J Cryptol (2017) 30:1067–1115
DOI: 10.1007/s00145-016-9242-8

Journal of
CRYPTOLOGY

Information Theoretical Cryptogenography

Sune K. Jakobsen
Department of Computer Science, University College London, London, UK
s.k.jakobsen@qmul.ac.uk

Communicated by Tal Rabin.

Received 25 September 2014 / Revised 19 May 2016
Online publication 14 October 2016

Abstract. We consider problems where n people are communicating and a random subset of them is trying to leak information, without making it clear who are leaking the information. We introduce a measure of suspicion and show that the amount of leaked information will always be bounded by the expected increase in suspicion, and that this bound is tight. Suppose a large number of people have some information they want to leak, but they want to ensure that after the communication, an observer will assign probability at most c to the events that each of them is trying to leak the information. How much information can they reliably leak, per person who is leaking? We show that the answer is $\left(\frac{-\log(1-c)}{c} - \log(e)\right)$ bits.

Keywords. Anonymity, Code-based cryptography, Cryptography, Information theory, Steganography.

1. Introduction

Consider a world with complete surveillance: everything you do, every letter you send, and every key you press on your keyboard is recorded by an adversary with unbounded computational power. Suppose further that you want to reveal a secret X, but if the adversary learns that you were trying to reveal X, it will punish you. What can you do?

The two obvious options are "do nothing", in which case X will not be revealed and "announce X" in which case you will be punished. Brody, Jakobsen, Scheder and Winkler [4] defined a game where a third option, "hinting at X", is better. Here "hinting" means sending a message with a probability distribution which may or may not depend on the value of X. Consider the following example:[1]

n people are active bloggers and each of them is a *leaker* with probability b, independently of each other. All the leakers want to reveal the same secret bit X. To do so, they all make sure that their next blog post is posted on a minute number with the same parity

[1] This example is inspired by [4] but the game in that paper only has one leaker, which makes in impossible to get a small error probability ϵ.

as X. If sufficiently many of the people are leakers, an observer, Frank, who knows that this leakage it going on and who sees the timestamps will, with probability at least $1-\epsilon$, be able find X by taking the majority of the parities of the timestamps. On the other hand, if not too many of the n people are leakers, any particular person will not look too likely to be a leaker. That is, for any particular person, another observer, Eve, who knows X and sees the entire communication, will compute the probability of that person being a leaker to be at most some $c < 1$.

If the secret X is not just one bit, but is known to be a string of h bits, we can just let the first $\frac{n}{h}$ people send the first bit as above, the next $\frac{n}{h}$ send the next bit, and so on. If n is sufficiently large, each of the h groups of people will with high probability contain a large number of leakers and the most common parity of the timestamp among people in this group will give you the hth bit of X.

These protocols will only work for particular values of n, h, b, c and ϵ; however, we will see that it is possible to create better protocols. The purpose of this paper is to find upper and lower bounds on how much information can be leaked this way. In order to do so, we will define a measure of suspicion, which exactly captures the loss of anonymity when hinting at information.

1.1. *Previous Work and Our Results*

There is a large body of research about how to get anonymity against a less powerful adversary. If we assume that each pair of people have access to common randomness that Eve cannot see, then we can use the Dining Cryptographers Protocol [6], and if we instead assume that Eve has bounded computational power, we could for example use MIXes [5]. Against an even weaker adversary, that does not have complete surveillance of the internet, we can use a network of onion routers, for example, the Tor network [11,17]. For a survey about anonymous communication methods, see [9], and for a currently up to date list of more than 300 papers in the area, see [15].

Many different ways of measuring anonymity have been suggested. Diaz, Seys, Claessens and Preneel [10] and independently Serjantov and Danezis [20] suggested to use entropy as a measure of how well a cryptographic protocol preserves anonymity. Clauß and Schiffner [7] suggested a measure based on Rényi-Entropy, Zhu and Bettati [25] suggested a measure based on mutual information, and several other measures have been suggested [3,14,16,22].

Another way of measuring the loss of anonymity is using the language of differential privacy [13]. In differential privacy, we say that a randomised mechanism $\mathcal{M}$ is (ϵ, δ)-differential private if, for any two *neighbouring* inputs x and y and any set $\mathcal{S}$ of possibly outputs, we have

$$\Pr(\mathcal{M}(x)) \leq e^{\epsilon} \Pr(\mathcal{M}(y)) + \delta.$$

Here x and y are said to be neighbouring if only one person's input have changed. Backes et al. [2] suggested a framework for measuring a loss of anonymity, using a similar inequality. For example, they say that a protocol has (ϵ, δ)-sender anonymity if an adversary's probability of outputting 1 increases by at most a factor e^{ϵ} plus a constant $e^{\epsilon}\delta$ whenever the sender of one message is changed. Van den Hooff et al. [23] suggested

a protocol, Vuvuzela, with a similar (ϵ, δ)-privacy guarantee. Protocols that provide (ϵ, δ)-differential privacy will both protect the fact that Alice sent a message and the fact that Bob did *not* send it. In this paper, we will assume that people do not mind being revealed as a non-sender.

Throughout the paper we assume that the observers have unbounded computational power and that the observers see all messages sent. The idea is to let every person send random messages and have the leakers make their message correlated with the secret X. For example the messages could be "I think X belongs to the set S". However, every time you make a correct hint about what the secret X is, it will increase the observer's suspicion that you know X. The more precise the hint is or the more unlikely it is that you would give the hint without knowing X, the more useful the statement is to Frank. But at the same time, such statements would also be the statements that increases Eve suspicion towards you the most. Our main contribution is to introduce a measure of suspicion that captures this and to show that if you want to leak some amount of information about X in the information theoretical sense, then your suspicion will, in expectation, have to increase by at least the same amount. This measure of suspicion turns out to be useful for showing upper bounds on how much information you can leak, without making it clear that you are leaking.

