist graph classes on which 1-IT-THAB is impossible and flooding is not topology-hiding, but there still exist 1-IT-THB. One such class is the family of butterfly graphs (Figure 10), on which 1-IT-THAB is impossible by Proposition 2.

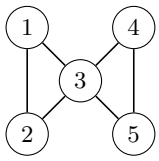


Fig. 10: The class of butterfly graphs consists of all possible permutations of the graph depicted above with nodes in $\{1, 2, 3, 4, 5\}$.

Theorem 10 (IT-THB on butterfly). *There exists a 1-IT-THB protocol with respect to* $\mathcal{G}_{\mathsf{butterfly}}$ *with perfect security.*

What makes this problem non-trivial is that the center node cannot learn which of its neighbors are connected, while the other nodes cannot learn which node is the center node. We present an information-theoretic protocol which runs in two phases. In the first, the broadcaster sends the message to all its neighbors which ensures the center node gets hold of it. In the second phase, the center node broadcasts the message in parallel on a bunch of subgraphs. Each of these is the graph induced by the center and any other two parties. Note that each of these subgraphs is either a triangle or a 3-path, much like in 'the Triangle', described in Section 6.1. Crucially, we ensure the center node does not learn which of them are triangles and which are 3-paths, while also preventing the other parties learning the identity of the center node. Now, every neighbor of the center—i.e., every party—knows the broadcast bit. The protocol is detailed in the full version of the paper.

7 Key-Agreement Upper Bounds

We show that key-agreement is sufficient to achieve 1-THAB on all graphs with *at least* 3 *nodes*—in other words, on all graphs where we are guaranteed an *honest majority*. As shown in Section 5.2 this is the best we can hope to achieve for general graph classes. This result can then be extended to 1-THB for *all graphs*.

Theorem 11 (KA is sufficient for 1-THAB on all graphs of size at least 3). *If there exists a key-agreement protocol, there exists a 1-THAB protocol on the class of all graphs with at least* 3 *and at most* B *vertices.*

Based on [18], the idea is to broadcast the message via *flooding*, but in a way that hides from the parties at which round they received the broadcast message. We leverage the fact there is a guaranteed honest local majority nearly everywhere in the network to run a protocol between locally simulated virtual parties.[24] The weakened assumption when compared to [18] (KA instead OT) means we have to take extra steps to run secure protocols locally; to that effect we introduce the trick of '*dead-end channels*'. The details of this construction are deferred to the full version of this paper.

Theorem 12 (KA is sufficient for 1-THB on all graphs). *If there exists a key-agreement protocol, there exists a 1-THB protocol on the class of all graphs with at most* B *vertices.*

The proof of Theorem 12 follows almost immediately from that of Theorem 11, and only involves introducing a small step to handle the case of size-2 networks. Again, it is left to the full version.

[24] In fact, we critically exploit the fact that if a party has a single neighbor (and thus no guaranteed local honest majority), she *knows* that neighbor's neighborhood is majority honest.

Acknowledgments. We thank the anonymous reviewers of TCC 2020 for pointing to the connection between anonymous communication and key agreement in [13].

M. Ball's research is supported in part by an IBM Research PhD Fellowship. M. Ball and T. Malkin's work is supported in part by JPMorgan Chase & Co. as well as the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research under award number DE-SC-0001234. E. Boyle's research is supported in part by ISF grant 1861/16, AFOSR Award FA9550-17-1-0069, and ERC Starting Grant 852952 (HSS). R. Cohen's research is supported by NSF grant 1646671. L. Kohl's research is supported by ERC Project NTSC (742754). P. Meyer's research is supported in part by ISF grant 1861/16, AFOSR Award FA9550-17-1-0069, and ERC Starting Grant 852952 (HSS).

Bibliography

[1] A. Akavia and T. Moran. Topology-hiding computation beyond logarithmic diameter. In *EUROCRYPT'17, part III*, pages 609–637, 2017.

[2] A. Akavia, R. LaVigne, and T. Moran. Topology-hiding computation on all graphs. In *CRYPTO'17, part I*, pages 447–467, 2017.

[3] M. Ball, E. Boyle, T. Malkin, and T. Moran. Exploring the boundaries of topology-hiding computation. In *EUROCRYPT'18, part III*, pages 294–325, 2018.

[4] M. Ball, E. Boyle, R. Cohen, T. Malkin, and T. Moran. Is information-theoretic topology-hiding computation possible? In *TCC'19, part I*, pages 502–530, 2019.

[5] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *STOC*, pages 1–10, 1988.

[6] R. Canetti. Security and composition of multiparty cryptographic protocols. *J. Cryptology*, 13(1):143–202, 2000.

[7] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981.

[8] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *J. Cryptology*, 1(1):65–75, 1988.

[9] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols (extended abstract). In *STOC*, pages 11–19, 1988.

[10] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public key encryption and oblivious transfer. In *FOCS*, pages 325–335, 2000.

[11] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC*, pages 218–229, 1987.

[12] M. Hirt, U. Maurer, D. Tschudi, and V. Zikas. Network-hiding communication and applications to multi-party protocols. In *CRYPTO'16, part II*, pages 335–365, 2016.

[13] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Cryptography from anonymity. In *FOCS*, pages 239–248, 2006.

[14] J. Kilian. A general completeness theorem for two-party games. In *STOC*, pages 553–560, 1991.

[15] L. Lamport, R. E. Shostak, and M. C. Pease. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[16] R. LaVigne, C. L. Zhang, U. Maurer, T. Moran, M. Mularczyk, and D. Tschudi. Topology-hiding computation beyond semi-honest adversaries. In *TCC'18, part II*, pages 3–35, 2018.

[17] R. LaVigne, C. L. Zhang, U. Maurer, T. Moran, M. Mularczyk, and D. Tschudi. Topology-hiding computation for networks with unknown delays. In *PKC*, pages 215–245, 2020.

[18] T. Moran, I. Orlov, and S. Richelson. Topology-hiding computation. In *TCC'15, part I*, pages 159–181, 2015.

[19] M. C. Pease, R. E. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980.

[20] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *FOCS*, pages 73–85, 1989.

[21] A. C. Yao. Protocols for secure computations (extended abstract). In *FOCS*, pages 160–164, 1982.