Broadcast Secret-Sharing, Bounds and Applications

Ivan Bjerre Damgård ✉
Dept. of Computer Science, Aarhus University, Denmark

Kasper Green Larsen ✉
Dept. of Computer Science, Aarhus University, Denmark

Sophia Yakoubov ✉
Dept. of Computer Science, Aarhus University, Denmark

Abstract

Consider a sender $\mathcal{S}$ and a group of n recipients. $\mathcal{S}$ holds a secret message m of length l bits and the goal is to allow $\mathcal{S}$ to create a secret sharing of m with privacy threshold t among the recipients, by broadcasting a single message c to the recipients. Our goal is to do this with information theoretic security in a model with a simple form of correlated randomness. Namely, for each subset $\mathcal{A}$ of recipients of size q, $\mathcal{S}$ may share a random key with all recipients in $\mathcal{A}$. (The keys shared with different subsets $\mathcal{A}$ must be independent.) We call this *Broadcast Secret-Sharing* (BSS) with parameters l, n, t and q.

Our main question is: how large must c be, as a function of the parameters? We show that $\frac{n-t}{q}l$ is a lower bound, and we show an upper bound of $(\frac{n(t+1)}{q+t} - t)l$, matching the lower bound whenever $t = 0$, or when $q = 1$ or $n - t$.

When $q = n - t$, the size of c is exactly l which is clearly minimal. The protocol demonstrating the upper bound in this case requires $\mathcal{S}$ to share a key with *every* subset of size $n - t$. We show that this overhead cannot be avoided when c has minimal size.

We also show that if access is additionally given to an idealized PRG, the lower bound on ciphertext size becomes $\frac{n-t}{q}\lambda + l - negl(\lambda)$ (where λ is the length of the input to the PRG). The upper bound becomes $(\frac{n(t+1)}{q+t} - t)\lambda + l$.

BSS can be applied directly to secret-key threshold encryption. We can also consider a setting where the correlated randomness is generated using computationally secure and non-interactive key exchange, where we assume that each recipient has an (independently generated) public key for this purpose. In this model, any protocol for non-interactive secret sharing becomes an *ad hoc threshold encryption (ATE) scheme*, which is a threshold encryption scheme with no trusted setup beyond a PKI. Our upper bounds imply new ATE schemes, and our lower bound becomes a lower bound on the ciphertext size in any ATE scheme that uses a key exchange functionality and no other cryptographic primitives.

2012 ACM Subject Classification Security and privacy → Information-theoretic techniques

Keywords and phrases Secret-Sharing, Ad-hoc Threshold Encryption

Digital Object Identifier 10.4230/LIPIcs.ITC.2021.11

Funding *Ivan Bjerre Damgård*: The European Research Council (ERC) under the European Unions's Horizon 2020 research and innovation programme under grant agreement No 669255 (MPCPRO).
Kasper Green Larsen: Independent Research Fund Denmark (DFF) Sapere Aude Research Leader grant No 9064-00068B.
Sophia Yakoubov: The European Research Council (ERC) under the European Unions's Horizon 2020 research and innovation programme under grant agreement No 669255 (MPCPRO).

2nd Conference on Information-Theoretic Cryptography (ITC 2021).
Editor: Stefano Tessaro; Article No. 11; pp. 11:1–11:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

In this paper, we consider the following scenario: We have a sender $\mathcal{S}$ and a group of n recipients. $\mathcal{S}$ holds a secret message m of length l bits, and the goal is to allow $\mathcal{S}$ to create a secret sharing of m with privacy threshold t among the recipients. This should be done by broadcasting a single message c to the recipients, followed by local computation by the recipients.

Our goal is to do this with information theoretic security, and since this is clearly impossible in the plain model, we consider a model with correlated randomness.

Note that, if the correlated randomness is "strong enough", the problem becomes trivial: we could ask that $\mathcal{S}$ has a random secret r of the same length as m and the recipients have shares of r in, e.g., Shamir's secret sharing scheme. Now, $\mathcal{S}$ can broadcast $\mathsf{m} - r$ which is clearly of minimal size, and the recipients adjust their shares accordingly. The problem, however, is that each instance of the correlation can only be used once. And if we want to use the Shamir-based solution several times, the only known approach is to create every correlation instance from scratch, implying a communication cost for every instance. In other words, there is no known way to create new instances from old ones using only local computation, not even if we settle for computational security.

We therefore choose an arguably arguably simpler and easier to implement form of correlated randomness where $\mathcal{S}$ shares random strings with one or more of the recipients. More precisely, for each subset $\mathcal{A}$ of recipients of size q, $\mathcal{S}$ may share a secret random bit string $\mathsf{s}_{\mathcal{A}}$ with all recipients in $\mathcal{A}$. Note that this form of correlated randomness can be set up using only communication between the sender and the receivers; receivers do not need to interact. Furthermore, from one instance of such correlated randomness, the parties can generate as many new (pseudorandom) instances as they like using a PRF and *only local computation*. These properties are very useful for applications. See, for instance, Section 1.1.

For any q, we also allow $\mathcal{S}$ to share a secret with any subset smaller than q [1]. This means that, for larger q, we have stronger forms of correlated randomness.

We consider protocols where $\mathcal{S}$ computes c from m and all the shared secrets ($\mathsf{s}_{\mathcal{A}}$'s). Then c is broadcast, and each recipient computes his share of m from c and the shared secrets he holds. Security means that c and the information held by up to t recipients contain no information on m, but c and the information held by any $t + 1$ recipients determine m.

We call the notion we just sketched *Broadcast Secret-Sharing* (*BSS*), with parameters l, n, t and q. In the following, we will sometimes refer to c as the *ciphertext* and the correlated randomness as *shared keys*, which is motivated by the fact that any broadcast secret sharing scheme can be used as is for a secret key threshold encryption scheme. More on this interpretation below.

Our main question is: how large must c be, as a function of the parameters? And, as a secondary question, how much secret correlated data do we need? To the best of our knowledge, these questions, as well the notion of broadcast secret-sharing, have not been considered before.

Let l_{c} be the length of c. It is easy to see that

$$l \leq l_{\mathsf{c}} \leq n \cdot l.$$

Namely, c must always carry enough information to transmit m to the receivers — and on

[1] The motivation is that, for virtually any way to implement the shared randomness, $\mathcal{S}$ could always share with $q' < q$ parties by imagining $q - q'$ virtual parties and emulate these herself.

the other hand, $\mathcal{S}$ can always solve the problem by sharing a one-time pad key with each receiver, then making a standard secret sharing of m and letting c consist of the one-time pad encryptions of each of the shares.

In this paper, we show the much stronger conditions

$$\frac{n-t}{q} l \leq l_{\mathsf{c}} \leq (\frac{n(t+1)}{q+t} - t) l.$$

Note that our upper bound matches the lower bound whenever $t = 0$ or when $q = 1$ or $n - t$. Note also that when $q = n - t$, the size of c is exactly l which is minimal, so $q = n - t$ is the largest value it makes sense to consider. The protocol demonstrating the upper bound in this case requires $\mathcal{S}$ to share a key with *every* subset of size $n - t$. We show that this (possibly exponential) overhead cannot be avoided when c has minimal size.

The BSS schemes we mentioned so far produce Shamir secret-sharings as output. In the final part of the paper, we show that if access is additionally given to an idealized PRG[2], other solutions become possible. Namely, the sender chooses a PRG-input, shares it among the receivers using the best available BSS, and one-time pad encrypts the message using the output from the PRG. Note that this produces a non-standard, non-linear secret sharing. The lower bound on ciphertext size becomes $\frac{n-t}{q}\lambda + l - negl(\lambda)$ (where λ is the length of the input to the PRG). The upper bound becomes $(\frac{n(t+1)}{q+t} - t)\lambda + l$.

1.1 Applications

We believe broadcast secret-sharing is interesting in its own right, and we describe below a couple of applications that make use of a BSS-scheme "out of the box". As further motivation, we also consider in the following subsection two different ways to provide the correlated randomness, leading to other applications.

1.1.1 (Secret-Key) Threshold Encryption

The first application is to secret-key threshold encryption, where a sender sends a ciphertext to set of receivers such that only large enough subsets can decrypt. The main difference between broadcast secret sharing and secret-key threshold encryption is that, in secret-key threshold encryption, it is important that the shared keys be *reusable*. We can easily achieve this by interpreting each key shared between $\mathcal{S}$ and a (subset of) receiver(s) as a key for a pseudorandom function (PRF) ϕ. To encrypt, $\mathcal{S}$ chooses a random nonce r, and for each shared key K, computes $\phi_K(r)$. Note that these PRF values form a (pseudorandom) set of values that can be used as fresh correlated randomness for the broadcast secret-sharing scheme we use. $\mathcal{S}$ now uses this scheme to share her message m among the receivers, resulting in a ciphertext c, and sends the pair (r, c). Decryption can clearly be done by any subset consisting of at least $t + 1$ receivers, and no smaller subset learns anything, which follows easily from security of the PRF and the underlying BSS-scheme. Note that decryption requires minimal interaction: each receiver just has to send his share to the others.

Note also that this application works exactly for the simple form of correlated randomness we use, where $\mathcal{S}$ knows some keys, and each receiver knows a subset of them. Had we allowed

[2] We warn the reader that in an actual implementation, a real PRG would have to be used, and the scheme would only be computationally secure.

a more complicated correlation, the receivers could not have generated new (pseudorandom) correlations of the same form simply by applying the PRF locally.

