Broadcast Amplification

Martin Hirt, Ueli Maurer, Pavel Raykov
{hirt,maurer,raykovp}@inf.ethz.ch

ETH Zurich, Switzerland

Abstract. A d-broadcast primitive is a communication primitive that allows a sender to send a value from a domain of size d to a set of parties. A broadcast protocol emulates the d-broadcast primitive using only point-to-point channels, even if some of the parties cheat, in the sense that all correct recipients agree on the same value v (consistency), and if the sender is correct, then v is the value sent by the sender (validity). A celebrated result by Pease, Shostak and Lamport states that such a broadcast protocol exists if and only if $t < n/3$, where n denotes the total number of parties and t denotes the upper bound on the number of cheaters.

This paper is concerned with broadcast protocols for any number of cheaters ($t < n$), which can be possible only if, in addition to point-to-point channels, another primitive is available. Broadcast amplification is the problem of achieving d-broadcast when d'-broadcast can be used once, for $d' < d$. Let $\phi_n(d)$ denote the minimal such d' for domain size d. We show that for $n = 3$ parties, broadcast for any domain size is possible if only a single 3-broadcast is available, and broadcast of a single bit ($d' = 2$) is not sufficient, i.e., $\phi_3(d) = 3$ for any $d \geq 3$. In contrast, for $n > 3$ no broadcast amplification is possible, i.e., $\phi_n(d) = d$ for any d.

However, if other parties than the sender can also broadcast some short messages, then broadcast amplification is possible for *any* n. Let $\phi^*_n(d)$ denote the minimal d' such that d-broadcast can be constructed from primitives d'_1-broadcast,..., d'_k-broadcast, where $d' = \prod_i d'_i$ (i.e., $\log d' = \sum_i \log d'_i$). Note that $\phi^*_n(d) \leq \phi_n(d)$. We show that broadcasting $8n \log n$ bits in total suffices, independently of d, and that at least $n-2$ parties, including the sender, must broadcast at least one bit. Hence $\min(\log d, n-2) \leq \log \phi^*_n(d) \leq 8n \log n$.

1 Introduction

1.1 Byzantine Broadcast

We consider a set $\mathcal{P} = \{P_1, \ldots, P_n\}$ of n parties connected by authenticated synchronous point-to-point channels.[1] The broadcast problem (also known as

[1] Synchronous means that the parties work in synchronous rounds such that the messages are guaranteed to be delivered within the same round in which they were sent. If the sending party inputs no message (or a message outside the agreed domain) to the channel, then the receiving party gets a default output.

the Byzantine generals problem) is defined as follows [PSL80]: A specific party, the sender, wants to distribute a message to the other parties in such a way that all correct parties obtain the same message, even if up to t of the parties cheat (also called Byzantine) and deviate arbitrarily from the prescribed protocol. We assume that P_1 is the sender and $\mathcal{R} = \mathcal{P} \setminus \{P_1\}$ is the set of recipients. Formally:

Definition 1. *A protocol for the set $\mathcal{P} = \{P_1, \ldots, P_n\}$ of parties, where P_1 has an input value $v \in \mathcal{D}$ and each party in $\mathcal{R}$ outputs a value in $\mathcal{D}$, is called a* broadcast protocol for domain $\mathcal{D}$ *if the following conditions are satisfied:*[2]

CONSISTENCY: *All correct parties $P_i \in \mathcal{R}$ output the same value $v \in \mathcal{D}$.*
VALIDITY: *If the sender P_1 is correct, then v is the input value of P_1.*
TERMINATION: *Every correct party in $\mathcal{P}$ terminates.*

A broadcast protocol can be understood as emulating a so-called broadcast primitive (or channel), i.e., an ideal communication primitive where P_1 inputs a value which is output to all other parties. Broadcast is one of the most fundamental primitives in distributed computing. It is used as building block in various protocols like voting, bidding, collective contract signing, secure multiparty computation, etc.

A celebrated result by Pease, Shostak and Lamport states that for any nontrivial $\mathcal{D}$ (i.e., $|\mathcal{D}| \geq 2$), a broadcast protocol exists if and only if the upper bound t on the number of cheaters satisfies $t < n/3$ [PSL80,BGP92,CW92].

1.2 Broadcast Amplification

This paper is concerned with broadcast protocols for any number of cheaters ($t < n$), which can be possible only if, in addition to point-to-point channels, another primitive is available.[3] We consider perfect security, which means that the cheating probability is zero.

The perhaps most natural choice of such an additional primitive is the availability of some broadcast primitives for smaller domain sizes. Let d-broadcast be a broadcast primitive (or broadcast channel) for message domain size d for a specific sender.

We assume that in addition to point-to-point channels, the parties have access to a system called `BBB` which provides a broadcast primitive as a black-box. If invoked for sender P_i and domain $\mathcal{D}'$, `BBB` takes input $v \in \mathcal{D}'$ from P_i and outputs v to all parties in $\mathcal{P}$ (except P_i).

In this setting, the first and most natural question that arises is: Can a sender broadcast a message with domain size d by using point-to-point communication and broadcasting only a *single* message with domain size $d' < d$?

[2] The domain can without essential loss of generality be assumed to be $\mathcal{D} = [d]$, where here and below we define $[k] = \{1, \ldots, k\}$.

[3] One type of primitive considered previously is a so-called trusted setup [DS83,PW96]. In such a model, perfect security is not achievable, but statistical or cryptographic security is.

Definition 2. *Let $\phi_n(d)$ denote the minimal d' such that d-broadcast can be constructed from d'-broadcast.*

Trivially, $\phi_n(d) \leq d$, as d-broadcast can be constructed directly from d-broadcast.

The most natural generalization of the above question is the following:[4] If any party can broadcast short messages, what is the minimal total number of bits that need to be broadcast to construct an ℓ-bit broadcast? More precisely, since we consider arbitrary alphabet sizes (not just powers of 2), the question is to determine the quantity $\phi_n^*(d)$ defined below.

Definition 3. *Let $\phi_n^*(d)$ denote the minimal d' such that d-broadcast can be constructed from the k primitives d'_1-broadcast, ..., d'_k-broadcast, where $d' = \prod_i d'_i$.*

Note that $\log d' = \sum_i \log d'_i$ is the total number of bits of information[5] broadcast using BBB. It is therefore often natural to state results for the quantity $\log \phi_n^*(d)$. It is obvious that $\phi_n^*(d) \leq \phi_n(d)$.

A protocol that amplifies the domain of a broadcast, in the sense of the above two definitions, is called a *broadcast-amplification protocol*. A broadcast amplification protocol for domain size d can be used to replace a call to a d-broadcast primitive within another protocol. Hence broadcast amplification protocols can be constructed recursively.

One can call $\phi_n(d)$ and $\phi_n^*(d)$ the *intrinsic broadcast complexity* of domain size d, in the single-sender and in the general multi-sender model, respectively.[6]

The goals of this paper are twofold. First, we study feasibility results, i.e., what is possible in principle. Therefore while studying the quantities $\phi_n(d)$ and $\phi_n^*(d)$ we do not make any restriction on the use of point-to-point channels (In fact, our protocols which are optimized for the BBB usage communicate exponential number of messages over point-to-point channels and are built for succinctness of the proof, not for communication complexity.) Second, based on the obtained bounds for $\phi_n(d)$ and $\phi_n^*(d)$ we search for protocols which are both efficient in terms of the BBB and point-to-point channels usage.

1.3 Contributions of this Paper

This paper introduces the concept of broadcast amplification and proves a number of results, both feasibility results in terms of protocols as well as infeasibility results in terms of impossibility proofs.

We first study the first question mentioned above, namely the setting where the sender uses a single broadcast primitive of smaller domain. For the case of

[4] More refined versions of this question exist but will not be considered.

[5] Not necessarily exactly the number of actual bits.