We consider n players, each of whom is a leaker with probability b independently of each other, and the leakers are told the secret value x taken by X which is uniformly distributed on $\{1, \ldots, 2^h\}$. The players want to communicate in such a way that an observer, Frank, with probability $1 - \epsilon$ can guess x based on the communication, but for any particular player, an observer, Eve, who knows x would never assign probability greater than c to the event that that player was leaking. We are interested in the number of bits $\frac{h}{n}$ that can be revealed per person and in the number of bits $\frac{h}{bn}$ that can be revealed per expected leaker. For example, for fixed c and ϵ is there a bound on the number of bits per expected leaker $\frac{h}{bn}$? Or could this value tend to infinity if $b \to 0$ while $n \to \infty$?

We show that the ratio $\frac{h}{bn}$ is bounded for sufficiently small ϵ. In fact, we show that the supremum of the values that can be achieved for all ϵ is given by $\frac{-b\log(1-c)+c\log(1-b)}{bc}$ for fixed b and by $\frac{-\log(1-c)}{c} - \log(e)$ if b can be arbitrarily small. Here e is the base of the natural logarithm. To show the upper bound we use the measure of suspicion, and to show the lower bound we use Shannon's noisy-channel coding theorem. We also consider a model where the total number of leakers, l, is fixed and known. Also in this model the ratio $\frac{h}{l}$ of bits per leaker is bounded by $\frac{-\log(1-c)}{c} - \log(e)$ independently of n. For $c = 0.95$ this gives around $3.1067\ldots$, so if a small group of leakers blend into a much larger group of people, they should be able to leak around 3.1 bits of information per leaker, while keeping reasonable doubt at the 5 % level.

The value c in this problem corresponds to a measure of anonymity suggested by Tóth et al. [22]. Communication is modelled as in communication complexity, with a model essentially identical to the one introduced by Yao [24].

Instead of some people being leakers all the time, and the rest being non-leakers all the time, we could also consider an adaptive model where players turn into leakers during the execution of the protocol. This gives an advantage to the leakers, because even people who are willing to leak information count as non-leakers until they start to leak information. In this adaptive model we need to make some choices about what

information the players have available when deciding whether to become leakers, and how to limit the number of player who turn into leakers. Different answers to these questions give six different models, and we find the capacities for two of these. For example, we show that if the players already know the secret when deciding whether to leak it, and we require that the expected number of leakers at the end of the protocol is at most bn, then asymptotically they can leak $\frac{-\min(b,c)\log(1-c)}{c} - \min(b,c)\log(e)$ bits of information per player.

The measure of suspicion is also useful for analysing a generalisation of the original cryptogenography (hidden-origin-writing) problem, as introduced in [4]. Here the authors considered a game where one person among n was randomly chosen and given the result of an otherwise secret coin flip. The goal for the n players is to communicate in such a way that an observer, Frank, would guess the correct result of the coin flip, but another observer, Eve, who has the same information, cannot guess who of the n players originally knew the result of the coin flip. The main method in [4] is a concavity characterisation, and is very different from the information theory methods we use. We generalise the problem so l players have the secret, and the secret is h bits. In this model we show that if $h = o(l)$ the winning probability tends to 1 and if $l = o(h)$ the winning probability tends to 0.

Finally, we show that in general to do cryptogenography, you do not need the non-leakers to collaborate. Instead, we can use the fact that people send out random messages anyway, and use this in a similar way to steganography (see [18]). All we need is that people are communicating in a way that involves sufficiently randomness and that they do not change this communication, when we build a protocol on top of that. We can for example assume that they are not aware of the protocol, or they do not care about the leakage.

1.2. *Paper Outline*

We define notation and recall some concepts and theorems from information theory and introduce a communication model in Sect. 2. In Sect. 3 we introduce a measure of suspicion and use this to show upper bounds on how much information the players can leak if they want Eve to have reasonable doubt that they are leaking. We use this in Sect. 4 where we turn to reliable leakage, and define and determine the capacity for some cryptogenography problems. Each of the next three sections builds on the first four, but can be read independently of each other. In Sect. 5 we define some adaptive models where non-leakers can turn into leakers, and find the capacities for two of these models. In Sect. 6 we show how our results from Sect. 4 can be used to analyse a generalisation of the original cryptogenography problem. Finally, in Sect. 7 we show that we can do equally well, even if the non-leakers are not collaborating in leaking, but are just communicating innocently. We end with an open problem section.

2. Preliminaries

Unless stated otherwise, all random variables in this paper are assumed to be discrete. Random variables are denoted by capital letters and they take values from the set denoted by the calligraphic version of the same letter (e.g. X takes values from $\mathcal{X}$). If X and Y are random variables and $\Pr(Y = y) > 0$, we let $X|_{Y=y}$ denote the random variable X

conditioned on $Y = y$. That is

$$\Pr(X|_{Y=y} = x) = \frac{\Pr(X = x, Y = y)}{\Pr(Y = y)}.$$

We typically write $\Pr(X = x|Y = y)$ instead of $\Pr(X|_{Y=y} = x)$.
For a tuple or infinite sequence a, we let a_i denote the i'th element of a, and let $a^i = (a_1, \ldots, a_i)$ be the tuple of the i first elements from a. We use similar notation if A is a tuple or sequence of random variables. For tuples a and b of n_a and n_b elements, we let $a \circ b$ denote the tuple $(a_1, \ldots, a_{n_a}, b_1, \ldots, b_{n_b})$. If a' is a single element, we abuse notation and write $a \circ a'$ instead of $a \circ (a')$.

In the rest of this subsection we will give some definitions and results from information theory. For an introduction to these concepts and for proofs, see [8]. For a random variable X and a value $x \in \mathcal{X}$ with $\Pr(X = x) > 0$ the *surprisal* or the *code-length*[2] of x is given by

$$-\log(\Pr(X = x)),$$

where log, as in the rest of this paper, is the base-2 logarithm.