1.1.2 Secure Multiparty Computation

A second application of BSS is to use it to non-interactively supply input to a secret-sharing based multiparty computation protocol, where the keys held by the sender and receivers can be generated in an earlier setup phase. Given an ideal functionality for distributing keys, we get information theoretic security if the keys are used once. But if we are happy with computational security, we can use a PRF as explained in the previous subsection to extend the key material and support any number of inputs. Note that this will not work when using the well-known method of "pre-cooking" a Shamir secret sharing of a random value known to the sender. Note also that our construction generates Shamir secret-sharings and so is compatible with standard MPC protocols.

1.2 Implementing Shared Keys

Broadcast secret sharing assumes keys shared between the sender and (subsets of) the receiver(s). To discuss the use of BSS in practice, we must also consider the distribution of these keys. We suggest two approaches: non-interactive key exchange (NIKE), and quantum key agreement.

1.2.1 Using NIKE to get (Public-Key) Ad-Hoc Threshold Encryption

In this subsection, we discuss a way to generate the shared keys on the fly, via computationally secure and non-interactive key exchange. Here, we assume that each recipient has an (independently generated) public key and secret key for this purpose.

In this model, any protocol for BSS (including our upper bounds) implies a (public-key) *ad hoc threshold encryption (ATE) scheme*, which is a threshold encryption scheme with no trusted setup beyond a PKI. Namely, the sender creates a ciphertext that includes the information required for the key exchange as well as the c created for broadcast secret-sharing of the message m. To decrypt, at least $t+1$ recipients will first compute the shared randomness using the key exchange, then use this to compute their shares, and finally exhange the shares to reconstruct m. In the related work section below, we give more background on ATE and its relation to standard threshold encryption.

Note that for $q = 1$ the non-interactive key exchange can be done very efficiently based on the DDH assumption: if each receiver i has a public key of form g^{x_i} in some appropriate group, then $\mathcal{S}$ just needs to include a single element g^r for random r in the ciphertext, then the shared key will be of form $g^{x_i r}$ for receiver i. A similar solution for $q = 2$ can be designed using pairing friendly groups. Thus, for these cases, our upper bounds become (essentially) upper bounds on the ciphertext size of the corresponding ATE-scheme. In particular, the ATE-scheme that follows from this and our construction for $q = 2$ has smaller ciphertext size than the best previous scheme of Daza *et al.* [DHMR08]. For instance, when $t = 1$, that scheme has ciphertext size $(n-1)l$ while we can obtain $(\frac{2n}{3} - 1)l$.

Less efficient non-interactive key exchange solutions also exist for larger values of q. They can be constructed from multilinear maps, indistinguishability obfuscation [BZ14], universal samplers [HJK$^+$16, GPSZ17] (which can be built from indistinguishability obfuscation or functional encryption), or encryption combiners satisfying perfect independence [MZ17] (which can be built from universal samplers).

On the other hand, in this setting, our lower bound becomes a lower bound on the ciphertext size in any ATE scheme that uses an ideal functionality for key exchange (and perhaps for PRG), and no other cryptographic primitives. We formalize the demand that no other cryptographic primitives are used by requiring that the scheme is information theoretically secure when using the ideal functionalities.

We stress that these lower bounds hold for ATE-schemes that have access to the cryptographic primitives only via the ideal functionalities they implement. This is more restrictive than if black-box access were given to the corresponding algorithms; one might say that we allow the protocol to use them “only as intended”. However, to the best our knowledge, no general lower bound was known for ATE before.

1.2.2 Using Quantum Agreement

The correlated randomness needed for BSS can also be provided in a setting where the sender shares entangled quantum states with each of the receivers. As is well known, if sender and receiver share a pair of particles that are in the so-called EPR state, then measuring each particle results in the same random bit being obtained by both parties. Moreover, as long as the state really is the pure EPR state, no third party has any information on the randomness obtained. Thus this setting gives us exactly what we want for $q = 1$, with perfect security assuming perfect ability to prepare states and measure them. The same is true if one assumes that sender and receiver has executed a secure quantum key exchange protocol at some earlier time.

The case of $q > 1$ also has a quantum implementation, namely if we assume that the sender shares multipartite entangled states with subsets of receivers. In a multipartite entangled state, each involved party holds a particle, and the global state of the particles can be designed to be fully entangled so that local measurements return the same random result for all parties.

1.3 Related Work

1.3.1 Threshold Secret-Key Cryptosystems

There is not much work on secret-key (symmetric) cryptosystems where the decryption and/or the encryption process can be distributed among a number of parties. A formal study of this was done by Agrawal et al. [AMMR18], in which formal security definitions and constructions were given for the case where both encryption and decryption is distributed. Our construction is in a different model where only the decryption is distributed. This allows us to offer new tradeoffs for constructions using only secret-key primitives and no public-key techniques, which is usually the more efficient case. The one construction from [AMMR18] using only secret-key primitives (a PRF) is very similar to our solution where $q = n - t$. It has minimal ciphertext size l but requires $\binom{n}{t}$ keys, potentially leading to exponential in n overhead. At the other extreme, we have the trivial solution where $q = 1$ and the sender secret-shares the message and sends a share to each receiver, leading to ciphertext size nl and a total of n keys. However, the construction leading to our upper bound implies a spectrum of options “in between”, namely we can get ciphertext size $(\frac{n(t+1)}{q+t} - t)l$ using $\frac{n}{q+t}\binom{q+t}{t}$ keys.

1.3.2 Threshold Public-Key Cryptogsystems

The concept of public-key threshold encryption is very well known. It goes back at least to Desmedt *et al.* [DDFY94], and has since then been studied in a very long line of research.

For this type of scheme, the key generation outputs a public key pk and a set of secret keys $\mathsf{sk}_1, \ldots, \mathsf{sk}_n$ which are generated with respect to a threshold value t, where $0 \le t < n$. Informally, the important security properties are that given any set of at least $t+1$ secret keys, one can decrypt a ciphertext encrypted under pk, while the encryption remains secure even given any set of t secret keys. For efficiency, ciphertexts should have size independent of n.

Requiring a single trusted execution of key generation can be very limiting, particularly in a system where parties may join at any point, or where senders want to dynamically choose subsets of the parties to be the recipients of a particular message. *Dynamic* threshold public-key encryption, introduced by Delerablée and Pointcheval [DP08], has a reduced setup requirement where the sender can pick the set of n recipients at encryption time; however, each recipient's secret key must be derived from a common master secret key, so a trusted authority is still necessary. *Ad hoc threshold encryption* (ATE), first introduced by Daza *et al.* [DHMR08] as *threshold broadcast encryption*[3] (motivated by its applicability to mobile ad hoc networks), requires no trusted setup beyond the absolute minimum — a PKI.

ATE considers a universe of users, where each user i has a public key pk_i and corresponding secret key sk_i, and where all key pairs are independently generated. A sender can select a set $\mathcal{R}$ of n users and a threshold value t at the time at which he decides to send a message m. He can then construct a ciphertext $\mathsf{c} = E_{\mathsf{pk}_{\mathcal{R}},t}(\mathsf{m})$, where $\mathsf{pk}_{\mathcal{R}}$ is the set of public keys belonging to parties in $\mathcal{R}$. ATE requires properties similar to those of standard threshold encryption: namely, that any $t+1$ parties in $\mathcal{R}$ can decrypt, while the encryption remains semantically secure even given the secret keys of any t parties in $\mathcal{R}$.

Clearly, ATE has a number of attractive properties that standard threshold encryption lacks: no trusted authority, and the ability to decide on the set of receivers and the threshold on the fly. On the other hand, it is not clear that an ATE ciphertext can be as small as a standard one. The best known solution is from Daza *et al.* [DHMR08]. They show how to get ciphertext size linear in $n-t$. This solution is in our model discussed earlier (though it was not presented this way). Namely, it combines a BSS-scheme with non-interactive key exchange, where $q=1$. In fact, their BSS scheme is a special case of our upper bound.

In this context, our lower bound shows that the ATE scheme of Daza *et al.* has optimal ciphertext size in the class of ATE schemes that use non-interactive key exchange with $q=1$ and no other cryptographic tools (but as mentioned above, it can be improved using $q=2$). To the best of our knowledge, our bound is the first lower bound obtained for ATE schemes.

Reyzin *et al.* [RSY18] show that using indistinguishability obfuscation, as well as a few standard primitives, it is possible to get ciphertext size independent of n. There are several reasons, however, why this is not a very satisfactory answer. For one thing, the construction requires that senders (as well as receivers) have public and secret keys, which is not usually assumed for ATE. Moreover, obfuscation requires strong assumptions; and with current state of the art techniques, it comes at the price of a huge loss of efficiency in practice.

1.3.3 Pseudorandom Secret-Sharing

In [CDI05], Cramer *et al.* show that, in a model where sufficiently many independent random values are generated and each player is given an appropriate subset of these, the players can locally convert this information to a random Shamir secret-sharing (with a fixed threshold

[3] One should note that ATE for $t=0$ is very similar to broadcast encryption: each party can decrypt on his own. However, in broadcast encryption, centralized key generation is usually allowed (or at least key generation is coordinated between receivers). This is exactly what is not allowed in ATE.

that depends on the set-up). This model is a somewhat similar to ours. The crucial difference, however, is that we have a distinguished player - the sender - who knows all the values and can send a single message to the others. This allows us to create secret-sharings with any threshold, and while we do make use of their technique in our construction, we need additional new ideas to do so.

1.4 Open Problems

There is a very rich space of problems to explore. The most obvious open question is of course to close the gap between the upper and the lower bound on ciphertext size. Another problem is to understand how large the correlated randomness must be. Can the lower bound for minimal ciphertext size be generalized, or is there a way to get polynomial size randomness when the ciphertext is (close to) minimal size?

2 Definitions

In this section, we give the syntax and security definitions for broadcast secret sharing (BSS).

We consider the following random variables:

- $\mathsf{S}_{\mathcal{A}}$, the random variable shared by the sender with the q parties in the set $\mathcal{A}$,
- the message M, and
- the ciphertext C.