[6] One could also consider a single-sender multi-shot model, i.e., the model where the sender can broadcast with BBB multiple times. Later we give a protocol for the single-sender setting which requires only a single call to BBB and is optimal even in the multi-shot model.

three parties ($n = 3$), the smallest non-trivial case, we show the quite surprising result that broadcast for any domain size d is possible if only a single 3-broadcast ($d' = 3$) is available. Moreover broadcast of a single bit ($d' = 2$) is not sufficient. In other words, $\phi_3(d) = 3$ for any $d \geq 3$.

In contrast, for $n > 3$ no broadcast amplification is possible, i.e., $\phi_n(d) = d$ for any d.

If not only the sender, but also other parties can broadcast some short messages, then (strong) broadcast amplification is possible for *any* n. We show that broadcasting $8n \log n$ bits of information in total suffices, independently of d, i.e., $\log \phi_n^*(d) \leq 8n \log n$. On the negative side, we show that at least $n - 2$ parties must broadcast at least one bit, i.e., $\min(\log d, n - 2) \leq \log \phi_n^*(d)$.

The protocol that uses $8n \log n$ bits to broadcast a value of domain size d communicates exponentially many messages over point-to-point channels. We give an optimized version of this protocol which communicates a polynomial number of messages over point-to-point channels but needs to broadcast $\mathcal{O}(n^2 \log \log d)$ bits with `BBB`.

1.4 Related Work

All known protocols for efficient multi-valued broadcast [TC84,FH06,LV11,Pat11] can be interpreted as broadcast-amplification protocols, as they actually employ an underlying broadcast scheme for short messages (besides the point-to-point channels). These protocols tolerate only $t < n/3$ or $t < n/2$, where the underlying broadcast itself is realizable with a normal broadcast protocol (hence the given broadcast channels are not needed at all).

Another approach for broadcast amplification can be derived from existing signature-based broadcast protocols [DS83,PW96]. One can use the available black-box broadcast to generate an appropriate setup (e.g., a PKI) and then use the corresponding protocol over point-to-point channels to broadcast the ℓ bit message. Thus we obtain broadcast-amplification protocols for $t < n$ with cryptographic and statistical security that require all parties to broadcast $\mathrm{Poly}(n) \log \ell$ bits in total for the construction of an ℓ-bit broadcast.

Fitzi and Maurer considered amplification of the broadcast recipient set [FM00]. That is, they showed that with the access to local broadcast among every k parties one can construct broadcast among n parties iff $t < \frac{k-1}{k+1} n$ [FM00,CFF+05].

Another related line of research is the amplification of other primitives, like OT extension [Bea96,IKNP03] or coin-toss extension [HMQU06].

In [HR13] the authors give a protocol for 3 players allowing to broadcast message of any length by broadcasting 10 bits only is given. In our notation this shows that $\log \phi_3(d) \leq 10$ for all d.

Broadcast amplification is an example of the construction of a consistency primitive from another consistency primitive as defined in [Mau04].

2 Broadcast-Amplification Model

A broadcast-amplification protocol consists of the programs $\pi_1, \ldots, \pi_n$ that the players $P_1, \ldots, P_n$ use. Each program π_i is a randomized algorithm (which takes an input from domain $\mathcal{D}$ in case of the sender's program π_1) and produces an output. The program π_i has $n-1$ interfaces to point-to-point channels to communicate with the other programs and additional interfaces to access BBB.

We now describe how the programs interact with BBB. First, we extend the notion of d-broadcast given in Section 1.2. Let (r, P_i, d)-broadcast be a broadcast channel available in round r which allows P_i to broadcast one single value from a domain of size d among the parties. We assume that each program π_j has an interface to each (r, P_i, d)-broadcast channel. Whenever we say that the parties broadcast with BBB, we mean that they actually access the corresponding broadcast channel by explicitly giving input/asking for an output on that channel's interface.

The protocol must ensure that the correct parties agree on which (r, P_i, d)-broadcast channels to invoke, that is, on r, P_i and d.[7] We say that a (r, P_i, d)-broadcast channel is *used* if the correct parties access it, i.e., in round r correct parties expect an output provided by P_i of a domain of the size d (in case of a correct P_i, he provides the corresponding input). Note that which channels are used by the protocol may not be necessarily fixed a priori and may depend on the execution. We say that a broadcast-amplification protocol has a *static* BBB usage pattern if the broadcast channels used are fixed beforehand. As opposed to the static case, protocols with a *dynamic* BBB usage pattern allow to broadcast with BBB adaptively to the execution, where of course still agreement on which broadcast channels to use is required among the correct parties.

Depending on which channels are used we distinguish the following models.

Definition 4. *The **single-sender** model allows for protocols where only (r, P_1, d)-broadcast channels are used, i.e., only P_1 broadcasts with BBB (If only one channel is used then such a single-sender model is called* single-shot*; otherwise, it is called* multi-shot*.) The **multi-sender** model does not put any limitations on the broadcast channels used.*

The costs d' of BBB usage of a broadcast-amplification protocol with a static BBB pattern is defined to be $\prod_i d_i$, where d_i's are the domain sizes of the broadcast channels used. The protocols with a dynamic BBB usage pattern have costs d' to be computed as the maximum of $\prod_i d_i$ among all possible executions.

We say that a broadcast-amplification protocol is *non-trivial* if its costs d' is strictly smaller than the size of the broadcast value domain $d = |\mathcal{D}|$, i.e., $d' < d$.

[7] This requirement stems from the observation that the broadcast channel may be implemented via a different protocol and hence in order to employ it all correct parties must start its execution together while agreeing on the broadcasting party and the domain of the broadcast value. Note that without this requirement, the BBB could be abused to reach agreement on "hidden" information, e.g., one could broadcast an ℓ-bit message v with using BBB only for a single bit (in round v).

3 Single-Sender Model

In this section we consider a single-sender model, that is, only the sender is allowed to use the BBB oracle. First, we completely investigate the situation for $n = 3$, that is, we show that 3-broadcast is enough to simulate any d-broadcast while 2-broadcast is not. On the negative side, we prove that for any $n > 3$ perfectly secure broadcast amplification is not possible, showing that $n = 3$ is a peculiar case in the context of broadcast-amplification protocols.

3.1 Broadcast Amplification for 3 Parties

We construct a broadcast-amplification protocol for three parties that allows the sender to broadcast a value v from domain $\mathcal{D}$ of size d, where the sender uses BBB to broadcast one value from a domain $\mathcal{D}'$ of size $d' = 3$. For ease of presentation, we assume that $\mathcal{D} = [d]$ and $\mathcal{D}' = [3]$.

The protocol works recursively. For $d = 3$, v is broadcast directly via BBB. For $d \geq 4$, the sender transmits v to both recipients, who then exchange the received values and forward the exchanged values back to the sender. Finally, the sender broadcasts a hint h from domain $[d-1]$, which allows each recipient to decide which of the values he holds is the right one. Broadcasting the hint is realized via recursion.[8]

The crucial trick in this protocol is the computation of the hint h. Very generically, this computation is expressed as a special function which takes as input three values (the original value v and the two values sent back to the sender) and outputs h. Given the hint h, the recipients decide on the value received from the sender if it is consistent with h. Otherwise, if the other recipient's value (as received in the exchange phase) is consistent with h, then that value is taken. Otherwise, some default value (say $\perp$) is taken.

More formally, denote the value of the sender by v; the values received by the recipients P_2 and P_3 by v_2 and v_3, respectively; the values received by the recipients in the exchange phase by v_{32} and v_{23}, respectively; and the values sent back to the sender by v_{321} and v_{231}, respectively. The function producing the hint is denoted with g_d and maps triples of values from $[d] \times [d] \times [d]$ into the hint domain $[d-1]$. Then the sender computes the hint $h = g_d(v, v_{321}, v_{231})$ and broadcasts it. Recipient P_2 outputs v_2 if $h = g_d(v_2, v_{32}, \widetilde{v_{231}})$ for some $\widetilde{v_{231}} \in [d]$. Otherwise, P_2 outputs v_{32} if $h = g_d(v_{32}, \widetilde{v_{321}}, v_2)$ for some $\widetilde{v_{321}} \in [d]$. Otherwise, P_2 outputs $\perp$. P_3 decides analogously. Clearly, this protocol guarantees validity. Consistency is achieved as long as

$$\forall v_2, v_3, \widetilde{v_{231}}, \widetilde{v_{321}} \in [d] : \; v_2 \neq v_3 \;\Rightarrow\; g_d(v_2, v_3, \widetilde{v_{231}}) \neq g_d(v_3, \widetilde{v_{321}}, v_2). \quad (1)$$

For $d \geq 4$, the function $g_d(x, y, z)$ can be constructed as follows: For $x \leq d-1$, let $g_d(x, y, z) = x$ (for any y, z). For $x = d$, let $g_d(x, y, z) = \min([d-1] \setminus \{y, z\})$. One can easily verify that g_d satisfies (1).