The *entropy of* X, $H(X)$, is the expected code-length of X

$$\begin{aligned} H(X) &= \mathbb{E}\left[-\log(\Pr(X = x))\right] \\ &= -\sum_{x \in \mathcal{X}} \Pr(X = x) \log(\Pr(X = x)), \end{aligned}$$

where we define $0 \log(0) = 0$. If $p, q : \mathcal{X} \to [0, 1]$ are two probability distributions on $\mathcal{X}$ we have the inequality

$$-\sum_{x \in \mathcal{X}} p(x) \log(p(x)) \leq -\sum_{x \in \mathcal{X}} p(x) \log(q(x)), \tag{1}$$

with equality if and only if $p = q$ [8]. One interpretation is, if X's distribution is given by p, and you encode values of X using a code optimised to the distribution q, you get the shortest average code-length if and only if $p = q$.

The entropy of a random variable X can be thought of as the uncertainty about X, or as the amount of information in X. For a tuple of random variables $(X_1, \ldots, X_k)$ the entropy $H(X_1, \ldots, X_k)$ is simply the entropy of the random variable $(X_1, \ldots, X_k)$. The *entropy of* X *given* Y, $H(X|Y)$ is

$$H(X|Y) = \sum_{y \in \mathcal{Y}} \Pr(Y = y) H(X|_{Y=y}). \tag{2}$$

A simple computation shows that

[2] If $-\log(\Pr(X = x))$ is an integer for all $x \in \mathcal{X}$, and we want to find an optimal prefix-free binary code for X, the length of the code for x should be $-\log(\Pr(X = x))$, thus the name code-length. If they are not integers, we can instead use $\lceil -\log(\Pr(X = x)) \rceil$ and waste at most one bit.

$$H(X|Y) = H(X, Y) - H(Y).$$

The *mutual information* $I(X; Y)$ of two random variables X, Y is given by

$$I(X; Y) = H(X) + H(Y) - H(X, Y) = H(Y) - H(Y|X).$$

This is known to be nonnegative. The *mutual information* $I(X; Y|Z = z)$ *of* X *and* Y *given* $Z = z$ is given by

$$I(X; Y|Z = z) = I(X|_{Z=z}; Y|_{Z=z}),$$

where the joint distribution of $(X|_{Z=z}, Y|_{Z=z})$ is given by $(X, Y)|_{Z=z}$. The *mutual information* $I(X; Y|Z)$ *of* X *and* Y *given* Z is

$$I(X; Y|Z) = \mathbb{E}_z I(X; Y|Z = z).$$

A simple computation shows that

$$I(X; Y|Z) = H(X, Z) + H(Y, Z) - H(X, Y, Z) - H(Z).$$

We will need the chain rule for mutual information,

$$I(X; (T_1, \ldots, T_k)) = \sum_{i=1}^{k} I(X; T_i|(T_1, \ldots, T_{i-1})).$$

Let X and Y be random variables, and $f : \mathcal{Y} \to \mathcal{X}$ a function. We think of $f(Y)$ as a guess about what X is. The probability of error, P_e is now $\Pr(f(Y) \neq X)$. We will need (a weak version of) Fano's inequality,

$$P_e \geq \frac{H(X|Y) - 1}{\log(|\mathcal{X}|)}. \tag{3}$$

A *discrete memoryless channel* (or *channel* for short) q consists of a finite *input set* $\mathcal{Y}$, a finite *output set* $\mathcal{Z}$ and for each element $y \in \mathcal{Y}$ of the input set a probability distribution $q(z|y)$ on the output set. If Alice has some information X that she wants Bob to know, she can use a channel. To do that, Alice and Bob will have to both know a code. An *error correcting code*, or simply a *code*, $\mathfrak{C} : \mathcal{X} \to \mathcal{Y}^n$ is a function that for each $x \in \mathcal{X}$ specifies what Alice should give as input to the channel. Here n is the *length* of the code. Now the probability that Bob receives $Z_{\mathfrak{C}} = z_1 \ldots z_n$ when $X = x$ is given by

$$\Pr(Z_{\mathfrak{C}} = z|X = x) = \prod_{i=1}^{n} q\left(z_i|\mathfrak{C}(x)_i\right).$$

Bob will then use a *decoding function* D which sends outputs z to guesses about X. A *rate* R is *achievable* if for all $\epsilon > 0$ there is a $n > 0$ such that for X uniformly distributed

on $\{1, \ldots, 2^{\lceil Rn \rceil}\}$ there is a code $\mathfrak{c}$ of length n for q and a decoding function D giving $\Pr(D(Z) = x | X = x) > 1 - \epsilon$ for all $x \in \mathcal{X}$.

For a distribution p on the input set $\mathcal{Y}$, we get a joint distribution of (Y, Z) given by $\Pr(Y = y, Z = z) = p(y)q(z|y)$. Now define the capacity C of q to be

$$C = \max_p I(Y; Z),$$

where max is over all distributions p of Y and the joint distribution of (Y, Z) is as above. Shannon's noisy-channel coding theorem says that any rate below C is achievable, and no rate above C is achievable [21].

2.1. *Model*

In this paper we consider problems where one or more players might be trying to leak information about the outcome of a random variable X. The number of players is denoted n and the players are called $\text{PLR}_1, \ldots, \text{PLR}_n$. Sometimes we will call PLR_1 Alice and PLR_2 Bob. We let L_i be the random variable that is 1 if player i knows the information and 0 otherwise. If there is only one player we write L instead of L_1. The joint distribution of $(X, L_1, \ldots, L_n)$ is known to everyone.