For ease of notation, we also let U be the random variable giving all the secrets $\mathsf{S}_{\mathcal{A}}$ shared by the sender with any subset of receivers, U_i be the random variable giving all the secrets held by party $\mathcal{P}_i$ (that is, $\mathsf{U}_i = \{\mathsf{S}_{\mathcal{A}}\}_{i\in\mathcal{A}}$), and $\mathsf{U}_{\mathcal{A}}$ be the random variable giving the union of all the secrets held by parties in $\mathcal{A}$.

We use uppercase variables — $\mathsf{S}, \mathsf{U}, \mathsf{M}, \mathsf{C}$ — to refer to distributions, and lowercase variables — $\mathsf{s}, \mathsf{u}, \mathsf{m}, \mathsf{c}$ — to refer to concrete values.

2.1 BSS Syntax

We assume that any BSS scheme comes with a specification of finite sets from where the random variables are to be chosen. Hence, when we say in the following "any distribution of M", for instance, this means any distribution over the specified set of outcomes.

A BSS scheme with parameters (l, n, t, q) consists of two algorithms, described below.

$E_{\mathsf{u}_{\mathcal{R}}}(\mathbf{m}) \rightarrow \mathbf{c}$ is a secret sharing algorithm (which we also sometimes dub *encryption*) that uses a set of keys $\mathsf{u}_{\mathcal{R}} = \{\mathsf{u}_i\}_{i\in\mathcal{R}}$ belonging to the parties in the size-n set $\mathcal{R}$ of intended recipients (where each u_i consists of all secrets known to sets $\mathcal{A}$ where $i \in \mathcal{A}$) to transform a length-l message m into a secret sharing (or *ciphertext*) c.

$D_{\mathsf{u}_{\mathcal{A}}}(\mathbf{c}) \rightarrow \mathbf{m}$ is a reconstruction (or *decryption*) algorithm that uses keys $\mathsf{u}_{\mathcal{A}} = \{\mathsf{u}_i\}_{i\in\mathcal{A}}$ belonging to a subset $\mathcal{A}$ of the intended recipient set $\mathcal{R}$ (where $|\mathcal{A}| > t$) to recover the message m from the sharing / ciphertext c.

2.2 BSS Security

Informally, a BSS scheme is secure if any t parties in the designated set of receivers $\mathcal{R}$ can learn nothing about a message from a ciphertext, but any $t + 1$ parties in $\mathcal{R}$ can recover the message. More precisely:

▶ **Definition 1** (BSS Perfect Security). *A BSS scheme* (E, D) *is* perfectly secure with threshold t *if for any set of receivers* $\mathcal{R}$ *of size* n*, for* $\mathsf{C} = E_{\mathsf{U}_\mathcal{R}}(\mathsf{M})$*, the following two properties hold for any distribution of* M*:*

Security *For any* $\mathcal{A} \subset \mathcal{R}$ *of size at most* t*, we have* $H(\mathsf{M}|\mathsf{C}, \mathsf{U}_\mathcal{A}) = H(\mathsf{M})$*.*

Correctness *For any* $\mathcal{A} \subset \mathcal{R}$ *of size greater than* t*, we have* $H(\mathsf{M}|\mathsf{C}, \mathsf{U}_\mathcal{A}) = 0$*. Furthermore,* $\mathsf{M} = D_{\mathsf{U}_\mathcal{R}}(\mathsf{C})$*.*

We can define statistical security similarly, where we assume that the distribution of the variables may also depend on a security parameter λ, but we always assume that the parameters l, n, t are polynomial in λ.

▶ **Definition 2** (BSS Statistical Security). *A BSS scheme* (E, D) *is* statistically secure with threshold t *if for any set of receivers* $\mathcal{R}$ *of size* n*, for* $\mathsf{C} = E_{\mathsf{U}_\mathcal{R}}(\mathsf{M})$*, the following two properties hold for any distribution of msg:*

Security *For any* $\mathcal{A} \subset \mathcal{R}$ *of size at most* t*, we have* $H(\mathsf{M}|\mathsf{C}, \mathsf{U}_\mathcal{A}) \geq H(\mathsf{M}) - negl(\lambda)$*.*

Correctness *For any* $\mathcal{A} \subset \mathcal{R}$ *of size greater than* t*, we have* $H(\mathsf{M}|\mathsf{C}, \mathsf{U}_\mathcal{A}) \leq negl(\lambda)$*. Furthermore,* $\mathsf{M} = D_{\mathsf{U}_\mathcal{R}}(\mathsf{C})$ *with overwhelming probability.*

Finally we define a different type of security that we will need later for technical reasons. It is designed for a situation where $t = 0$, so C alone reveals nothing about the message. Moreover, each player on her own can learn l' bits of the message, but not necessarily the entire message.

▶ **Definition 3** (BSS l'-Security). *A BSS scheme* (E, D) *is* $l'-$secure *if for any set of receivers* $\mathcal{R}$ *of size* n*, for* $\mathsf{C} = E_{\mathsf{U}_\mathcal{R}}(\mathsf{M})$*, the following two properties hold for any distribution of* M *and some* $l' \leq H(\mathsf{M})$*:*

Security $H(\mathsf{M}|\mathsf{C}) \geq H(\mathsf{M}) - negl(\lambda)$*.*

Correctness *For any receiver* $\mathcal{P}_i$ *we have* $H(\mathsf{M}|\mathsf{C}, \mathsf{U}_i) \leq H(M) - l' + negl(\lambda)$*.*

Clearly, if a BSS-scheme is l'-secure for $l' = H(M)$, it is statistically secure in the case where $t = 0$.

3 Lower Bounds for Broadcast Secret Sharing

In this section, we prove a lower bound for BSS schemes with statistical security. Throughout the proofs, we consider sending a uniform random message M of l bits. We then prove that the corresponding ciphertext of a BSS scheme must (roughly) satisfy $H(\mathsf{C}) \geq nH(\mathsf{M})/q = nl/q$. Since the entropy of a random variable giving a bit string is a lower bound on its expected length (Shannon's source coding theorem), this also lower bounds the length of the ciphertext. We prove the lower bound in steps, starting with the warm-up case $t = 0, q = 1$ and then extending it to arbitrary q and finally also to arbitrary t.

3.1 Warm-Up: BSS with $t = 0$ and $q = 1$

We start with a lower bound proof in the simple setup with threshold $t = 0$ and shared keys among $q = 1$ recipients. We let the message M be a uniform random bit string of length l (hence $H(\mathsf{M}) = l$). We prove the following lower bound, where $negl(\lambda)$ may be replaced by 0 for perfect security:

▶ **Theorem 4.** *For any BSS scheme with statistical security,* n *recipients, threshold* $t = 0$ *and sharing of keys with* $q = 1$ *recipients, we must have:*

$$H(\mathsf{C}) \geq n(l - negl(\lambda)).$$

To prove the lower bound, let S_i for $i = 1, \ldots, n$ denote the shared key received by the i'th recipient (for $q = 1$, only i receives that random key). The high level idea in our proof is to argue that C must contain a lot of information about the randomness S_i for every index i. Since the shared keys are independent, this implies a lower bound on the entropy of C. More formally, consider the mutual information $I(\mathsf{S}_i; \mathsf{C} \mid \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_{i-1})$. We will show:

▶ **Lemma 5.** *For all recipients i, it holds that $I(\mathsf{C}; \mathsf{S}_i \mid \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_{i-1}) \geq l - negl(\lambda)$.*

Before proving Lemma 6, let us see how we use it to prove Theorem 5. Using non-negativity of entropy and the chain rule of mutual information, we have

$$\begin{aligned} H(\mathsf{C}) &\geq H(\mathsf{C} \mid \mathsf{M}) \\ &\geq H(\mathsf{C} \mid \mathsf{M}) - H(\mathsf{C} \mid \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_n) \\ &= I(\mathsf{C}; \mathsf{S}_1, \ldots, \mathsf{S}_n \mid \mathsf{M}) \\ &= \sum_{i=1}^{n} I(\mathsf{C}; \mathsf{S}_i \mid \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_{i-1}) \\ &\geq n(l - negl(\lambda)). \end{aligned}$$

This completes the proof of Theorem 5. Thus what remains is to prove Lemma 6.

Proof of Lemma 6. The basic idea in the proof of Lemma 6 is that C and S_i together reveal M, thus collectively they must have $l - negl(\lambda)$ bits of information about M. Since $\mathsf{S}_1, \ldots, \mathsf{S}_i$ alone have no information about M, those $l - negl(\lambda)$ bits must be accounted for in $I(\mathsf{C}; \mathsf{S}_i \mid \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_{i-1})$. We prove that formally in the following. By definition, the mutual information in Lemma 6 equals:

$$\begin{aligned} I(\mathsf{C}; \mathsf{S}_i \mid \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_{i-1}) &= \\ H(\mathsf{S}_i \mid \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_{i-1}) - H(\mathsf{S}_i \mid \mathsf{C}, \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_{i-1}). \end{aligned}$$

The message M and all the shared keys are independent, hence $H(\mathsf{S}_i \mid \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_{i-1}) = H(\mathsf{S}_i)$. Since entropy may only increase by dropping variables we condition on, we also conclude $H(\mathsf{S}_i \mid \mathsf{C}, \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_{i-1}) \leq H(\mathsf{S}_i \mid \mathsf{C}, \mathsf{M})$. Using the definition of mutual information, we thus have:

$$\begin{aligned} I(\mathsf{S}_i; \mathsf{C} \mid \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_{i-1}) &\geq H(\mathsf{S}_i) - H(\mathsf{S}_i \mid \mathsf{C}, \mathsf{M}) \\ &= I(\mathsf{S}_i; \mathsf{C}, \mathsf{M}) \\ &= H(\mathsf{C}, \mathsf{M}) - H(\mathsf{C}, \mathsf{M} \mid \mathsf{S}_i). \end{aligned}$$