[8] As we see later, the recursion can be made much more efficient with the help of so-called identifying predicates. We focus on the feasibility results and hence do not optimize the protocols.

Protocol $\texttt{AmplifyBC}_3(d, v)$

1. If $d = 3$ then broadcast v using the $\texttt{BBB}$.
2. Otherwise:
   - 2.1 P_1 sends v to P_2 and P_3. Denote the values received with v_2 and v_3, respectively.
   - 2.2 P_2 sends v_2 to P_3 and P_3 sends v_3 to P_2. Denote the values received by P_2 and P_3 with v_{32} and v_{23}, respectively.
   - 2.3 P_2 sends v_{32} to P_1 and P_3 sends v_{23} to P_1. Denote the values received by v_{321} and v_{231}, respectively.
   - 2.4 P_1 computes $h = g_d(v, v_{321}, v_{231})$. Parties invoke $\texttt{AmplifyBC}_3(d-1, h)$.
   - 2.5 P_2: If there exists $\widetilde{v_{231}}$ such that $h = g_d(v_2, v_{32}, \widetilde{v_{231}})$ decide on v_2. Else if there exists $\widetilde{v_{321}}$ such that $h = g_d(v_{32}, \widetilde{v_{321}}, v_2)$ decide on v_{32}. Otherwise decide on $\perp$.
   - 2.6 P_3: If there exists $\widetilde{v_{321}}$ such that $h = g_d(v_3, \widetilde{v_{321}}, v_{23})$ decide on v_3. Else if there exists $\widetilde{v_{231}}$ such that $h = g_d(v_{23}, v_3, \widetilde{v_{231}})$ decide on v_{23}. Otherwise decide on $\perp$.

Lemma 1. *The protocol* $\texttt{AmplifyBC}_3$ *achieves broadcast. The sender* P_1 *broadcasts one value from domain* $[3]$ *via* $\texttt{BBB}$.

Proof. We prove by induction that the broadcast properties are satisfied. For $d = 3$, broadcast is achieved by assumption of $\texttt{BBB}$. Now consider $d \geq 4$:

Validity: If the sender is correct, then P_2 and P_3 receive $h = g_d(v, v_{321}, v_{231})$ as output from the recursive call to $\texttt{AmplifyBC}_3$. As $h = g_d(v_2, v_{32}, \widetilde{v_{231}})$ for $\widetilde{v_{231}} = v_{231}$, a correct P_2 decides on $v_2 = v$. Analogously, $h = g_d(v_3, \widetilde{v_{321}}, v_{23})$ for $\widetilde{v_{321}} = v_{321}$, a correct P_3 decides on $v_3 = v$.

Consistency: This property is non-trivial only if both P_2 and P_3 are correct, hence $v_{23} = v_2$ and $v_{32} = v_3$. Due to the Consistency property of the recursive call to $\texttt{AmplifyBC}_3$ both P_2 and P_3 receive the same hint h. If $v_2 = v_3$, then by inspection of the protocol both parties decide on the same value (namely on v_2 if $h \in \{g_d(v_2, v_2, \cdot), g_d(v_2, \cdot, v_2)\}$ and on $\perp$ otherwise). If $v_2 \neq v_3$, then (1) implies that if P_2 decides on v_2 (i.e., $h = g_d(v_2, v_{32}, \widetilde{v_{231}})$), then P_3 does not decide on v_3 (i.e., $h \neq g_d(v_3, \widetilde{v_{321}}, v_{23})$), but decides on $v_{23} = v_2$ (i.e., $h = g_d(v_{23}, v_3, \widetilde{v_{231}})$). Analogously, if P_3 decides on v_3, then P_2 decides on v_3 as well.

Termination: Follows by inspection.

□

3.2 Generic Structure of Impossibility Proofs

The given lower-bounds proofs employ a standard indistinguishability argument that is used to prove that certain security goals cannot be achieved by any protocol in the Byzantine environment [PSL80]. Such a proof goes by contradiction, i.e., by assuming that the security goals can be satisfied by means of some protocol $(\pi_1, \ldots, \pi_n)$. Then the programs π_i are used to build a *configuration* with

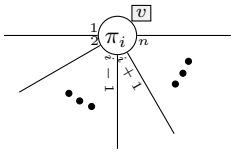

Fig. 1. Drawing of a program π_i. It has $n-1$ interfaces to bilateral channels with other players $\mathcal{P} \setminus \{P_i\}$ labeled accordingly. The program π_i is given v as input.

contradictory behavior. The configuration consists of multiple copies of π_i connected with bilateral channels and given admissible inputs. A pictorial drawing of a program in such a configuration is shown in Figure 1. When describing a configuration we will often use such a drawing accompanied with a textual description. If in the drawing an interface to a bilateral channel is not depicted then it is connected to a "null" device which simulates the program sending no messages. The interfaces to BBB are never drawn. Once the configuration is built, one simultaneously starts all the programs in the configuration and analyzes the outputs produced by the programs locally. By arguing that the view of some programs π_i and π_j in the configuration is indistinguishable from their view when run by the corresponding players P_i and P_j (while the adversary corrupts the remaining players in $\mathcal{P} \setminus \{P_i, P_j\}$) one deduces consistency conditions on the outputs by π_i and π_j that lead to a contradiction.

The main novelty of the proofs presented in this paper is that we consider an extended communication model where in addition to bilateral channels players are given access to BBB. While following the path described above, one needs to additionally define the BBB behavior in the configuration.

In the following impossibility proofs we assume that the BBB usage pattern is static. (In the full version of the paper we show how to adapt the impossibility proofs given to include protocols with a dynamic BBB usage pattern.) Furthermore, the lower bounds are given only for perfectly-secure protocols, i.e., those that fail with probability 0.

3.3 Lower Bounds in the Single-Sender Model

Lemma 2. *There is no perfectly-secure protocol among 3 parties achieving broadcast amplification for domain $\mathcal{D}$ with $|\mathcal{D}| \geq 3$ by broadcasting only 1 bit via BBB.*

Proof. Assume towards a contradiction that there is such a protocol (π_1, π_2, π_3). Without loss of generality, assume that $\mathcal{D} = [d]$ for some $d \geq 3$.

We consider the following configuration: For $i = 1, 2, 3$ and $j = 1, 2, 3$ let π_i^j be an instance of π_i. For $j = 1, 2, 3$ let π_1^j be given input j. We construct the configuration by connecting programs π_i^j as shown in Figure 2. Now we execute the programs. Whenever any program π_1^j broadcasts a bit with BBB it is given to programs π_2^j and π_3^j.

Since there are 3 programs $\pi_1^1, \pi_1^2, \pi_1^3$ broadcasting 1 bit only, there exist two of them π_1^i and π_1^j broadcasting the same bit. Without loss of generality, assume

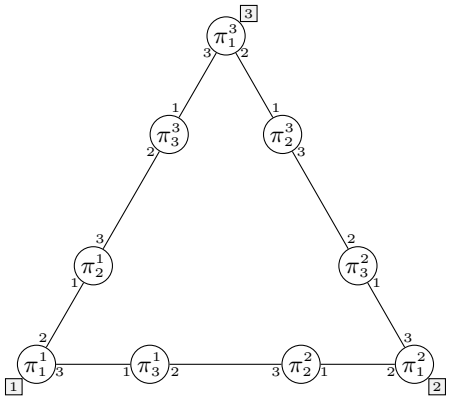

Fig. 2. The configuration for $n = 3$ to show the impossibility of broadcast amplification with broadcasting 1 bit only via BBB.