All messages are broadcast to all players and to two observers, Eve and Frank. They can both see all the communication, and Eve also knows the value taken by X. We want to reveal information about X to Frank, while at the same time make sure that for all i, Eve does not get too sure that $L_i = 1$. The players will send messages, each chosen from a distribution given by a protocol as defined below. The tuple t of all messages sent is called a transcript. As each message is chosen randomly, we can consider the transcript to be an instance of a random variable T, which we will also refer to as the transcript. These are tuples of messages, so we can use the notation T^k, T_k, t^k, t_k as define in the beginning of this section. For example, T^k denotes the random variable that gives the tuple of the first k messages.

In this section we define the collaborating model. In Sect. 7 we will define a model, where we do not need the non-leakers to collaborate. The model in Sect. 7 will be more useful in practice; however, when constructing protocols, it is easier first to construct them in the collaborating model (we will also define a different model in Sect. 5). In the *collaborating model* we can tell all the players, including the non-leaking players, to follow some communication protocol, called a collaborating cryptogenography protocol. The messages send by a leaking player may depend on the value of X, but the messages of non-leaking players have to be independent of X given the previous transcript. Formally, a *collaborating cryptogenography protocol* π specifies for any possible value t^k of the current transcript T^k:

- Should the communication stop or continue, and if it should continue,
- Who is next to send a message, say PLR_i, and
- A distribution $p_?$ and a set of distributions, $\{p_x\}_{x \in \mathcal{X}}$ (the distributions $p_?$ and $\{p_x\}_{x \in \mathcal{X}}$ depend on π and t^k). Now PLR_i should choose a message using $p_?$, if $L_i = 0$ and choose a message using p_x if $L_i = 1$ and $X = x$.

Furthermore, for any protocol π, there should a number $length(\pi)$ such that the protocol will always terminate after at most $length(\pi)$ messages. We assume that both Frank and Eve know the protocol. They also know the prior distribution of $(X, L_1, \ldots, L_n)$, and we assume that they have computational power to compute $(X, L_1, \ldots, L_n)|_{T=t}$ for any transcript t. Notice that this assumption rules out the use of cryptography.

Another way of stating the above definition of collaborating cryptogenography protocols is that the players follow a communication protocol,[3] and the leakers are given x as input while the non-leakers are given a fixed input, say "you are not a leaker", which is not in $\mathcal{X}$.

One way that everyone can know the protocol, is if one person, e.g., Frank, announces the protocol that they will use, and we assume that everyone follows that protocol. Another possibility is that the players and Frank and Eve (or their ancestors) have played a game about leaking information many times and slowly developed (or evolved) a protocol for leaking information and learned (or evolved) to play the game optimally. In this paper we will not consider the question of whether and how the protocol could be developed or evolved.

While we think of different players as different people, two or more different players could be controlled by the same person. For example, they could be communicating using a service that preserves anonymity, except that a profile's identity will be revealed if the profile can be shown to be guilty in leaking with probability greater than 95 %. Here each player would correspond to a profile, but the same person could have more profiles. However, we will use "player" and "person" as synonyms in the paper.

3. Bounds on $I(X; T)$

3.1. *Suspicion*

First we will look at the problem where only one player is communicating and she may or may not be trying to leak information. We will later use these results when we analyse the many-player problem.

In the one-player case, Alice sends one message A. If she is not trying to leak information, she will choose this message in $\mathcal{A}$ randomly using a distribution $p_?$. If she is trying to leak information, and $X = x$, she will use a distribution p_x. For a random variable Y jointly distributed with L and a value $y \in \mathcal{Y}$ with $\Pr(Y = y) > 0$ we let $c_{Y=y} = \Pr(L = 1|Y = y)$. We usually suppress the random variable, and write c_y instead. Here Y could be a tuple of random variables, and y a tuple of values. If $y = (y_1, y_2)$ is a tuple, we write $c_{y_1 y_2}$ instead of $c_{(y_1, y_2)}$.

We want to see how much information Alice can leak to Frank, without being too suspicious to Eve. The following measure of suspicion turns out to be useful.

[3] These were first defined by Yao [24]. Unlike in [24] we allow more than two players, allow the protocol to specify who to send the next message, and allow each message to be more than one bit. All this is standard in communication complexity, see for example [19].

Definition 1. Let Y be a random variable jointly distributed with L. Then the *suspicion (of Alice) given* $Y = y$ is

$$\begin{aligned}\mathrm{susp}(Y = y) &= -\log(1 - c_y) \\ &= -\log(\Pr(L = 0|Y = y)).\end{aligned}$$

We see that $\mathrm{susp}(Y = y)$ depends on y and the joint distribution of L and Y, but to keep notation simple, we suppress the dependence on L. The suspicion of Alice measures how suspicious Alice is to someone who knows that $Y = y$ and knows nothing more. For example Y could be the tuple that consists of the secret information X and the current transcript.

We can think of the suspicion as the surprisal of the event "Alice did not have the information". Next we define the suspicion given a random variable Y, without setting it equal to something.

Definition 2. The *expected suspicion (of Alice) given* Y or just the *suspicion (of Alice) given* Y is

$$\begin{aligned}\mathrm{susp}(Y) &= \mathbb{E}_y \mathrm{susp}(Y = y) \\ &= \sum_{y \in \mathcal{Y}} \Pr(Y = y)\mathrm{susp}(Y = y). \end{aligned} \tag{4}$$

In each of these definitions, Y can consist of more than one random variable, e.g. $Y = (X, A)$. Finally, we can also combine these two definitions, giving

$$\mathrm{susp}(X, A = a) = \sum_{x \in \mathcal{X}} \Pr(X = x|A = a)\mathrm{susp}((X, A) = (x, a)),$$

where X and A can themselves be tuples of random variables.

The definitions imply that

$$\mathrm{susp}(X, A) = \sum_{a \in \mathcal{A}} \Pr(A = a)\mathrm{susp}(X, A = a),$$

which can be thought of as (4) given X.