Since the ciphertext C contains no information about M alone (up to $negl(\lambda)$), we have $H(\mathsf{C}, \mathsf{M}) = H(\mathsf{C}) + H(\mathsf{M} \mid \mathsf{C}) \geq H(\mathsf{C}) + H(\mathsf{M}) - negl(\lambda)$. By the chain rule of entropy, we have $H(\mathsf{C}, \mathsf{M} \mid \mathsf{S}_i) = H(\mathsf{C} \mid \mathsf{S}_i) + H(\mathsf{M} \mid \mathsf{C}, \mathsf{S}_i) \leq H(\mathsf{C}) + H(\mathsf{M} \mid \mathsf{C}, \mathsf{S}_i)$. But $H(\mathsf{M} \mid \mathsf{C}, \mathsf{S}_i) \leq negl(\lambda)$ since recipient i can recover M from C and S_i. We therefore have:

$$\begin{aligned} I(\mathsf{S}_i; \mathsf{C} \mid \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_{i-1}) &\geq H(\mathsf{C}) + H(\mathsf{M}) - negl(\lambda) - (H(\mathsf{C}) + negl(\lambda)) \\ &= H(\mathsf{M}) - negl(\lambda) \\ &= l - negl(\lambda). \end{aligned}$$

◀

3.2 BSS with $t = 0$

In this section, we generalize the lower bound from Section 3.1 to $q \geq 1$ (still assuming $t = 0$ and that the message M is a uniform random l bit string):

► **Theorem 6.** *For any BSS scheme with statistical security, n recipients, security threshold $t = 0$ and sharing of keys with q recipients, we must have:*

$$H(\mathsf{C}) \geq n(l - negl(\lambda))/q.$$

To show this, we will show a stronger statement that will be useful for other purposes in the following:

► **Theorem 7.** *For any l'-secure BSS scheme with n recipients, and sharing of keys with q recipients, we must have:*

$$H(\mathsf{C}) \geq n(l' - negl(\lambda))/q.$$

Clearly, this result implies Theorem 7: when M is uniform and $H(\mathsf{M}) = l$, the assumption in Theorem 7 is equivalent to requiring l-security.

The basic idea in the proof for $q = 1$ was to argue that the ciphertext C contained a lot of information about each S_i. Formally, Lemma 6 showed that $I(\mathsf{C}; \mathsf{S}_i \mid \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_{i-1}) \geq l - negl(\lambda)$. In the following, we discuss the obstacles we face when generalizing the proof to $q \geq 1$ and show how we overcome them.

First, in order to prove Lemma 6, we used the fact that S_i together with C revealed M to conclude that $I(\mathsf{C}; \mathsf{S}_i \mid \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_{i-1}) \geq l - negl(\lambda)$. Considering instead l'-security this statement would be $I(\mathsf{C}; \mathsf{S}_i \mid \mathsf{M}, \mathsf{S}_1, \ldots, \mathsf{S}_{i-1}) \geq l' - negl(\lambda)$ and it could be proved in exactly the same way for $q = 1$.

However, since a recipient may now use all his shared keys to recover M, we define a random variable U_i for each recipient i: We let U_i denote all shared keys held by recipient i ($\mathsf{U}_i = \{\mathsf{S}_\mathcal{A}\}_{i \in \mathcal{A}}$). Intuitively, the analog of Lemma 6 would state that $I(\mathsf{C}; \mathsf{U}_i \mid \mathsf{M}, \mathsf{U}_1, \ldots, \mathsf{U}_{i-1}) \geq l' - negl(\lambda)$.

With this definition of U_i we again have that U_i and C together reveal l' bits of M. Unfortunately, the sets of shared keys held by different recipients are not disjoint. This means that U_i may depend on $\mathsf{U}_1, \ldots, \mathsf{U}_{i-1}$ and thus the lower bound on the mutual information is not necessarily true.

Our key idea for addressing the above issue is to further partition U_i into subset $\mathsf{U}_{i,1}, \ldots, \mathsf{U}_{i,q}$ where $\mathsf{U}_{i,k}$ contains all shared keys $\mathsf{S}_\mathcal{A}$ for which i is the k'th smallest index in $\mathcal{A}$. Note that with this definition $\mathsf{U}_{i,k}$ and $\mathsf{U}_{j,k}$ with $i \neq j$ are disjoint sets of shared keys (only one index can be the k'th smallest in a set $\mathcal{A}$) and thus are independent. The same holds for $\mathsf{U}_{i,j}$ and $\mathsf{U}_{i,k}$ with $j \neq k$ (i cannot both be the j'th and k'th smallest index in $\mathcal{A}$). Finally, we also define $F_{i,k}$ to denote the set of all shared keys $\mathsf{S}_\mathcal{A}$ in which i is the largest index in $\mathcal{A}$ and $|\mathcal{A}| < k$. Our generalization of Lemma 6 then becomes:

► **Lemma 8.** *There is an index $k \in \{1, \ldots, q\}$ such that*

$$\sum_{i=1}^{n} I(\mathsf{U}_{i,k} F_{i,k}; \mathsf{C} \mid \mathsf{M}, \mathsf{U}_{i+1,k}, F_{i+1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}) \geq n(l' - negl(\lambda))/q.$$

Before proving Lemma 9, let us see that it implies Theorem 7. We have:

$$\begin{aligned} H(\mathsf{C}) &\geq H(\mathsf{C} \mid \mathsf{M}) \\ &\geq H(\mathsf{C} \mid \mathsf{M}) - H(\mathsf{C} \mid \mathsf{M}, \mathsf{U}_{1,k}, F_{1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}) \\ &= I(\mathsf{C}; \mathsf{U}_{1,k}, F_{1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k} \mid \mathsf{M}) \\ &= \sum_{i=1}^{n} I(\mathsf{C}; \mathsf{U}_{i,k}, F_{i,k} \mid \mathsf{M}, \mathsf{U}_{i+1,k}, F_{i+1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}) \\ &\geq n(l' - negl(\lambda))/q. \end{aligned}$$

What remains is thus to prove Lemma 9. The key step in doing so is to replace each mutual information in the sum by a term that only depends on the sets $\mathsf{U}_{i,1}, \ldots, \mathsf{U}_{i,q}$ seen by the i'th recipient. The rewriting is quite non-trivial and crucially relies on the fact that we applied the chain rule in reverse order of indices such that we condition on $\mathsf{U}_{j,k}, F_{j,k}$ for indices $j > i$. The rewriting we make uses the following:

▶ **Lemma 9.** *For every recipient i and every index $k \in \{1, \ldots, q\}$ we have*

$$I(\mathsf{U}_{i,k} F_{i,k}; \mathsf{C} \mid \mathsf{M}, \mathsf{U}_{i+1,k}, F_{i+1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}) \geq I(\mathsf{U}_{i,k}; \mathsf{C} \mid \mathsf{M}, \mathsf{U}_{i,1}, \ldots, \mathsf{U}_{i,k-1}).$$

Let us first use Lemma 10 to prove Lemma 9.

Proof of Lemma 9. Consider summing over all recipients and all choices of k, applying Lemma 10 on each term:

$$\begin{aligned} \sum_{k=1}^{q} \sum_{i=1}^{n} I(\mathsf{U}_{i,k} F_{i,k}; \mathsf{C} \mid \mathsf{M}, \mathsf{U}_{i+1,k}, F_{i+1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}) &\geq \\ \sum_{k=1}^{q} \sum_{i=1}^{n} I(\mathsf{U}_{i,k}; \mathsf{C} \mid \mathsf{M}, \mathsf{U}_{i,1}, \ldots, \mathsf{U}_{i,k-1}) &= \\ \sum_{i=1}^{n} \sum_{k=1}^{q} I(\mathsf{U}_{i,k}; \mathsf{C} \mid \mathsf{M}, \mathsf{U}_{i,1}, \ldots, \mathsf{U}_{i,k-1}) &= \\ \sum_{i=1}^{n} I(\mathsf{U}_{i,1}, \ldots, \mathsf{U}_{i,q}; \mathsf{C} \mid \mathsf{M}) &= \\ \sum_{i=1}^{n} I(\mathsf{U}_i; \mathsf{C} \mid \mathsf{M}). \end{aligned}$$

Since U_i and M are independent, we have $I(\mathsf{U}_i; \mathsf{C} \mid \mathsf{M}) = H(\mathsf{U}_i \mid \mathsf{M}) - H(\mathsf{U}_i \mid \mathsf{C}, \mathsf{M}) = H(\mathsf{U}_i) - H(\mathsf{U}_i \mid \mathsf{C}, \mathsf{M}) = I(\mathsf{U}_i; \mathsf{C}, \mathsf{M}) = H(\mathsf{C}, \mathsf{M}) - H(\mathsf{C}, \mathsf{M} \mid \mathsf{U}_i)$. Since M cannot be recovered from C, we have

$$H(\mathsf{C}, \mathsf{M}) = H(\mathsf{C}) + H(\mathsf{M} \mid \mathsf{C}) \geq H(\mathsf{C}) + H(\mathsf{M}) - negl(\lambda).$$