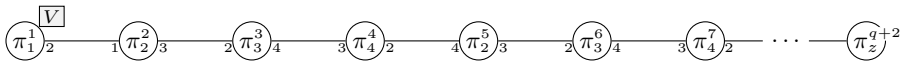

Fig. 3. The configuration for $n = 4$ to show the impossibility of non-trivial broadcast amplification with only the sender broadcasting.

that π_1^1 and π_1^2 broadcast the same bit. The configuration can be interpreted in three different ways, which lead to contradicting requirements on the outputs of the programs. (i) P_1 holds input 1 and executes π_1^1, P_3 executes π_3^1, and P_2 is corrupted and executes the remaining programs in the configuration. Due to the validity property, π_3^1 must output 1. (ii) P_1 holds input 2 and executes π_1^2, P_2 executes π_2^2, and P_3 is corrupted and executes the remaining programs in the configuration. Due to the validity property, π_2^2 must output 2. (iii) P_3 executes π_3^1, P_2 executes π_2^2, and P_1 is corrupted and executes the remaining programs in the configuration. Due to the consistency property, π_3^1 and π_2^2 must output the same value. These three requirements cannot be satisfied simultaneously, hence whatever output the programs make, the protocol (π_1, π_2, π_3) is not a perfectly-secure broadcast-amplification protocol. □

Lemma 3. *There is no perfectly-secure protocol among $n \geq 4$ parties achieving non-trivial broadcast amplification in the single-sender multi-shot model.*

Proof. We first prove the lemma for $n = 4$, then reduce the case of arbitrary $n > 4$ to $n = 4$.

(Case n = 4) Assume towards a contradiction that there exists a perfectly-secure protocol $(\pi_1, \pi_2, \pi_3, \pi_4)$ achieving non-trivial broadcast-amplification in the single-sender model in q rounds (for some $q \in \mathbb{N}$). On the highest level our proof consists of three steps. (i) we define a configuration. (ii) we show that all programs in the configuration must output the same value v. (iii) we use an

information flow argument to prove that there is a program in the configuration that does not have enough information to output v with probability 1 (this argument is inspired by [Lam83]).

(i) We consider the following configuration: Let π_i^j denote an instance of the program π_i. Consider a chain of $q+2$ programs $\pi_1^1, \pi_2^2, \pi_3^3, \pi_4^4, \pi_2^5, \pi_3^6, \ldots, \pi_z^{q+2}$ connected as shown in Figure 3. The chain is built starting with a program π_1^1 and then by repeatedly alternating copies of the programs π_2, π_3 and π_4 until the chain has $q+2$ programs. To simplify the notation we will sometimes refer to the programs in the chain without the subscript, i.e., as to $\pi^1, \pi^2, \ldots, \pi^{q+2}$. Let π_1^1 be given as input a uniform random variable V chosen from domain $\mathcal{D}$. Now we execute the programs. Whenever π_1^1 uses `BBB` to broadcast some x, the value x is given to all programs in the configuration.

(ii) First, we prove that any pair of connected recipients' programs (π_i^a, π_j^{a+1}) $(a \geq 2)$ in the chain output the same value. One can view the configuration as the player P_i running the program π_i^a and P_j running π_j^{a+1} while the adversary corrupting $\{P_1, P_2, P_3, P_4\} \backslash \{P_i, P_j\}$ is simulating the programs $\pi^1, \ldots, \pi^{a-1}$ and $\pi^{a+2}, \ldots, \pi^{q+2}$. Due to the consistency property, π_i^a and π_j^{a+1} must output the same value. Since every connected pair of the recipients' programs in the chain outputs the same value, then the programs $\pi^2, \ldots, \pi^{a+1}$ in the configuration output the same value. Moreover, the configuration can be viewed as P_1 executing π_1^1, P_2 executing π_2^2 while the adversary who corrupts $\{P_3, P_4\}$ is simulating the remaining chain. Due to the validity property, π_2^2 must output V. Finally, each recipient's program $\pi^2, \ldots, \pi^{q+2}$ in the chain outputs V.

(iii) Let S_i^r be a random variable denoting the state of the program π^i in the chain after r rounds of the protocol execution. By state we understand the input that the program has, the set of all messages that the program received up to the r^{th} round over point-to-point channels and on the `BBB`'s interface together with the random coins it has used. Let B^r be a random variable denoting the list of the values that have been broadcast with `BBB` up to the r^{th} round.

After r rounds only programs $\pi^1, \pi^2, \ldots, \pi^{r+1}$ can receive full information about V. The remaining programs in the chain $\pi^{r+2}, \pi^{r+3}, \ldots, \pi^{q+2}$ can receive only the information that was distributed with `BBB`, i.e., the information contained in B^r. That is, one can verify by induction that for any r and for all $i \geq r+2$ holds $I(V; S_i^r | B^r) = 0$. Hence, for the last program in the chain π^{q+2} after q rounds of computation it holds that $I(V; S_{q+2}^q | B^q) = 0$ and hence $I(V; S_{q+2}^q) \leq H(B^q)$. Because we assumed that the protocol achieves non-trivial broadcast-amplification we have that $H(B^q) < H(V)$. Combining these facts we get that $I(V; S_{q+2}^q) < H(V)$. Hence, the last program π^{q+2} cannot output V with probability one, a contradiction.

(Case n > 4) Assume towards a contradiction that there is a protocol $(\pi_1, \pi_2, \pi_3, \ldots, \pi_n)$ allowing to do broadcast amplification in the single-sender model. One particular strategy of the adversary is to corrupt parties $P_5, \ldots, P_n$ and make them not execute their corresponding programs $\pi_5, \ldots, \pi_n$. Still, the remaining protocol $(\pi_1, \pi_2, \pi_3, \pi_4)$ must achieve broadcast, which contradicts the first case. □

3.4 Summary

Theorem 1. *If $n = 3$ then $\forall d \geq 3\ \phi_3(d) = 3$; otherwise, if $n > 3$ then $\forall d\ \phi_n(d) = d$.*

The first statement follows from combining Lemma 1 and Lemma 2. The second statement follows from Lemma 3.

4 Multi-Sender Model

As we have seen in the previous section in the single-sender model no broadcast-amplification is achievable for $n \geq 4$. In this section we consider a generalization of this model by allowing recipients to broadcast with BBB as well. In such a model we show that broadcast-amplification is achievable for any n. Moreover, we prove that in order to achieve a non-trivial broadcast-amplification for arbitrary n essentially all recipients must broadcast with BBB.

4.1 Broadcast Amplification for n Parties

In this section we present a broadcast-amplification protocol for n parties, where the parties broadcast with BBB at most $8n \log n$ bits in total. We first introduce the notion of identifying predicates and give an efficient construction of them. Then we present a protocol for graded broadcast, which achieves only a relaxed variant of broadcast, but only requires the sender to use BBB. Finally, we give the main broadcast-amplification protocol, which uses graded broadcast and BBB (by each party) to achieve broadcast.

While the presented protocol is very efficient in terms of the BBB usage (it broadcasts via BBB only $8n \log n$ bits to achieve broadcast of any ℓ bits), it communicates exponentially many messages over authenticated channels. We then show how to optimize this protocol such that it communicates only a polynomial (in n) number of messages at the expense of a higher BBB usage.

Identifying Predicates. An identifying predicate allows to identify a specific element v from some small subset $S \subseteq \mathcal{D}$, where $\mathcal{D}$ is a potentially large domain. To our knowledge, this concept has been firstly introduced in [HR13].