When Alice sends a message A this might reveal some information about X, but at the same time, she will also reveal some information about whether she is trying to leak X. We would like to bound $I(A; X)$ by the information A reveals about L. This is not possible. If, for example, we set $A = X$ whenever $L = 1$ and $A = a \notin \mathcal{X}$ when $L = 0$, then $I(A; X) = \Pr(L = 1)H(A)$ which can be large. However, we have $I(A; L) \leq H(L) \leq 1$. The lemma below shows that instead, $I(A; X)$ can be bounded by the expected increase in suspicion given X, and that this bound is tight.

Lemma 1. *If Alice sends a message A, we have*

$$I(X;\, A) \leq susp(X,\, A) - susp(X).$$

That is, the amount of information she sends about X is at most her expected increase in suspicion given X. There is equality if and only if the distribution of A is the same as $A|_{L=0}$.

Proof. With no information revealed, Alice's suspicion given X is

$$\mathrm{susp}(X) = -\sum_{x\in\mathcal{X}} \Pr(X = x)\log(1 - c_x).$$

We want to compute Alice's suspicion given X and her message A.

$$\begin{aligned}\mathrm{susp}(X,\, A) &= \sum_{x,a} \Pr(X = x,\, A = a)\mathrm{susp}(X = x,\, A = a)\\ &= -\sum_{x,a} \Pr(X = x,\, A = a)\log(1 - c_{xa})\\ &= -\sum_{x,a} \Pr(X = x,\, A = a)\left(\log(1 - c_x) + \log\left(\frac{1 - c_{xa}}{1 - c_x}\right)\right).\end{aligned}$$

Now it follows that the cost in suspicion given X of sending A is

$$\mathrm{susp}(X,\, A) - \mathrm{susp}(X) = -\sum_{x,a} \Pr(X = x,\, A = a)\log\left(\frac{1 - c_{xa}}{1 - c_x}\right). \tag{5}$$

Next we want to see how much information A gives about X, that is $I(A;\, X) = H(A) - H(A|X)$. We claim that this is bounded by the cost in suspicion, or equivalently, $H(A) \leq \mathrm{susp}(X,\, A) - \mathrm{susp}(X) + H(A|X)$. First we compute $H(A|X)$ using (2):

$$\begin{aligned}H(A|X) &= \sum_x \Pr(X = x)H(A|X = x)\\ &= -\sum_x \Pr(X = x)\sum_a \Pr(A = a|X = x)\log(\Pr(A = a|X = x))\\ &= -\sum_{x,a} \Pr(X = x,\, A = a)\log(\Pr(A = a|X = x)).\end{aligned} \tag{6}$$

We have

$$\begin{aligned}&\frac{1 - c_{xa}}{1 - c_x}\Pr(A = a|X = x)\\ &\quad= \frac{\Pr(L = 0|X = x,\, A = a)}{\Pr(L = 0|X = x)}\Pr(A = a|X = x)\end{aligned}$$

$$
\begin{aligned}
&= \frac{\Pr(L=0, X=x, A=a)}{\Pr(X=x, A=a)} \frac{\Pr(X=x)}{\Pr(L=0, X=x)} \frac{\Pr(X=x, A=a)}{\Pr(X=x)} \\
&= \frac{\Pr(L=0, X=x, A=a)}{\Pr(L=0, X=x)} \\
&= \Pr(A=a|X=x, L=0) \\
&= \Pr(A=a|L=0) \qquad (7)
\end{aligned}
$$

Here, the last equation follows from the assumption that A is independent of X when $L = 0$. From this we conclude

$$
\begin{aligned}
&\operatorname{susp}(X, A) - \operatorname{susp}(X) + H(A|X) \\
&\quad = -\sum_{x,a} \Pr(X=x, A=a) \log\left(\frac{1-c_{xa}}{1-c_x} \Pr(A=a|X=x)\right) \\
&\quad = -\sum_{x,a} \Pr(X=x, A=a) \log\left(\Pr(A=a|L=0)\right) \\
&\quad = -\sum_{a} \Pr(A=a) \log\left(\Pr(A=a|L=0)\right) \\
&\quad \geq -\sum_{a} \Pr(A=a) \log(\Pr(A=a)) \\
&\quad = H(A).
\end{aligned}
$$

Here the first equality follows from (5) and (6), the second follows from (7) and the inequality follows from inequality (1). There is equality if and only if $\Pr(A = a) = \Pr(A = a|L = 0)$ for all a. □

We will now turn to the problem where many people are communicating. We assume that they send messages one at a time, so we can break the protocol into time periods where only one person is communicating, and see the entire protocol as a sequence of one-player protocols. To make the notation simpler, we will assume that the protocol runs for a fixed number of messages, and the player to talk in round k only depends on k, not on which previous messages was sent. Any protocol π can be turned into such a protocol π' by adding dummy messages: In round k of π' we let $\text{PLR}_{k \bmod n}$ talk. They follow protocol π in the sense that if it is not $\text{PLR}_{k \bmod n}$'s turn to talk according to π she sends some fixed message 1, and if it is her turn, she chooses her message as in π. The following Corollary show that a statement similar to Lemma 1 holds for each single message in a protocol with many players.

Corollary 2. *Let (L, T^{k-1}, X) have some joint distribution, where T^{k-1} denotes previous transcript. Let T_k be the next message sent by Alice. Then*

$$
I\left(X; T_k|T^{k-1}\right) \leq \mathit{susp}\left(X, T^k\right) - \mathit{susp}\left(X, T^{k-1}\right).
$$

Proof. For a particular value t^{k-1} of T^{k-1} we use Lemma 1 with $(X, T_k)|_{T^{k-1}=t^{k-1}}$ as (X, A) to get

$$I\left(X; T_k|T^{k-1} = t^{k-1}\right) \leq \operatorname{susp}\left(X, T_k, T^{k-1} = t^{k-1}\right) - \operatorname{susp}\left(X, T^{k-1} = t^{k-1}\right).$$

By multiplying each side by $\Pr(T^{k-1} = t^{k-1})$ and summing over all possible t^{k-1} we get the desired inequality. □

A protocol consists of a sequence of messages that each leaks some information and increases the suspicion of the sender. We can add up the increases in suspicion, and using the chain rule for mutual information we can also add up the amount of revealed information. However, Bob's message might not only affect his own suspicion, it might also affect Alice's suspicion. To show an upper bound on the amount of information the players can leak, we need to show that one person's message will, in expectation, never make another person's suspicion decrease. We get this from the following proposition by setting $Y = (X, T^{k-1})$ and $B = T_k$.