By the chain rule, $H(\mathsf{C}, \mathsf{M} \mid \mathsf{U}_i) = H(\mathsf{C} \mid \mathsf{U}_i) + H(\mathsf{M} \mid \mathsf{C}, \mathsf{U}_i) \leq H(\mathsf{C}) + H(\mathsf{M} \mid \mathsf{C}, \mathsf{U}_i)$. But, by l'-security, l' bits of M are determined from C and U_i, more precisely

$$H(\mathsf{M} \mid \mathsf{C}, \mathsf{U}_i) \leq H(M) - l' + negl(\lambda).$$

We have thus shown $I(\mathsf{U}_i; \mathsf{C} \mid \mathsf{M}) \geq H(\mathsf{C}) + H(\mathsf{M}) - negl(\lambda) - (H(\mathsf{C}) + H(\mathsf{M}) - l' + negl(\lambda)) = l' - negl(\lambda)$. We therefore have:

$$\begin{aligned} \sum_{k=1}^{q}\sum_{i=1}^{n} I(\mathsf{U}_{i,k}F_{i,k}; \mathsf{C} \mid \mathsf{M}, \mathsf{U}_{i+1,k}, F_{i+1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}) &\geq \\ \sum_{i=1}^{n} l' - negl(\lambda) &= \\ n(l' - negl(\lambda)). \end{aligned}$$

Averaging over all choices of k completes the proof of Lemma 9. ◀

To finish, we thus need to prove Lemma 10:

Proof of Lemma 10. We need to show that for all recipients i and every index k, it holds that

$$I(\mathsf{U}_{i,k}F_{i,k}; \mathsf{C} \mid \mathsf{M}, \mathsf{U}_{i+1,k}, F_{i+1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}) \geq I(\mathsf{U}_{i,k}; \mathsf{C} \mid \mathsf{M}, \mathsf{U}_{i,1}, \ldots, \mathsf{U}_{i,k-1}).$$

The main observation needed in the proof is the fact every shared key in $\mathsf{U}_{i,1}, \ldots, \mathsf{U}_{i,k}$ also appears in $\mathsf{U}_{i,k}, F_{i,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}$. More formally, we start by observing that:

$$\begin{aligned} I(\mathsf{U}_{i,k}F_{i,k}; \mathsf{C} \mid \mathsf{M}, \mathsf{U}_{i+1,k}, F_{i+1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}) &\geq \\ I(\mathsf{U}_{i,k}; \mathsf{C} \mid \mathsf{M}, F_{i,k}, \mathsf{U}_{i+1,k}, F_{i+1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}) &= \\ H(\mathsf{U}_{i,k} \mid \mathsf{M}, F_{i,k}, \mathsf{U}_{i+1,k}, \ldots, F_{n,k}) - H(\mathsf{U}_{i,k} \mid \mathsf{C}, \mathsf{M}, F_{i,k}, \mathsf{U}_{i+1,k}, \ldots, F_{n,k}). \end{aligned}$$

Notice that the set of shared keys $\mathsf{U}_{i,k}$ is disjoint from the sets $\mathsf{U}_{j,k}$ with $j \neq i$. This holds since for any set of receivers $\mathcal{A}$, only one receiver can be the k'th smallest. Moreover, $\mathsf{U}_{i,k}$ is also disjoint from $F_{j,k}$ for all j. This is true since $F_{j,k}$ contains only shared keys for sets of receivers with cardinality less than k. This means that $\mathsf{U}_{i,k}$ is independent of $\mathsf{M}, F_{i,k}, \mathsf{U}_{i+1,k}, F_{i+1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}$ and thus we have

$$H(\mathsf{U}_{i,k} \mid \mathsf{M}, F_{i,k}, \mathsf{U}_{i+1,k}, \ldots, F_{n,k}) = H(\mathsf{U}_{i,k}).$$

We therefore have:

$$\begin{aligned} I(\mathsf{U}_{i,k}F_{i,k}; \mathsf{C} \mid \mathsf{M}, \mathsf{U}_{i+1,k}, F_{i+1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}) &\geq \\ H(\mathsf{U}_{i,k}) - H(\mathsf{U}_{i,k} \mid \mathsf{C}, \mathsf{M}, F_{i,k}, \mathsf{U}_{i+1,k}, F_{i+1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}). \end{aligned}$$

Since entropy may only increase by removing variables that we condition on, we remove all shared keys from $F_{i,k}, \mathsf{U}_{i+1,k}, F_{i+1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}$ which do not appear in $\mathsf{U}_{i,1}, \ldots, \mathsf{U}_{i,k-1}$. We claim that we are left with precisely the full set of shared keys appearing in $\mathsf{U}_{i,1}, \ldots, \mathsf{U}_{i,k-1}$. To see this, consider a shared key $\mathsf{S}_\mathcal{A}$ appearing in $\mathsf{U}_{i,j}$ for some $j < k$. Assume first that i is the largest index in the set $\mathcal{A}$. Then the cardinality of $\mathcal{A}$ is $j < k$ and we have $\mathsf{S}_\mathcal{A} \in F_{i,k}$ by definition of $F_{i,k}$. Next, assume that the cardinality of $\mathcal{A}$ is less than k, but i is not the largest index in $\mathcal{A}$. Let $i' > i$ be the largest index. Then by definition, we have $\mathsf{S}_\mathcal{A} \in F_{i',k}$. Finally, assume that the cardinality of $\mathcal{A}$ is at least k. Let $i' > i$ be the k'th smallest index in $\mathcal{A}$, then $\mathsf{S}_\mathcal{A} \in \mathsf{U}_{i',k}$. In all cases, we have that $\mathsf{S}_\mathcal{A}$ is in one of $F_{i,k}, \mathsf{U}_{i+1,k}, F_{i+1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}$ and we conclude that we are left with $\mathsf{U}_{i,1}, \ldots, \mathsf{U}_{i,k-1}$. We therefore have:

$$\begin{aligned} I(\mathsf{U}_{i,k}F_{i,k}; \mathsf{C} \mid \mathsf{M}, \mathsf{U}_{i+1,k}, F_{i+1,k}, \ldots, \mathsf{U}_{n,k}, F_{n,k}) &\geq \\ H(\mathsf{U}_{i,k}) - H(\mathsf{U}_{i,k} \mid \mathsf{C}, \mathsf{M}, \mathsf{U}_{i,1}, \ldots, \mathsf{U}_{i,k-1}). \end{aligned}$$

Conditioning on a random variable may only decrease entropy, we can therefore bound the above by:

$$I(\mathsf{U}_{i,k}F_{i,k};\mathsf{C} \mid \mathsf{M},\mathsf{U}_{i+1,k},F_{i+1,k},\ldots,\mathsf{U}_{n,k},F_{n,k}) \geq$$
$$H(\mathsf{U}_{i,k} \mid \mathsf{M},\mathsf{U}_{i,1},\ldots,\mathsf{U}_{i,k-1}) - H(\mathsf{U}_{i,k} \mid \mathsf{C},\mathsf{M},\mathsf{U}_{i,1},\ldots,\mathsf{U}_{i,k-1}) =$$
$$I(\mathsf{U}_{i,k};\mathsf{C} \mid \mathsf{M},\mathsf{U}_{i,1},\ldots,\mathsf{U}_{i,k-1}).$$

This concludes the proof of Lemma 10 and thus also of Theorem 8. ◀

3.3 Final BSS Lower Bound

In this section, we finally extend the lower bound in Theorem 7 to the general case of $t \geq 0$ and $q \geq 1$. Our final result is the following:

▶ **Theorem 10.** *For any BSS scheme with statistical security, n recipients, security threshold t and sharing of keys with q recipients, we must have:*

$$H(\mathsf{C}) \geq (n-t)(l - negl(\lambda))/q.$$

The proof follows via a reduction from the case with $t = 0$ (Theorem 7). The basic idea is to show that any BSS scheme for arbitrary threshold $t \geq 0$ can be converted into a scheme for $t = 0$ and $n - t$ receivers. This is done by treating the first t receivers as dummy receivers for which all shared keys are public information. This way, we get a BSS scheme with $t = 0$ for the remaining receivers $t + 1, \ldots, n$.

In detail, consider all shared keys $\mathsf{U}_1, \ldots, \mathsf{U}_t$ held by the first t parties in a BSS scheme with threshold t. Consider any concrete instantiation $\mathsf{u}_1, \ldots, \mathsf{u}_t$ of the random variables and let $E_{\mathsf{u}_1,\ldots,\mathsf{u}_t}$ denote the event that $\mathsf{U}_i = \mathsf{u}_i$ for $i = 1, \ldots, t$. We will prove that for most instantiations of $\mathsf{U}_1 = \mathsf{u}_1, \ldots, \mathsf{U}_t = \mathsf{u}_t$, conditioned on $E_{\mathsf{u}_1,\ldots,\mathsf{u}_t}$, the BSS statistical security definitions hold for the remaining $n - t$ receivers with threshold $t = 0$. Formally, we require that:

Security We have $H(\mathsf{M} \mid \mathsf{C}, E_{\mathsf{u}_1,\ldots,\mathsf{u}_t}) \geq H(\mathsf{M}) - negl(\lambda)$.

Correctness For any receiver i with $i \in \{t + 1, \ldots, n\}$, we have

$$H(\mathsf{M} \mid \mathsf{C}, U_i, E_{\mathsf{u}_1,\ldots,\mathsf{u}_t}) \leq negl(\lambda).$$

Call $\mathsf{u}_1, \ldots, \mathsf{u}_t$ *typical* if they satisfies the above Security and Correctness. If $\mathsf{u}_1, \ldots, \mathsf{u}_t$ are typical, then we have a BSS scheme with threshold $t = 0$ for the remaining $n - t$ receivers $t+1, \ldots, n$ if we hard code $\mathsf{U}_1 = \mathsf{u}_1, \ldots, \mathsf{U}_t = \mathsf{u}_t$ and let those be shared knowledge. Therefore, by Theorem 7, it must be the case for typical $\mathsf{u}_1, \ldots, \mathsf{u}_t$, that

$$H(\mathsf{C} \mid E_{\mathsf{u}_1,\ldots,\mathsf{u}_t}) \geq \frac{n-t}{q}(1 - negl(\lambda)).$$

We will show:

▶ **Lemma 11.** $\mathsf{U}_1, \ldots, \mathsf{U}_t$ *are typical with probability at least* $1 - negl(\lambda)$.