Definition 5. *A* c-identifying predicate *for domain $\mathcal{D}$ is a family of functions $Q_{k \in \mathcal{K}} : \mathcal{D} \to \{0,1\}$ such that for any $S \subseteq \mathcal{D}$ with $|S| \leq c$ and any value $v \in S$ there exists a key $k \in \mathcal{K}$ with $Q_k(v) = 1$ and $Q_k(v') = 0$ for all $v' \in S \setminus \{v\}$. We say that such v is* uniquely identified *by Q_k in S.*

Note that any identifying predicates Q_k achieve monotonicity in the following sense:

Lemma 4. *If v is uniquely identified by Q_k in S, then v in uniquely identified in any $S' \subseteq S$ with $v \in S'$.*

The goal of constructing an identifying predicate family is to have $|\mathcal{K}|$ as small as possible given c and $|\mathcal{D}|$. We give a construction of a c-identifying predicate with domain $\mathcal{D}$ below.

Polynomial-based identifying predicate construction. Let $\ell = \log|\mathcal{D}|$. For $\kappa \in \mathbb{N}$, let any value $v \in \mathcal{D}$ be interpreted as a polynomial f_v over $\mathrm{GF}(2^\kappa)$ of degree at most $\lfloor \ell/\kappa \rfloor$. We find a point $x \in \mathrm{GF}(2^\kappa)$ such that $f_v(x)$ is different from all other values $f_{v'}(x)$ for $v' \in S \setminus \{v\}$. For such a point x to always exist we need that the total number of points in the field is larger than the number of points in which f_v may coincide with other polynomials $f_{v'}$, i.e., $2^\kappa > (c-1)\lfloor \ell/\kappa \rfloor$. To satisfy this condition, it is enough to choose $\kappa := \lceil \log(c\ell) \rceil$. The key for the identifying predicate is defined as $k = (x, f_v(x))$, which is encoded using $2\lceil \log(c\ell) \rceil$ bits.[9] The predicate is defined as follows:

$$Q_{(x,y)}(v) = \begin{cases} 1, \text{ if } f_v(x) = y; \\ 0, \text{ otherwise.} \end{cases}$$

Lemma 5. *The polynomial-based construction gives a c-identifying predicate Q with domain $\mathcal{D}$ and key space $\mathcal{K}_c^{\mathcal{D}} = \{0,1\}^{2\lceil \log(c \log|\mathcal{D}|) \rceil}$.*

Graded Broadcast. Graded broadcast (a.k.a. gradecast) was introduced by Feldman and Micali [FM88]. It allows to broadcast a value among the set of recipients but with weaker consistency guarantees. In addition to the value v_i each recipient P_i also outputs a grade g_i describing the level of agreement reached by the players. In this paper we extend the original gradecast definition [FM88] with a more flexible grading system:

Definition 6. *A protocol achieves graded broadcast if it allows the sender P_1 to distribute a value v among parties $\mathcal{R}$ with every party P_i outputting a value v_i with a grade $g_i \in [n]$ such that:*

Validity: *If the sender P_1 is correct, then every correct $P_i \in \mathcal{R}$ outputs $(v_i, g_i) = (v, 1)$.*

Graded Consistency: *If a correct $P_i \in \mathcal{R}$ outputs (v_i, g_i) with $g_i < n$, then every correct $P_j \in \mathcal{R}$ outputs (v_j, g_j) with $v_j = v_i$ and $g_j \leq g_i + 1$.*

Termination: *Every correct party in $\mathcal{P}$ terminates.*

Intuitively, the grade can be understood as the consistency level achieved. The "strongest" grade $g_i = 1$ means that from the point of view of P_i, the sender "looks correct". Grade $g_i = 2$ means that P_i actually knows that the sender is incorrect; however, there might be an honest P_j for whom the sender looks correct. Grade $g_i = 3$ means that P_i knows that the sender is incorrect and every honest P_j knows so, too; however, there might be an honest P_k who does not know that every honest P_j knows that the sender is incorrect. And so on till the "weakest" grade $g_i = n$.

The protocol proceeds as follows: The sender sends the value v he wants to broadcast to all parties, who then exchange the received value(s) during $2n$

[9] Such a point x can be efficiently found by random sampling elements in $\mathrm{GF}(2^\kappa)$. Indeed, for $\kappa = \lceil \log(c\,\ell) \rceil$ more than half of the elements in $\mathrm{GF}(2^\kappa)$ are points where f_v is different from all other $f_{v'}$.

rounds. That is, in every round each party sends the set of values received so far to every other party. In this way each recipient P_i forms a growing sequence of sets $M_i^1 \subseteq M_i^2 \subseteq \cdots \subseteq M_i^{2n}$ (the set M_i^r represents the set of all messages received by P_i up to the round r). Finally, the sender distributes a hint consisting of the key k for an identifying predicate Q_k that should identify v among the values that the recipients hold. Then each recipient P_i computes his grade g_i to be the smallest number in $[n]$ such that both $M_i^{g_i}$ and $M_i^{2n-g_i}$ contain a uniquely identified message. There could be only one value v_i uniquely identified in both sets since $M_i^{g_i} \subseteq M_i^{2n-g_i}$. Then P_i outputs v_i with the grade g_i. Clearly, if the sender is correct, then each correct recipient outputs $g_i = 1$. Otherwise, since for every pair P_i, P_j of correct recipients it holds that $M_i^{g_i} \subseteq M_j^{g_i+1}$ and $M_j^{2n-(g_i+1)} \subseteq M_i^{2n-g_i}$ we have $g_j \leq g_i + 1$.

Let us detail the step when sender distributes his hint k. While it can be done directly with the help of `BBB` (which would lead to a less efficient construction), we let the parties to invoke gradecast recursively for the distribution of k. Once each player P_i outputs a key k_i with a grade g_i' he uses k_i as a hint. Then the final grade is computed by P_i as the maximum of two grades g_i and g_i', i.e., it is computed as the "weakest" grade among the two.

Protocol `GradedBC`$(P_1, \mathcal{D}, v)$

1. If $|\mathcal{D}| \leq |\mathcal{K}^{\mathcal{D}}_{n^{2n}}|$ then P_1 broadcasts v using `BBB`, and every $P_i \in \mathcal{R}$ outputs $(v, 1)$.
2. Otherwise:
   - 2.1 Sender P_1: Set $M_1^0 := \{v\}$. $\forall P_i \in \mathcal{R}$: Set $M_i^0 := \emptyset$.
   - 2.2 For $r = 0, \ldots, 2n-1$:
     $\forall P_i \in \mathcal{P}$: Send M_i^r (of size at most n^r) to all $P_j \in \mathcal{P}$, P_j denotes the union of the received sets with M_j^{r+1}, i.e., $M_j^{r+1} = \bigcup_i M_i^r$.
   - 2.3 Sender P_1: Choose a key k for the n^{2n}-identifying predicate Q with domain $\mathcal{D}$, the set of values M_1^{2n} and the value v.
   - 2.4 Players $\mathcal{P}$ invoke `GradedBC`$(P_1, \mathcal{K}^{\mathcal{D}}_{n^{2n}}, k)$ recursively. Let (k_i, g_i') denote the output of $P_i \in \mathcal{R}$.
   - 2.5 $\forall P_i \in \mathcal{R}$: Let g be the smallest number in $[n]$ such that there exists u which is uniquely identified by Q_{k_i} in M_i^g and in M_i^{2n-g}. Output $(v_i, g_i) = (u, \max(g, g_i'))$. If such g does not exist output $(v_i, g_i) = (\perp, n)$;

Lemma 6. *The protocol `GradedBC` achieves graded broadcast while requiring only the sender to use `BBB` to broadcast one value of at most $\lceil 7n \log n \rceil$ bits.*

Proof. We prove by induction that graded broadcast is achieved. For $|\mathcal{D}| \leq |\mathcal{K}^{\mathcal{D}}_{n^{2n}}|$, graded broadcast is achieved by assumption of `BBB`. For $|\mathcal{D}| > |\mathcal{K}^{\mathcal{D}}_{n^{2n}}|$:

Validity: If the sender is correct then he selects a key k for the n^{2n}-identifying predicate Q_k such that only his value v is identified by Q_k in M_1^{2n}. All correct players get $(k, 1)$ as output from the recursive call to `GradedBC` (due to the Validity property of the recursive `GradedBC`). Since for every correct player

P_i it holds that $M_i^1 \subseteq M_i^{2n-1} \subseteq M_1^{2n}$ and $v \in M_i^1$ this implies that v is uniquely identified in M_i^1 and in M_i^{2n-1}. Hence P_i computes $v_i = v$ and $g_i = 1$.