Proposition 3. *For any joint distribution on (L, Y, B) we have $susp(Y) \leq susp(Y, B)$.*

Proof. We have

$$\begin{aligned}
\operatorname{susp}(Y = y) &= -\log(\Pr(L = 0|Y = y)) \\
&= -\log\left(\sum_{b\in\mathcal{B}} \Pr(B = b|Y = y)\Pr(L = 0|Y = y, B = b)\right) \\
\operatorname{susp}(Y = y, B) &= -\sum_{b\in\mathcal{B}} \Pr(B = b|Y = y)\log\Pr(L = 0|Y = y, B = b). \qquad (8)
\end{aligned}$$

As $p \mapsto -\log(p)$ is convex, Jensen's inequality gives us

$$\operatorname{susp}(Y = y, B) \geq \operatorname{susp}(Y = y).$$

Multiplying each side by $\Pr(Y = y)$ and summing over all $y \in \mathcal{Y}$ gives us the desired inequality. □

Let susp_i denote the suspicion of PLR_i.[4]

Theorem 4. *If T is the transcript of the entire protocol we have*

$$I(X; T) \leq \sum_{i=1}^{n} \left(susp_i(X, T) - susp_i(X)\right).$$

[4]This is defined similar to the suspicion of Alice, except using L_i instead of L.

Proof. From the chain rule for mutual information, we know that

$$I(X;T) = \sum_{k=1}^{\text{length}(\pi)} I\left(X; T_k | T^{k-1}\right).$$

Now Corollary 2 shows that $I(X; T_k|T^{k-1}) \leq \text{susp}_i(X, T^k) - \text{susp}_i(X, T^{k-1})$ if PLR_i sends the k'th message and Proposition 3 shows that $\text{susp}_{i'}(X, T^k) \geq \text{susp}_{i'}(X, T^{k-1})$ for all other i'. Summing over all rounds in the protocol, we get the theorem. □

3.2. *Keeping Reasonable Doubt*

Until now we have bounded the amount of information the players can leak by the expected increase in some strange measure, suspicion, that we defined for the purpose. But there is no reason to think that someone who is leaking information cares about the expectation of this measure. A more likely scenario, is that each person leaking wants to ensure that after the leakage, an observer will assign probability at most c to the event that she was leaking information. If this is the case after all possible transcripts t, we see that $\text{susp}_i(X, T) \leq -\log(1-c)$. If we assume that each player before the protocol had probability $b < c$ of leaking independently of X, that is $\Pr(L_i|X = x) = b$ for all x and i, we have $\text{susp}_i(X) = -\log(1-b)$. Thus

$$I(X;T) \leq \sum_{i=1}^{n} \left(\text{susp}_i(X,T) - \text{susp}_i(X)\right) = (\log(1-c) + \log(1-b))\, n. \quad (9)$$

To reach this bound, we would need to have $\Pr(L_i = 1|X = x, T = t) = c$ for all x, t, i. But the probability $\Pr(L_i = 1|X = x) = b$ can also be computed as $\mathbb{E}_t \Pr(L_i = 1|X = x, T = t)$, so $\Pr(L_i = 1|X = x, T = t)$ cannot be constantly $c > b$. The following theorem improves the upper bound from (9) by taking this into account.

Theorem 5. *Let π be a collaborating cryptogenography protocol, and T be its transcript. If for all players PLR_i and all $x \in \mathcal{X}$ and all transcripts t we have $\Pr(L_i = 1|X = x) = b$, and $\Pr(L_i = 1|T = t, X = x) \leq c$ then*

$$I(X;T) \leq \frac{-b\log(1-c) + c\log(1-b)}{c} n.$$

For an illustration of this theorem, see Fig. 1.

Proof. If $\Pr(L_i = 1|X = x, T = t) \leq c$ then

$$\begin{aligned} \text{susp}_i(X = x, T = t) &= -\log(1 - \Pr(L_i = 1|X = x, T = t)) \\ &\leq \frac{-\log(1-c)}{c} \Pr(L_i = 1|X = x, T = t). \end{aligned} \quad (10)$$

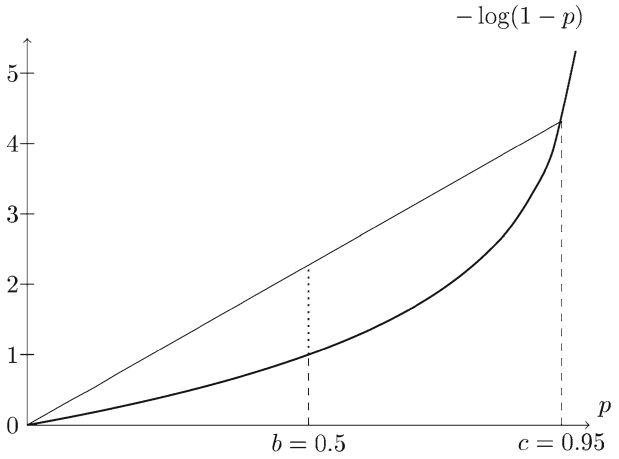

Fig. 1. This figure illustrates Theorem 5. The curve shows the function $p \mapsto -\log(1-p)$, which is used for computing the suspicion. The line from $(0, 0)$ to $(c, -\log(1-c))$ shows the maximum expected posterior suspicion PLR_i can have if he started with $\Pr(L_i = 1|X = x) = p$ and we have $\Pr(L_i|X = x, T = t) \leq c$ for all transcripts t. The second coordinate of $(b, -\log(1-b))$ gives the prior suspicion towards PLR_i, so the dotted line gives the amount of information that player i can leak.