Before we prove Lemma 12, we use the lemma to finish the proof of Theorem 11. We see that

$$\begin{aligned} H(\mathsf{C}) &\geq H(\mathsf{C} \mid \mathsf{U}_1, \ldots, \mathsf{U}_t) \\ &= \sum_{\mathsf{u}_1,\ldots,\mathsf{u}_t} H(\mathsf{C} \mid E_{\mathsf{u}_1,\ldots,\mathsf{u}_t}) \Pr[E_{\mathsf{u}_1,\ldots,\mathsf{u}_t}] \\ &\geq \sum_{\mathsf{u}_1,\ldots,\mathsf{u}_t : \mathsf{u}_1,\ldots,\mathsf{u}_t \text{ are typical}} H(\mathsf{C} \mid E_{\mathsf{u}_1,\ldots,\mathsf{u}_t}) \Pr[E_{\mathsf{u}_1,\ldots,\mathsf{u}_t}] \\ &\geq \frac{n-t}{q}(1 - negl(\lambda)) \Pr[\mathsf{U}_1, \ldots, \mathsf{U}_t \text{ are typical}] \\ &= \frac{n-t}{q}(1 - negl(\lambda)). \end{aligned}$$

What remains is thus to prove Lemma 12.

Proof of Lemma 12. Let $X(\mathsf{u}_1, \ldots, \mathsf{u}_t)$ take the value $H(\mathsf{M}) - H(\mathsf{M} \mid \mathsf{C}, E_{\mathsf{u}_1,\ldots,\mathsf{u}_t})$. Observe that since M is independent of $\mathsf{U}_1, \ldots, \mathsf{U}_t$, we have $H(\mathsf{M}) = H(\mathsf{M} \mid E_{\mathsf{u}_1,\ldots,\mathsf{u}_t})$ and thus $X(\mathsf{u}_1, \ldots, \mathsf{u}_t) = H(\mathsf{M} \mid E_{\mathsf{u}_1,\ldots,\mathsf{u}_t}) - H(\mathsf{M} \mid \mathsf{C}, E_{\mathsf{u}_1,\ldots,\mathsf{u}_t})$. Conditioning on C may only decrease entropy, hence X is non-negative for all $\mathsf{u}_1, \ldots, \mathsf{u}_t$. It follows by Markov's inequality that

$$\Pr\left[X(\mathsf{U}_1, \ldots, \mathsf{U}_t) > \sqrt{\mathbb{E}[X(\mathsf{U}_1, \ldots, \mathsf{U}_t)]}\right] < \sqrt{\mathbb{E}[X(\mathsf{U}_1, \ldots, \mathsf{U}_t)]}.$$

Now recall from the security requirements of a BSS scheme with threshold t that:

$$\begin{aligned} H(\mathsf{M}) - negl(\lambda) &\leq H(\mathsf{M} \mid \mathsf{C}, \mathsf{U}_1, \ldots, \mathsf{U}_t) \\ &= \sum_{\mathsf{u}_1,\ldots,\mathsf{u}_t} H(\mathsf{M} \mid \mathsf{C}, E_{\mathsf{u}_1,\ldots,\mathsf{u}_t}) \Pr[E_{\mathsf{u}_1,\ldots,\mathsf{u}_t}], \end{aligned}$$

which implies

$$\begin{aligned} \mathbb{E}[X(\mathsf{U}_1, \ldots, \mathsf{U}_t)] &= H(\mathsf{M}) - \sum_{\mathsf{u}_1,\ldots,\mathsf{u}_t} H(\mathsf{M} \mid \mathsf{C}, E_{\mathsf{u}_1,\ldots,\mathsf{u}_t}) \Pr[E_{\mathsf{u}_1,\ldots,\mathsf{u}_t}] \\ &\leq negl(\lambda). \end{aligned}$$

Thus by Markov's, we have $\Pr\left[X(\mathsf{U}_1, \ldots, \mathsf{U}_t) > negl(\lambda)\right] < negl(\lambda)$.

Next, for any receiver $i > t$, define $Y_i(\mathsf{u}_1, \ldots, \mathsf{u}_t)$ to take the value $H(\mathsf{M} \mid \mathsf{C}, \mathsf{U}_i, E_{\mathsf{u}_1,\ldots,\mathsf{u}_t})$. Since entropy is always non-negative, so is Y_i. By definition of conditional entropy, we have $\mathbb{E}[Y_i(\mathsf{U}_1, \ldots, \mathsf{U}_t)] = H(\mathsf{M} \mid \mathsf{C}, \mathsf{U}_i, \mathsf{U}_1, \ldots, \mathsf{U}_t)$. Thus from Markov's we again have $\Pr[Y_i(\mathsf{U}_1, \ldots, \mathsf{U}_t) > negl(\lambda)] < negl(\lambda)$. It finally follows by a union bound that with probability at least $1-(n-t+1)negl(\lambda) = 1-negl(\lambda)$, we simultaneously have $X(\mathsf{U}_1, \ldots, \mathsf{U}_t) < negl(\lambda)$ and $Y_i(\mathsf{U}_1, \ldots, \mathsf{U}_t) < negl(\lambda)$ for all $i = t+1, \ldots, n$. That is, $\mathsf{U}_1, \ldots, \mathsf{U}_t$ are typical with probability at least $1 - negl(\lambda)$. ◀

4 Upper Bound on Ciphertext Size

In this section, we explore constructions of broadcast secret-sharing.

4.1 Building Block: Pseudorandom Secret Sharing

Our results in this section leverage *pseudorandom secret sharing*, which is a technique for the local (that is, non-interactive) conversion of a replicated secret sharing to a Shamir secret sharing.

A *replicated secret sharing* for the $(t+1)$-out-of-n threshold access structure proceeds as follows. First, the dealer splits the secret M into $\binom{n}{t}$ additive secret shares, where each share $\mathsf{r}_{\mathcal{A}}$ corresponds to a different maximally unqualified set $\mathcal{A}$ of size t. Then, the complement of each set $\mathcal{A}$ (that is, the $n-t$ parties that are *not* in $\mathcal{A}$) are all given $\mathsf{r}_{\mathcal{A}}$. It is then clear that any maximally unqualified set $\mathcal{A}$ is only missing knowledge of one share $\mathsf{r}_{\mathcal{A}}$, which any additional party holds.

Pseudorandom secret sharing [CDI05] locally converts such a replicated secret sharing into a Shamir secret sharing (a degree-t polynomial f with $f(0) = \mathsf{M}$ as the secret, and $f(i) = \mathsf{s}_i$ as party i's share for $i \in [1, \ldots, n]$). Pseudorandom secret sharing proceeds as follows: let $f_{\mathcal{A}}$ be the degree-t polynomial such that $f_{\mathcal{A}}(0) = 1$, and $f_{\mathcal{A}}(i) = 0$ for all $i \in \mathcal{A}$. Each player $\mathcal{P}_i$ can then compute their Shamir share as

$$\mathsf{s}_i = \sum_{\mathcal{A} \subseteq [n] : |\mathcal{A}| = t, i \notin \mathcal{A}} \mathsf{r}_{\mathcal{A}} f_{\mathcal{A}}(i).$$

We stress that, despite the name, pseudorandom secret-sharing as presented here provides perfect information theoretic security. The name comes from an application of the technique that uses pseudorandom functions.

Cramer, Damgård and Ishai [CDI05] also prove a lower bound, stated in Theorem 13.

▶ **Theorem 12** (From [CDI05]). *Fewer than $\binom{n}{t}$ independent random values shared among various subsets of parties cannot be locally converted into a $(t+1)$-out-of-n threshold secret sharing.*

4.2 Lower Bounding the Correlated Randomness When $H(\mathsf{C}) = H(\mathsf{M})$

▶ **Theorem 13.** *For any perfectly secure BSS scheme with threshold $t = \theta(n)$, if $H(\mathsf{C}) = H(\mathsf{M})$, then correlated randomness of exponential size is necessary.*

Proof. If $H(\mathsf{C}) = H(\mathsf{M})$, then for any distribution of keys, there is exactly one ciphertext that corresponds to any given message. Therefore, choosing a ciphertext at random (without considering the correlated randomness) will always give a valid ciphertext that corresponds to some message, no matter which value the randomness takes. Choosing the randomness and ciphertext simultaneously independently at random thus produces a random $(t+1)$-out-of-n secret sharing (where the ciphertext is simply an additional random value given to all parties). So, the exponential lower bound by Cramer *et al.* [CDI05] (Theorem 13) on amount of independent randomness that can be converted into a $(t+1)$-out-of-n secret sharing applies. ◀

4.3 The Upper Bound

Construction 1 below achieves optimal ciphertext size whenever $t = 0$, or when $q = 0$ or when q is the maximal relevant value $n - t$. This construction leverages the techniques of replicated or pseudorandom secret sharing. The price we pays is that the overhead in terms of size of correlated randomness is sometimes exponential (that is, the sender and each of the receivers must use an exponential number of shared random values). Whether this happens depends on the parameter values.

▶ **Construction 1.** *Let $n' = q+t$. We partition the recipients into $\frac{n}{n'}$ subsets of size $n' = q+t$. (We assume for simplicity that $n' = q+t$ divides n.) An arbitrary but fixed one of these subsets is chosen and named $\mathcal{B}$. This is done publicly once and for all. We also assign once and for all a unique point in a suitable finite field to each recipient.*

Consider now any of the above subsets $\mathcal{A}$. We set up the correlated randomness such that the sender $\mathcal{S}$ shares a random value with any subset of $\mathcal{A}$, of size $n' - t = q$. These values form a random replicated secret-sharing among the players in $\mathcal{A}$ and hence, using the technique from [CDI05], $\mathcal{S}$ can share a random polynomial $f_{\mathcal{A}}$ of degree at most t with the participants in $\mathcal{A}$, using only the correlated randomness. Concretely, $\mathcal{S}$ knows $f_{\mathcal{A}}$ and each player in $\mathcal{A}$ knows a point on $f_{\mathcal{A}}$.