Graded Consistency: Let P_i denote a correct recipient outputting the smallest grade g_i. If $g_i = n$ then Graded Consistency holds trivially. Now assume that $g_i < n$, and hence $g_i' < n$. Consider any other correct recipient P_j. Due to the Graded Consistency property of the recursive `GradedBC`, the fact that $g_i' < n$ implies that P_i and P_j have the same keys k_i and k_j which we denote with k. Observe that $M_i^{g_i} \subseteq M_j^{g_i+1} \subseteq M_j^{2n-(g_i+1)} \subseteq M_i^{2n-g_i}$. The value v_i is uniquely identified by Q_k in both $M_i^{g_i}$ and $M_i^{2n-g_i}$, hence v_i is uniquely identified in both $M_j^{g_i+1}$ and $M_j^{2n-(g_i+1)}$. Hence the grade $g_j \in \{g_i, g_i+1\}$. If $g_j = g_i + 1$ then $v_j = v_i$. If $g_j = g_i$ then, since $M_j^{g_i} \subseteq M_j^{g_i+1}$ and v_i is uniquely identified in $M_j^{g_i+1}$, the only value that can be uniquely identified by Q_k in $M_j^{g_i}$ is v_i. This implies that $v_j = v_i$.

Termination: Follows by inspection.

It remains to prove the stated usage complexity of `BBB`. Note that `BBB` is only used at the deepest recursion level. We denote the logarithm of broadcast domain size at the r^{th} recursive level to be ℓ_r. We have that $\ell_0 = \log|\mathcal{D}|$ and ℓ_{i+1} is defined recursively to be $2\lceil\log(n^{2n}\ell_i)\rceil$. It can be verified that $\ell_{i+1} < \ell_i$ for any $\ell_i > 7n\log n$. Hence, the sender P_1 broadcasts with `BBB` at most $\lceil 7n\log n\rceil$ bits. □

Main Protocol. The broadcast-amplification protocol first invokes graded broadcast. Then, each party broadcasts his grade (using `BBB`), and decides depending on the grades broadcast whether to use the output of graded broadcast or to use some default value (say $\perp$) as output.

The core idea of the protocol lies in the analysis of the grades broadcast. Denote the set of all grades by $G = \{g_i\}_i$. As $|\mathcal{R}| = n - 1$, there exists a grade $g \in [n]$ with $g \notin G$. Consider the smallest grade g_i of an honest party P_i. If $g_i > g$, then clearly the grade g_j of each honest party P_j is $g_j > g$. On the other hand, if $g_i < g$, then by the definition of graded broadcast, the grade g_j of any honest party P_j is $g_j \leq g_i + 1$, hence $g_j < g$. In other words, either the grades of all honest parties are below g, or the grades of all honest parties are above g. In the former case, every honest party P_i has $g_i < n$ and hence all values v_i are equal (and are a valid output of broadcast). In the latter case, no honest party P_i has grade $g_i = 1$, hence the recipients can output some default value $\perp$.

Protocol $\texttt{AmplifyBC}_n(P_1, \mathcal{D}, v)$

1. Players $\mathcal{P}$ invoke $\texttt{GradedBC}(P_1, \mathcal{D}, v)$, let (g_i, v_i) denote the output of P_i.
2. $\forall P_i \in \mathcal{R}$: Broadcast g_i using `BBB`. Let G denote the set of all g_i broadcast.
3. $\forall P_i \in \mathcal{R}$: Let $g = \min([n] \setminus G)$. If $g_i < g$, then decide on v_i, otherwise decide on $\perp$.

Lemma 7. *The protocol* $\texttt{AmplifyBC}_n$ *achieves broadcast and requires the sender to broadcast with* BBB *one value of at most* $\lceil 7n \log n \rceil$ *bits and each of the recipients to broadcast one value from domain* $[n]$*. In total at most* $8n \log n$ *bits need to be broadcast via* BBB*.*

Proof. We show that each of the broadcast properties are satisfied:

Validity: If the sender is correct then all correct parties get $(v, 1)$ as an output from GradedBC and decide on v.

Consistency: Let $g = \min([n] \setminus G)$, and let P_i denote a correct recipient outputting the smallest grade g_i. If $g_i < g$, then clearly $g_i < n$, and all honest parties P_j hold the same value $v_j = v_i$ and grade $g_j \leq g_i + 1$. As $g_j \neq g$, it follows $g_j < g$. Hence, every honest party P_j outputs $v_j = v_i$. On the other hand, if $g_i > g$, then every honest party P_j holds $g_j > g$ and outputs $\perp$.

Termination: Follows by inspection.

It remains to prove the stated usage complexity of BBB. The protocol $\texttt{AmplifyBC}_n$ requires the sender to broadcast one value of at most $\lceil 7n \log n \rceil$ bits during the GradedBC invocation (cf. Lemma 6). Furthermore, each recipient broadcasts the grade (of domain $[n]$) using BBB. This sums up to $8n \log n$ bits overall. □

Efficient Protocol. The main disadvantage of the protocol $\texttt{AmplifyBC}_n$ is that the underlying gradecast protocol GradedBC requires exponential message communication. Here we briefly sketch how one can achieve polynomial communication complexity in GradedBC at the cost of higher BBB usage. The main idea of the optimized protocol $\texttt{GradedBC}^+$ is to allow recipients to use BBB such that they can filter out messages from the sets M_i^r. Roughly speaking, if a recipient holds a set of messages M_i^r then he broadcasts a "challenge" forcing the sender in his response to invalidate at least all but one values in M_i^r. After each of the recipients has his set M_i^r filtered, recipients continue exchanging sets consisting of at most one element. The detailed description of this protocol and its analysis is given in Appendix A.

Lemma 8. *The protocol* $\texttt{AmplifyBC}_n$ *with the underlying gradecast implementation by* $\texttt{GradedBC}^+$ *allows to broadcast an* ℓ*-bit message while broadcasting* $\mathcal{O}(n^2 \log \ell)$ *bits with* BBB *and communicating* $\mathcal{O}(n^3 \ell)$ *bits over point-to-point channels.*

4.2 Lower Bounds in the Multi-Sender Model

Based on the approach presented in Section 3.2 we investigate the lower bounds on the broadcast-amplification protocols in the multi-sender model. As it was shown for the single-sender model there is no broadcast-amplification possible when only the sender uses BBB for $n \geq 4$. We extend this result by showing that the sender and at least all but 2 recipients are required to broadcast some information via BBB to achieve non-trivial broadcast-amplification.

Fig. 4. The configuration for $n = 3$ to show that the sender must use BBB to broadcast at least one bit.

Lemma 9. *Every perfectly-secure broadcast-amplification protocol for domain $\mathcal{D}$ requires the sender P_1 to broadcast at least 1 bit via BBB.*

Proof. We first prove the theorem for $n = 3$, then reduce the case of arbitrary $n > 3$ to $n = 3$.

(Case n = 3) Assume towards a contradiction that there is a protocol (π_1, π_2, π_3) allowing the parties P_1, P_2, P_3 to do broadcast amplification, where the sender does not broadcast with BBB. We consider the following configuration: Let π_1^u and π_1^v denote two instances of the program π_1, where π_1^u is given input u and π_1^v is given input v for $u, v \in \mathcal{D}$ and $u \neq v$. We connect programs π_1^u, π_2, π_3 and π_1^v with bilateral channels as shown in Figure 4. Now we execute the programs. Whenever π_2 or π_3 use BBB to broadcast some x, the value x is given to all programs.