This follows from the fact that we have equality when $\Pr(L_i = 1|X = x, T = t)$ is 0 or c, and the left hand side is convex in $\Pr(L_i = 1|X = x, T = t)$ while the right hand side is linear.

Let π and T be as in the assumptions. Now we get

$$\begin{aligned}
\text{susp}_i(X, T) &= \sum_{x,t} \Pr(X = x, T = t)\text{susp}_i(X = x, T = t) \\
&\leq \sum_{x,t} \Pr(X = x, T = t)\frac{-\log(1-c)}{c} \Pr(L_i = 1|X = x, T = t) \\
&= \sum_{x,t} \frac{-\log(1-c)}{c} \Pr(L_i = 1, X = x, T = t) \\
&= \frac{-\log(1-c)}{c} \Pr(L_i = 1) \\
&= \frac{-b\log(1-c)}{c}.
\end{aligned}$$

Thus,

$$\begin{aligned}
I(X; T) &\leq \sum_{i=1}^{n} \big(\text{susp}_i(X, T) - \text{susp}_i(X)\big) \\
&\leq \left(\frac{-b\log(1-c)}{c} - (-\log(1-b))\right) n
\end{aligned}$$

$$= \frac{-b\log(1-c) + c\log(1-b)}{c} n. \qquad \square$$

It is clear that the upper bound from Theorem 5 cannot be achieved for all distributions of $(X, L_1, \ldots, L_n)$. If for example $H(X) < \frac{-b\log(1-c)+c\log(1-b)}{c} n$ we must also have $I(X,T) \leq H(X) < \frac{-b\log(1-c)+c\log(1-b)}{c} n$, that is, the players do not have enough information to send to reach the upper bound. Even if $H(X)$ is high, we may not be able to reach the upper bound. If it is known that $L_1 = L_2 = \cdots = L_n$ the suspicion of the players will not depend on the player, only on the messages sent. So this problem will be equivalent to the case where only one person is sending messages.

We will now give an example where the upper bound from Theorem 5 is achievable. We will refer back to this example when we prove that reliable leakage is possible.

Example 1. Assume that $X, L_1, \ldots, L_n$ are all independent, and $\Pr(L_i = 1) = b$ for all i. Furthermore, assume that $0 < b < c < 1$ and that $\frac{b(1-c)}{c(1-b)}$ is a rational number. Let $d, a \in \mathbb{N}$ be the smallest natural numbers such that $\frac{a}{d} = \frac{b(1-c)}{c(1-b)}$. We see that $\frac{b(1-c)}{c(1-b)} \in (0,1)$ so $0 < a < d$. We will assume that X is uniformly distributed on $\{1, \ldots, d\}^n$.

Each player PLR_i now sends one message, independently of which messages the other players send. If $L_i = 0$, PLR_i chooses a message in $\{1, \ldots, d\}$ uniformly at random. If $L_i = 1$ and $X_i = x_i$, then PLR_i chooses a message in

$$\{1 + (x_i - 1)a, 2 + (x_i - 1)a \ldots, x_i a\} \mod d$$

uniformly at random.[5]

We see that over random choice of X, the message, A_i, that PLR_i sends, is uniformly distributed on $\{1, \ldots, d\}$, so $H(A_i) = \log(d)$. We want to compute $H(A_i|X)$. Given X, each of the $d-a$ elements not in $\{1+(x_i-1)a, 2+(x_i-1)a \ldots, x_i a\} \mod d$ can only be sent if $L = 0$, so they will be send with probability $\frac{1-b}{d}$. Each of the a elements in the set $\{1+(x_i-1)a, 2+(x_i-1)a \ldots, x_i a\} \mod d$ are sent with probability $\frac{b}{a} + \frac{1-b}{d}$. Thus,

$$\begin{aligned} H(A_i|X = x) &= -\sum_{t_i \in \mathcal{A}_i} \Pr(A_i = t_i|X = x) \log(\Pr(A_i = t_i|X = x)) \\ &= -a\left(\frac{b}{a} + \frac{1-b}{d}\right)\log\left(\frac{b}{a} + \frac{1-b}{d}\right) - (d-a)\frac{1-b}{d}\log\left(\frac{1-b}{d}\right) \\ &= -\frac{b}{c}\log\left(\frac{1-b}{d(1-c)}\right) - \left(1 - \frac{b}{c}\right)\log\left(\frac{1-b}{d}\right). \end{aligned}$$

[5] We use $k \mod d$ to mean the number in $\{1, \ldots d\}$ that is equal to k modulo d.

The last equality follows from three uses of $\frac{a}{d} = \frac{b(1-c)}{c(1-b)}$, or of its equivalent formulation, $\frac{b}{a} + \frac{1-b}{d} = \frac{b}{ac}$. As this holds for all x, we get.

$$H(A_i|X) = -\frac{b}{c}\log\left(\frac{1-b}{d(1-c)}\right) - \left(1-\frac{b}{c}\right)\log\left(\frac{1-b}{d}\right).$$

Now

$$\begin{aligned} I(A_i;X) &= H(A_i) - H(A_i|X) \\ &= \log(d) + \frac{b}{c}\log\left(\frac{1-b}{d(1-c)}\right) + \left(1-\frac{b}{c}\right)\log\left(\frac{1-b}{d}\right) \\ &= \log(1-b) - \frac{b}{c}\log(1-c) \\ &= \frac{-b\log(1-c) + c\log(1-b)}{c}. \end{aligned} \tag{11}$$

The tuples (X_i, A_i, L_i) where i ranges over $\{1, \ldots n\}$ are independent from each other, so we have $I(T;X) = \frac{-b\log(1-c)+c\log(1-b)}{c}n$ as wanted.