The ciphertext consists of $\mathsf{m} + f_{\mathcal{B}}(0)$ and $f_{\mathcal{B}} - f_{\mathcal{A}}$ for every subset $\mathcal{A} \neq \mathcal{B}$.

Each recipient locally computes from the correlated randomness $f_{\mathcal{A}}(i)$ where $\mathcal{A}$ is the subset she is in and i is her assigned point in the field. Then she computes $f_{\mathcal{B}}(i) = f_{\mathcal{A}}(i) + (f_{\mathcal{B}} - f_{\mathcal{A}})(i)$. To reconstruct, any subset of size at least $t+1$ can interpolate $f_{\mathcal{B}}$ and compute $\mathsf{m} = (\mathsf{m} + f_{\mathcal{B}}(0)) - f_{\mathcal{B}}(0)$.

The security of this construction follows trivially from the security of replicated secret sharing: each $f_{\mathcal{A}}$ is uniformly random of degree at most t and so $f_{\mathcal{B}} - f_{\mathcal{A}}$ contains no information on m, even given $\mathsf{m} + f_{\mathcal{B}}(0)$. Since each polynomial $f_{\mathcal{B}} - f_{\mathcal{A}}$ can be specified using $t+1$ coefficients, the ciphetext size is

$$((t+1)(n/(q+t)-1)+1)l = (n(t+1)/(q+t)-t)l.$$

The size of the shared keys (correlated randomness) is $n/(q+t) \cdot \binom{(q+t)}{t}$ field elements. This can be as much as $\binom{n}{t}$ and so may be exponential in n. But as we showed above, at least when $q = n - t$, this overhead cannot be avoided.

5 Bounds Additionally Assuming an Idealized PRG

In this section, we add to our BSS model an idealized pseudorandom generator (PRG); an idealized functionality that takes in a random length-λ seed, and outputs a longer random value. (As long as the output is at least one bit longer than the input, we can bootstrap the PRG to give arbitrarily long outputs. In our case, the output length that most often makes sense is l, the length of the message.) Our BSS algorithms are augmented with oracle access to the idealized PRG.

We make some assumptions on how the BSS protocol may use the idealized PRG:

▶ **Definition 14.** *An* admissible *BSS-protocol satisfies the following:*

- *For any subset of receivers, any PRG-seed chosen by the sender can either be computed using what that subset of receivers knows, or has full entropy (possibly up to a negligible loss).*
- *During the sharing phase, the sender chooses all seeds that are input to PRG uniformly, independently of anything else.*
- *The idealized PRG is not called with any shared keys as input.*

In the following we will only consider admissible BSS constructions. The motivation for this is as follows:

- We want to make sure that an admissible protocol can be turned into a construction in the real world by replacing the idealized PRG by a real PRG construction. Now, if a seed has (essentially) full entropy in the view of the adversary, then (and only then) can we use the standard security of a real PRG to conclude that the output is pseudorandom. Seeds for which the adversary has partial information are not useful in this sense, and we may as well give the adversary full information on that seed for free.
  This is why we assume that in the view of a subset of receivers, any seed that the sender chose can either be computed or has (essentially) full entropy. However, for a seed to be

potentially useful it must have full entropy in the first place, which is why we assume that the sender chooses all seeds uniformly, independently of anything else.
- We assume that the idealized PRG is not called using shared keys as input for simplicity, because this does not cost us any generality: calls to the PRG using shared keys as input is equivalent to asking for longer shared keys. In both cases, the result is a greater amount of correlated randomness.

Finally, we will assume that privacy only needs to hold given ability to call the PRG a polynomial number of times. The reason for this is that otherwise protocols that actually make use of the PRG could not ensure that the message is hidden from a non-qualified subset of receivers. As an example, suppose the sender secret-shares a seed s and includes in the ciphertext a one-time pad encryption $\mathsf{m} \oplus PRG(s)$. A completely unbounded adversary can call the PRG on all inputs and, once all the outputs are given, the only uncertainty she has is which seed the sender used. Then, if m is longer than s, it cannot have full entropy.

To be able to talk about the information a set of receivers can get from the oracle, we abuse notation and let $PRG(\mathsf{C}, \mathsf{U}_\mathcal{A})$ denote the random variable that is obtained by calling the PRG on inputs that are selected by an unbounded randomized algorithm that gets $\mathsf{C}, \mathsf{U}_\mathcal{A}$ as input. The algorithm only returns a polynomial number of outputs. For simplicity of notation, we suppress the algorithm and the random coins it uses.

▶ **Definition 15** (BSS Statistical Security with PRG). *A BSS scheme (E, D) is* statistically secure with threshold t *with respect to a random oracle PRG if for any set of receivers $\mathcal{R}$ of size n, for $\mathsf{C} = E^{PRG}_{\mathsf{U}_\mathcal{R}}(\mathsf{M})$, the following two properties hold for any distribution of M:*

Security *For any $\mathcal{A} \subset \mathcal{R}$ of size at most t, we have $H(\mathsf{M}|\mathsf{C}, \mathsf{U}_\mathcal{A}, PRG(\mathsf{C}, \mathsf{U}_\mathcal{A})) \geq H(\mathsf{M}) - negl(\lambda)$.*

Correctness *For any $\mathcal{A} \subset \mathcal{R}$ of size greater than t, $H(\mathsf{M}|\mathsf{C}, \mathsf{U}_\mathcal{A}, PRG(\mathsf{C}, \mathsf{U}_\mathcal{A})) \leq negl(\lambda)$. Furthermore, $\mathsf{M} = D^{PRG}_{\mathsf{U}_\mathcal{R}}(\mathsf{C})$ with overwhelming probability.*

5.1 Lower Bound on Ciphertext Size

▶ **Theorem 16.** *Consider any BSS scheme that is statistically secure with threshold t with respect to PRG (which takes inputs of size λ) and shares messages of length $l \geq \lambda$. If the scheme is admissible it holds that*

$$H(\mathsf{C}) \geq \frac{n-t}{q}\lambda + l - \delta(\lambda)$$

for a negligible function $\delta(\lambda)$.

To show the above theorem, consider first a scheme that satisfies the assumption with threshold $t = 0$, so then the only unqualified set of receivers is the empty set. Since the scheme is admissible, there is a (possibly empty) set of seeds $\mathcal{S}$ that were chosen by the sender, but where each seed in $\mathcal{S}$ has full entropy given the ciphertext C, and all other seeds are determined by C.

We claim that we can transform this scheme into a new one (for a different distribution of messages) that is l'-secure (Definition 4) with $l' = \lambda$. In particular, this will be a scheme where the PRG is not available. Recall that in such a scheme a qualified subset of receivers can determine at least l' bits of the message.

To this end, we define the message M' in the new scheme to be the original M concatenated with the seeds in $\mathcal{S}$. Reconstruction in the new scheme by a qualified set $\mathcal{A}$ works as follows: If at least one seed $s \in \mathcal{S}$ is determined by $\mathsf{C}, \mathsf{U}_\mathcal{A}$, then return s. Otherwise, by admissibility, all seeds in $\mathcal{S}$ have full entropy given $\mathsf{C}, \mathsf{U}_{\mathcal{A}}$. Consider the random variable $PRG(\mathsf{C}, \mathsf{U}_{\mathcal{A}})$ that would have been used for reconstruction in the original scheme. Notice that since this variable is formed by calling the PRG a polynomial number of times, the inputs used will overlap with $\mathcal{S}$ with only negligible probability. Therefore unless this overlap event happens, access to the PRG can be perfectly simulated without calling the PRG, simply by choosing fresh randomness to play the role of the PRG's output. Hence, we can return M with overwhelming probability without calling the PRG, so $H(\mathsf{M}|\mathsf{C}, \mathsf{U}_{\mathcal{A}})$ is negligible, even without access to the PRG.

Since $l \geq \lambda$, we have shown that given $\mathsf{C}, \mathsf{U}_{\mathcal{A}}$ for a qualified set $\mathcal{A}$, the entropy of M' drops by at least l' bits (up to a negligible amount), and this is the correctness property of Definition 4.

The security property of Definition 4 follows immediately from admissibility and from the security property of Definition 16: given only C, all seeds in $\mathcal{S}$ have full entropy and $H(\mathsf{M}|\mathsf{C}, \mathsf{U}_{\mathcal{A}}, PRG(\mathsf{C}, \mathsf{U}_{\mathcal{A}}))$ can only increase if we take away the PRG and therefore do not condition on $PRG(\mathsf{C}, \mathsf{U}_{\mathcal{A}})$.

We can now apply Theorem 8 and since we did not change the distribution of C, we conclude:

▶ **Lemma 17.** *For any BSS-scheme satisfying Definition 16 with $t = 0$, we have:*

$$H(\mathsf{C}) \geq n(\lambda - \delta(\lambda))/q.$$

Proof of Theorem 17. Given any BSS-scheme satisfying Definition 16, we can construct from this a new scheme for $n' = n - t$ receivers and threshold 0 (but the same ciphertext dsitribution). This is done by fixing the shared keys of the first t players and making them public, exactly as in the proof of Theorem 11, so we will not repeat the details here. We then apply the above lemma, and conclude that $H(\mathsf{C}) \geq (n - t)(\lambda - \delta(\lambda))/q$. We finally obtain Theorem 17 by also noting that C must carry enough information to determine the message, so we can add l to the lower bound.