The configuration can be interpreted in three different ways, which lead to contradicting requirements on the outputs of the programs. (i) P_1 holds input u and executes π_1^u, P_2 executes π_2, and P_3 is corrupted and executes π_3 and π_1^v. Due to the validity property, π_2 must output u. (ii) P_1 holds input v and executes π_1^v, P_3 executes π_3, and P_2 is corrupted and executes π_2 and π_1^u. Due to the validity property, π_3 must output v. (iii) P_2 executes π_2, P_3 executes π_3, and P_1 is corrupted and executes π_1^u and π_1^v. Due to the consistency property, π_2 and π_3 must output the same value. These three requirements cannot be satisfied simultaneously, hence whatever output the programs make, the protocol (π_1, π_2, π_3) is not a perfectly-secure broadcast-amplification protocol.

(Case n > 3) Assume towards a contradiction that there is a protocol $(\pi_1, \pi_2, \pi_3, \ldots, \pi_n)$ allowing to do broadcast amplification where the sender does not broadcast with BBB. One particular strategy of the adversary is to corrupt parties $P_4, \ldots, P_n$ and make them not execute their corresponding programs $\pi_4, \ldots, \pi_n$. Still, the remaining protocol (π_1, π_2, π_3) must achieve broadcast, which contradicts the first case. □

Lemma 10. *Every perfectly-secure non-trivial broadcast-amplification protocol requires that at least all but 2 of the recipients broadcast at least 1 bit with BBB.*

Proof. Assume towards a contradiction that there is a protocol $(\pi_1, \pi_2, \pi_3, \ldots, \pi_n)$ allowing to do non-trivial broadcast amplification with three recipients' programs not broadcasting with BBB. Without loss of generality, assume that these programs are π_2, π_3, π_4.[10] One particular strategy of the adversary is to corrupt

[10] Such not broadcasting programs are fixed because we considered protocols with static BBB usage pattern.

parties $P_5, \ldots, P_n$ and make them not execute their corresponding programs $\pi_5, \ldots, \pi_n$. The programs $\pi_1, \pi_2, \pi_3, \pi_4$ of the remaining honest players can then put the values sent and broadcast by the corrupted parties to some default value (say $\perp$). The remaining protocol $(\pi_1, \pi_2, \pi_3, \pi_4)$ achieves non-trivial broadcast amplification, which contradicts Lemma 3. □

4.3 Summary

The following theorem summarizes results obtained in this section (The proof of this theorem follows from Lemmas 7, 9 and 10.)

Theorem 2. *For all n, d we have* $8n \log n \geq \log \phi_n^*(d) \geq \min(\log d, n-2)$.[11]

Additionally, we give an efficient protocol that allows to broadcast an ℓ-bit value while broadcasting $\mathcal{O}(n^2 \log \ell)$ bits with `BBB` and communicating $\mathcal{O}(n^3 \ell)$ bits over point-to-point channels.

5 Conclusions

Broadcast amplification is the task of achieving d-broadcast given point-to-point channels and access to a d'-broadcast primitive, for $d' < d$. The existence of such a broadcast-amplification protocol means in a certain sense that d-broadcast and d'-broadcast are equivalent (respectively that d'-broadcast is "as good as" d-broadcast).

It is well known that perfectly-secure broadcast cannot be constructed from point-to-point channels when the number of cheaters is not limited. In this paper, we have shown that:

- For three parties, 3-broadcast and d-broadcast are equivalent for any $d \geq 3$. However, 2-broadcast and 3-broadcast are not equivalent.
- For an arbitrary number of parties, $(8n \log n)$-bit broadcast and ℓ-bit broadcast are equivalent for any $\ell \geq 8n \log n$. However, for $n \geq 4$ parties, $(n-3)$-bit broadcast and ℓ-bit broadcast are not equivalent for large enough ℓ.

In summary, for three parties, we have given a complete picture of equivalence of broadcast primitives for different domains, under the assumption that point-to-point channels are freely available. For $n \geq 4$ parties, we have proved a lower bound and an upper bound on the broadcast primitive necessary for broadcasting arbitrary messages, namely $\Omega(n)$ and $\mathcal{O}(n \log n)$ bits, respectively.

[11] The last inequality combines the facts that any non-trivial broadcast amplification protocol broadcasts at least $n-2$ bits, whereas the trivial protocol always uses $\log d$ bits.

References

[Bea96] D. Beaver. Correlated pseudorandomness and the complexity of private computations. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (STOC '96)*, pages 479–488. ACM, 1996.

[BGP92] P. Berman, J. A. Garay, and K. J. Perry. Bit optimal distributed consensus. In *Computer Science Research*, pages 313–322. Plenum Publishing Corporation, New York, NY, USA, 1992.

[CFF+05] J. Considine, M. Fitzi, M. Franklin, L. A. Levin, U. Maurer, and D. Metcalf. Byzantine agreement given partial broadcast. *Journal of Cryptology*, 18(3):191–217, July 2005.

[CW92] B. A. Coan and J. L. Welch. Modular construction of a byzantine agreement protocol with optimal message bit complexity. *Information and Computation*, 97:61–85, March 1992.

[DS83] D. Dolev and H. R. Strong. Authenticated algorithms for Byzantine agreement. *SIAM Journal on Computing*, 12(4):656–666, 1983.

[FH06] M. Fitzi and M. Hirt. Optimally efficient multi-valued Byzantine agreement. In *Proceedings of the 26th annual ACM symposium on Principles of distributed computing, PODC '06*, pages 163–168, New York, NY, USA, 2006. ACM.

[FM88] P. Feldman and S. Micali. Optimal algorithms for byzantine agreement. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, STOC '88, pages 148–161, New York, NY, USA, 1988. ACM.

[FM00] M. Fitzi and U. Maurer. From partial consistency to global broadcast. In F. Yao, editor, *Proc. 32nd ACM Symposium on Theory of Computing — STOC 2000*, pages 494–503. ACM, May 2000.

[HMQU06] D. Hofheinz, J. Müller-Quade, and D. Unruh. On the (Im-)Possibility of extending coin toss. In *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 504–521. Springer, 2006.

[HR13] M. Hirt and P. Raykov. On the complexity of broadcast setup. In F. V. Fomin, R. Freivalds, M. Z. Kwiatkowska, and D. Peleg, editors, *ICALP (1)*, volume 7965 of *Lecture Notes in Computer Science*, pages 552–563. Springer, 2013.

[IKNP03] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. Extending oblivious transfers efficiently. In D. Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 145–161. Springer, 2003.

[Lam83] L. Lamport. The weak byzantine generals problem. *J. ACM*, 30(3):668–676, 1983.

[LV11] G. Liang and N. Vaidya. Error-free multi-valued consensus with Byzantine failures. In *Proceedings of the 30th annual ACM symposium on Principles of distributed computing*, PODC '11, pages 11–20, New York, NY, USA, 2011. ACM.

[Mau04] U. M. Maurer. Towards a theory of consistency primitives. In R. Guerraoui, editor, *DISC*, volume 3274 of *Lecture Notes in Computer Science*, pages 379–389. Springer, 2004.

[Pat11] A. Patra. Error-free multi-valued broadcast and Byzantine agreement with optimal communication complexity. In *Proceedings of the 15th international conference on Principles of Distributed Systems*, OPODIS '11, pages 34–49. Springer, 2011.

[PSL80] M. C. Pease, R. E. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, 1980.

[PW96] B. Pfitzmann and M. Waidner. Information-theoretic pseudosignatures and Byzantine agreement for $t \geq n/3$. Technical report, IBM Research, 1996.

[TC84] R. Turpin and B. A. Coan. Extending binary Byzantine agreement to multivalued Byzantine agreement. *Information Processing Letters*, 18(2):73–76, 1984.

A A Broadcast-Amplification Protocol with Polynomial Number of Messages

In this section we present an optimized implementation $\texttt{GradedBC}^+$ of the graded broadcast protocol `GradedBC`. We first introduce the notion of resolution functions (which is closely related to identifying predicates) and give an efficient construction of them. Then we describe the optimized protocol $\texttt{GradedBC}^+$ that communicates polynomially many messages over point-to-point channels. After substituting `GradedBC` with $\texttt{GradedBC}^+$ in $\texttt{AmplifyBC}_n$ we get a broadcast-amplification protocol with the following properties.