◀

5.2 Upper Bound

Construction 2 describes how, using an idealized PRG in addition to shared keys, we can achieve

$$H(\mathsf{C}) = (n(t + 1)/(q + t) - t)\lambda + l.$$

▶ **Construction 2.** *The sender chooses a random PRG seed, uses the scheme from Construction 1 to share this seed among the receivers, and uses the PRG output on this seed to one-time-pad-encrypt the message.*

Ciphertext size and correctness follow trivially from Construction 1. As for security, it follows from security of Construction 1 that an unqualified set $\mathcal{A}$ of receivers has no information on the seed chosen by the sender. Hence the event that the (polynomial number of) inputs to the PRG chosen by $\mathcal{A}$ include the sender's seed has negligible probability. Unless this event happens, the message has full entropy, so the security property follows.

It is important to remark that, unlike Construction 1, Construction 2 does not give the receivers a Shamir secret sharing of the message, but rather of a PRG seed. The receivers reconstruct by first recovering the PRG seed, and then expanding that seed and using the resulting longer string to recover the message. The downside is that this not a linear secret sharing of the message. However, the upside is that, since a PRG seed can be used to generate an arbitrarily long pseudorandom string, the shared PRG seed can be re-used and the sender can share additional messages of length l to the same set of receivers by sending only l additional bits.

6 Application: Ad hoc Threshold Encryption

We can use any (l, n, t, q) BSS scheme together with any non-interactive key exchange (NIKE) scheme for $q + 1$ parties to get (l, n, t) ad hoc threshold encryption (ATE). Informally, the message sender uses the NIKE scheme to set up the correlated randomness for BSS non-interactively. She simply generates a fresh NIKE key pair, uses the secret key to derive shared secrets with every size-q subset of receivers, uses those shared secrets to run BSS, and sends the NIKE public key along with the resulting ciphertext to enable the recipients to derive the same shared secrets.

We sketch the definitions of NIKE and ATE below, and formalize how ATE can be instantiated from NIKE and BSS.

6.1 NIKE Definitions

A non-interactive key exchange (NIKE) scheme consists of two algorithms:

$KG(1^\lambda) \rightarrow (\mathsf{pk}, \mathsf{sk})$ is a randomized key generation algorithm that takes in the security parameter λ and returns a public-private key pair.

$KA(\mathsf{sk}_i, \mathsf{pk}_\mathcal{A}) \rightarrow \mathsf{s}$ is a key agreement algorithm that takes in one secret key and q public keys $\mathsf{pk}_\mathcal{A} = \{\mathsf{pk}_j\}_{j \in \mathcal{A}}$ and returns a shared secret.

Informally, a NIKE scheme for q parties is *correct* as long as, for any $i \in \mathcal{A}$ (where $|\mathcal{A}| = q + 1$), $\mathsf{s}_\mathcal{A} \leftarrow KA(\mathsf{sk}_i, \{\mathsf{pk}_j\}_{j \in \mathcal{A}, j \neq i})$ gives the same value. It is *secure* as long as, given $\{\mathsf{pk}_i\}_{i \in \mathcal{A}}$ (but none of the associated secret keys sk_i), $\mathsf{s}_\mathcal{A}$ is computationally indistinguishable from random.

6.2 ATE Definitions

An ad hoc threshold encryption (ATE) scheme consists of three algorithms:

$KG(1^\lambda) \rightarrow (\mathsf{pk}, \mathsf{sk})$ is a randomized key generation algorithm that takes in the security parameter λ and returns a public-private key pair.

$E_{\mathsf{pk}_\mathcal{R}}(\mathsf{m}) \rightarrow \mathsf{c}$ is an encryption algorithm that encrypts a message m to a set of public keys $\mathsf{pk}_\mathcal{R} = \{\mathsf{pk}_i\}_{i \in \mathcal{R}}$ belonging to the parties in the intended recipient set $\mathcal{R}$ in such a way that any size-$(t + 1)$ subset of the recipient set should jointly be able to decrypt.

$D_{\mathsf{pk}_\mathcal{R}, \mathsf{sk}_\mathcal{A}}(\mathsf{c}) \rightarrow \mathsf{m}$ is a decryption algorithm that uses secret keys $\mathsf{sk}_\mathcal{A} = \{\mathsf{sk}_i\}_{i \in \mathcal{A}}$ belonging to a subset $\mathcal{A}$ of the intended recipient set $\mathcal{R}$ (where $|\mathcal{A}| > t$) to decrypt the ciphertext c and recover the message m.

Informally, an (l, n, t) ATE scheme is *correct* if $D(E(\mathsf{M})) = \mathsf{M}$ (where D and E are run with the appropriate keys). It is *secure* if, for any two messages m_0 and m_1 of the same length l, $\mathsf{c}_0 = E_{\mathsf{pk}_\mathcal{R}}(\mathsf{M}_0)$ and $\mathsf{c}_1 = E_{\mathsf{pk}_\mathcal{R}}(\mathsf{M}_1)$ are computationally indistinguishable even given t or fewer of the secret keys sk_i, $i \in \mathcal{A}$.

6.3 ATE from NIKE and BSS

We can build an ATE scheme from a NIKE scheme and a BSS scheme as follows:

$KG(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$:
1. Return $(\mathsf{pk}, \mathsf{sk}) \leftarrow NIKE.KG(1^\lambda)$.

$E_{\mathsf{pk}_{\mathcal{R}}}(\mathsf{m})$:
1. Run $(\mathsf{pk}, \mathsf{sk}) \leftarrow NIKE.KG(1^\lambda)$.
2. For every size-q subset $\mathcal{A} \subseteq \mathcal{R}$, run $\mathsf{s}_{\mathcal{A}} \leftarrow NIKE.KA(\mathsf{sk}, \mathsf{pk}_{\mathcal{A}})$.
3. Run $BSS.\mathsf{c} \leftarrow BSS.E_{\mathsf{u}_{\mathcal{R}}}(\mathsf{m})$.
4. Return $(\mathsf{pk}, BSS.\mathsf{c})$.

$D_{\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{A}}}(\mathsf{c} = (\mathsf{pk}, BSS.\mathsf{c}))$:
1. For every party $i \in \mathcal{A}$, for every size-q subset $\mathcal{A}'$ such that $i \in \mathcal{A}'$, run

$$\mathsf{s}_{\mathcal{A}'} \leftarrow NIKE.KA(\mathsf{sk}_i, \{\mathsf{pk}\} \cup \{\mathsf{pk}_j\}_{j \in \mathcal{A}', j \neq i}).$$

2. Recall that $\mathsf{u}_{\mathcal{A}}$ denotes $\{\mathsf{s}_{\mathcal{A}'}\}_{\mathcal{A}' \cup \mathcal{A} \neq \emptyset}$. Return $\mathsf{m} \leftarrow BSS.D_{\mathsf{u}_{\mathcal{A}}}(BSS.\mathsf{c})$.

The size of a ciphertext in this scheme will be equal to the size of the corresponding BSS ciphertext plus the size of a NIKE public key.

6.4 From ATE and NIKE to BSS

Assume we have an ATE-scheme whose algorithms use an ideal NIKE functionality. We also assume that the ATE scheme is statistically secure when using the ideal NIKE functionality, that is, ciphertexts of different messages are statistically indistinguishable, and the message has full entropy in the view of a non-qualified set of receivers (up to a negligible amount).

From this, we can obtain a BSS scheme: the keys returned from the NIKE functionality become the correlated randomness, the encryption algorithm becomes the sharing algorithm, and the view of each receiver (including the ciphertext) is a share. Reconstruction is done by emulating the decryption protocol.

It therefore follows that our lower bound for BSS ciphertext size is also a lower bound for ciphertext size in any ATE scheme of the type we assumed.

References

AMMR18 Shashank Agrawal, Payman Mohassel, Pratyay Mukherjee, and Peter Rindal. Dise: distributed symmetric-key encryption. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1993–2010, 2018.

BZ14 Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 480–499. Springer, Heidelberg, August 2014.

CDI05 Ronald Cramer, Ivan Damgård, and Yuval Ishai. Share conversion, pseudorandom secret-sharing and applications to secure computation. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 342–362. Springer, Heidelberg, February 2005.

DDFY94 Alfredo De Santis, Yvo Desmedt, Yair Frankel, and Moti Yung. How to share a function securely. In *26th ACM STOC*, pages 522–533. ACM Press, May 1994.

DHMR08 Vanesa Daza, Javier Herranz, Paz Morillo, and Carla Ràfols. Ad-hoc threshold broadcast encryption with shorter ciphertexts. *Electron. Notes Theor. Comput. Sci.*, 192(2):3–15, May 2008.

DP08 Cécile Delerablée and David Pointcheval. Dynamic threshold public-key encryption. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 317–334. Springer, Heidelberg, August 2008.

GPSZ17 Sanjam Garg, Omkant Pandey, Akshayaram Srinivasan, and Mark Zhandry. Breaking the sub-exponential barrier in obfustopia. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 156–181. Springer, Heidelberg, April / May 2017.

HJK+16 Dennis Hofheinz, Tibor Jager, Dakshita Khurana, Amit Sahai, Brent Waters, and Mark Zhandry. How to generate and use universal samplers. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 715–744. Springer, Heidelberg, December 2016.

MZ17 Fermi Ma and Mark Zhandry. Encryptor combiners: A unified approach to multiparty NIKE, (H)IBE, and broadcast encryption. Cryptology ePrint Archive, Report 2017/152, 2017. `http://eprint.iacr.org/2017/152`.

RSY18 Leonid Reyzin, Adam Smith, and Sophia Yakoubov. Turning hate into love: Homomorphic ad hoc threshold encryption for scalable mpc. Cryptology ePrint Archive, Report 2018/997, 2018. `https://eprint.iacr.org/2018/997`.