Lemma 8. *The protocol $\texttt{AmplifyBC}_n$ with the underlying gradecast implementation by $\texttt{GradedBC}^+$ allows to broadcast an ℓ-bit message while broadcasting $\mathcal{O}(n^2 \log \ell)$ bits with `BBB` and communicating $\mathcal{O}(n^3 \ell)$ bits over point-to-point channels.*

A.1 Resolution Functions

An identifying predicate allows to identify a specific element v from some set S of potentially a large domain $\mathcal{D}$. Resolution functions extend this notion by providing a collision-free way of choosing one of the values in S while explicitly not choosing the others.

Definition 7. *A* c-resolution function *for domain $\mathcal{D}$ and range $\mathcal{Y}$ is a family of functions $F_{k \in \mathcal{K}} : \mathcal{D} \to \mathcal{Y}$ such that for any $S \subseteq \mathcal{D}$ with $|S| \leq c$ there exists a key $k \in \mathcal{K}$ with $F_k(v) \neq F_k(v')$ for any $v \neq v'$ from S. Such a key k is said to* resolve *the set S.*

We say that v is *identified* by a pair (k, y) in S if $F_k(v) = y$ (trivially, only one value can be identified if k resolves the set S). The goal of constructing such a function F is to have $|\mathcal{K}|$ and $|\mathcal{Y}|$ as small as possible given c and $|\mathcal{D}|$. We give a construction of a c-resolution function with domain $\mathcal{D}$ below.

Polynomial-based resolution function construction. This construction is very similar to the polynomial-based construction for identifying predicates presented before. Let $\ell = \log |\mathcal{D}|$. Consider any set $S \subseteq \mathcal{D}$ with $|S| \leq c$. For $\kappa \in \mathbb{N}$, let any value $v \in \mathcal{D}$ be interpreted as a polynomial f_v over $\mathrm{GF}(2^\kappa)$ of degree at most $\lfloor \ell/\kappa \rfloor$. We find a point $x \in \mathrm{GF}(2^\kappa)$ such that $f_v(x) \neq f_{v'}(x)$ for any two

$v \neq v' \in S$. For such a point x to always exist we need that the total number of points in the field is larger than the number of points in which any f_v may coincide with other polynomials $f_{v'}$, i.e., $2^\kappa > \frac{c(c-1)}{2}\lfloor \ell/\kappa \rfloor$. To satisfy this condition, it is enough to choose $\kappa := \lceil \log(c^2\ell) \rceil$. The resolution function is defined as $F_x(v) = f_v(x)$ with the key space and range space being $\mathrm{GF}(2^\kappa)$. So, the key and the value of the resolution function can be encoded using $\lceil \log(c^2\ell) \rceil$ bits.

Lemma 11. *The polynomial-based construction gives a c-resolution function F for domain $\mathcal{D}$ with key and range spaces $\mathcal{K}_c^{\mathcal{D}} = \mathcal{Y}_c^{\mathcal{D}} = \{0,1\}^{\lceil \log(c^2 \log |\mathcal{D}|) \rceil}$.*

A.2 The Optimized Graded Broadcast Protocol

The protocol proceeds as follows: The sender sends the value v he wants to broadcast among all recipients $\mathcal{R}$, who then exchange the received value(s) during $n-1$ rounds. In each round r each party P_i sends the value it currently holds (denoted with v_i^r) to every other party and forms the set of at most n received values. Then each party broadcasts a key k_i for an n-resolution function which resolves the set of the values received. In the end of the round the sender broadcasts the values $y_1, \ldots, y_n$ of the resolution function for the keys $k_1, \ldots, k_n$ so that each of the recipients keeps at most one value identified by all (k_i, y_i). Finally, each recipient P_i decides on the grade g_i to be the first "stable" round starting from which the value he holds remain unchanged, i.e., $v_i^{g_i} = v_i^{g_i+1} = \cdots = v_i^n$.

Protocol $\texttt{GradedBC}^+(P_1, \mathcal{D}, v)$:

1. Sender P_1: Send v to every $P_i \in \mathcal{R}$.
   $\forall P_i \in \mathcal{R}$: Denote the message received from the sender by v_i^1.
r. In each step $r = 2, \ldots, n$, execute the following sub-steps:
   $r.1$ $\forall P_i \in \mathcal{R}$: Send the value v_i^{r-1} to all $P_j \in \mathcal{R}$, P_j denotes the set of the received values with M_j^r.
   $r.2$ $\forall P_i \in \mathcal{R}$: Choose a key k_i^r for an n-resolution function F with domain $\mathcal{D}$, that resolves the set of values S_i^r. Broadcast the key k_i^r using the BBB.
   $r.3$ Sender P_1: Broadcast a list of values $(F_{k_2^r}(v), \ldots, F_{k_n^r}(v))$ using the BBB. Denote the list broadcast with $(y_2^r, \ldots, y_n^r)$.
   $r.4$ $\forall P_i \in \mathcal{R}$: Select v_i^r to be some $u \in M_i^r$ such that u is identified by (k_j^r, y_j^r) for all j; set v_i^r to $\perp$ if no such u exists.
$n+1$. $\forall P_i \in \mathcal{R}$: Compute g_i to be the smallest step r such that $v_i^r = v_i^{r+1} = \cdots = v_i^n$. Output (v_i^n, g_i).

Lemma 12. *The protocol $\texttt{GradedBC}^+$ achieves graded broadcast while requiring $\mathcal{O}(n^2 \log \ell)$ bits to be broadcast with BBB and communicating $\mathcal{O}(n^3\ell)$ bits over point-to-point channels.*

Proof. We show that each of the graded broadcast properties is satisfied:

VALIDITY: If the sender is correct then for any key k he broadcasts $y = F_k(v)$ such that only his value v is chosen by correct recipients at every iteration. Hence each correct P_i computes $v_i = v$ and $g_i = 1$.

GRADED CONSISTENCY: Let P_i denote a correct recipient outputting the smallest grade g_i. If $g_i = n$ then Graded Consistency holds trivially. Now assume that $g_i < n$. Consider any other correct recipient P_j. Observe that $v_i^{g_i} \in M_j^{g_i+1}$. Since P_i kept the value $v_i^{g_i}$ till the round n it implies that $v_i^{g_i}$ is identified in S_i^r by all pairs (k_a^r, y_a^r) for all a and $r \geq g_i$. Hence, $v_i^{g_i}$ is identified in $S_j^{g_i+1}, S_j^{g_i+2}, \ldots, S_j^n$ by all pairs (k_a^r, y_a^r) for all a and $r \geq g_i + 1$. Moreover, since P_j chose the keys k_j^r for a resolution function faithfully, only a unique value can be identified in $S_j^{g_i+1}, S_j^{g_i+2}, \ldots, S_j^n$. Since only a unique value can be identified then P_j outputs $v_j = v_i^{g_i}$ with the grade $g_j \leq g_i + 1$.

TERMINATION: Follows by inspection.

It remains to prove the stated usage complexity of `BBB`. At each step $r = 2, \ldots, n$ of the protocol `GradedBC`$^+$, every recipient P_i broadcasts a key k_i^r for a family of n-resolution functions and the sender broadcasts a list of n values of the function. If the polynomial-based construction of the resolution function is used then each key and a value of function consists of $\lceil \log(n^2\ell) \rceil$ bits. Since in total the protocol works in $n-1$ rounds we broadcast $2(n-1)^2\lceil \log(n^2\ell) \rceil$ bits with `BBB`. This expression can be rewritten as $\mathcal{O}(n^2(\log n + \log \ell))$. We can assume that $\ell > n$, since for $\ell \leq n$ it is easier to run the trivial algorithm that broadcasts a message bit by bit. Summing up the analysis above, we have that the total number of the `BBB` invocations during the protocol run is $\mathcal{O}(n^2 \log \ell)$.

The communication costs of `GradedBC`$^+$ over authenticated channels consist of distributing $n-1$ ℓ-bit messages during Step 1 and exchanging of $(n-1)^2(n-1)$ ℓ-bit messages during Steps $2, \ldots, n$. Hence, the total number of bits that need to be communicated is $\mathcal{O}(n^3\ell)$. □