Next we want to compute $\Pr(L_i = 1|T = t, X = x)$. This is 0 if PLR$_i$ send a message not in $\{1 + (x_i - 1)a, 2 + (x_i - 1)a \ldots, x_i a\} \mod d$. Otherwise, we use independence and then Bayes' theorem to get

$$\begin{aligned} \Pr(L_i = 1|T = t, X = x) &= \Pr(L_i = 1|A_i = t_i, X_i = x_i) \\ &= \frac{\Pr(A_i = t_i|L_i = 1, X_i = x_i)\Pr(L_i = 1|X_i = x_i)}{\Pr(A_i = t_i|X_i = x_i)} \\ &= \frac{\frac{1}{a}b}{\frac{b}{a} + \frac{1-b}{d}} \\ &= \frac{\frac{b}{a}}{\frac{b}{ac}} \\ &= c. \end{aligned} \tag{12}$$

As we wanted.

4. Reliable Leakage

In the previous example, Frank would receive some information about X in the sense of information theory: Before he sees the transcript, any value of X would be as likely as any other value, and when he knows the transcript, he has a much better idea about what X is. However, his best guess about what X is, is still very unlikely to be correct. Next we want to show that we can have reliable leakage. That is, no matter what value X is taking, we want Frank to be able to guess the correct value with high probability. We will see that this is possible, even when X has entropy close to $\frac{-b\log(1-c)+c\log(1-b)}{c}n$. Frank's guess

would have to be a function D of the transcript t. Saying that Frank will guess X correct with high probability when $X = x$ is that same as saying that $\Pr(D(T) = x|X = x)$ is close to one.

Definition 3. Let $L = (L_1, \ldots, L_n)$ be a tuple of random variables, where the L_i takes values in $\{0, 1\}$.

A *risky* (n, h, L, c, ϵ)*-protocol* is a collaborating cryptogenography protocol together with a function D from the set of possible transcripts to $\mathcal{X} = \{1, \ldots, 2^{\lceil h \rceil}\}$ such that when X and L are distributed independently and X is uniformly distributed on $\mathcal{X}$, then for any $x \in \mathcal{X}$, there is probability at least $1 - \epsilon$ that a random transcript t distributed as $T|_{X=x}$ satisfies

Reasonable doubt: $\forall i : \Pr(L_i = 1|T = t, X = x) \leq c$, and
Reliable leakage: $D(t) = x$

That is, no matter the value of X, with high probability Frank can guess the value of X, and with high probability no player will be estimated to have leaked the information with probability greater than c by Eve. However, there might be a small risk that someone will be estimated to have leaked the information with probability greater than c. This is the reason we call it a risky protocol. A safe protocol is a protocol where this never happens.

Definition 4. A *safe* (n, h, L, c, ϵ)*-protocol* is a risky (n, h, L, c, ϵ)-protocol where $\Pr(L_i = 1|T = t, X = x) \leq c$ for all i, t, x with $\Pr(T = t, X = x) > 0$.

First we will consider the case where $L_1, \ldots, L_n$ are independent and identically distributed. The following definitions of achievability and capacity are based on the similar definitions from information theory, as given by Shannon [21], but instead of measuring these in bits per time unit or per usage of a channel, we measure them in bits per player.

Definition 5. Let $\text{Indep}_b(n)$ be the random variable $(L_1, \ldots, L_n)$ where $L_1, \ldots, L_n$ are independent, and each L_i is distributed on $\{0, 1\}$ and $\Pr(L_1 = 1) = b$.

A rate R is *safely/riskily c-achievable for* Indep_b if for all $\epsilon > 0$ and all n_0, there exists a safe/risky $(n, nR, \text{Indep}_b(n), c, \epsilon)$-protocol with $n \geq n_0$.

The *safe/risky c-capacity for* Indep_b is the supremum of all safely/riskily c-achievable rates for Indep_b.

It turns out that the safe and the risky c-capacities for Indep_b are the same, but at the moment we will only consider the safe capacity.

Proposition 6. *No rate* $R > \frac{-b\log(1-c)+c\log(1-b)}{c}$ *is safely c-achievable for* Indep_b.

Proof. Assume for contradiction that $R > \frac{-b\log(1-c)+c\log(1-b)}{c}$ is safely c-achievable for Indep_b, and let π be a safe $(n, Rn, \text{Indep}_b(n), c, \epsilon)$-protocol. Let $\delta = R - \frac{-b\log(1-c)+c\log(1-b)}{c}$. We know from Theorem 5 that

$$I(X;T) \leq \frac{-b\log(1-c)+c\log(1-b)}{c}n = (R-\delta)n.$$

Now

$$H(X|T) = H(X) - I(X;T) \geq Rn - (R-\delta)n = \delta n.$$

By Fano's inequality (3) we get that the probability of error for Frank's guess is

$$P_e \geq \frac{\delta n - 1}{nR}.$$

Thus, for sufficiently large n_0 and sufficiently small ϵ we cannot have $n \geq n_0$ and $P_e \leq \epsilon$. When $P_e > \epsilon$ there must exist an $x \in \mathcal{X}$ such that $\Pr(D(T) \neq x | X = x) > \epsilon$, so R is not safely c-achievable. □

In the example in the introduction, it was suggested that if X consists of many bits, we could divide the n people into h groups, and let each group reveal one bit. This protocol can be seen as considering each person to be a channel, where the protocol corresponds to using a repetition code. Repetition codes are very inefficient, so no positive rate is achievable using such protocols. However, if we use Shannon's noisy-channel coding theorem we can improve the protocol and achieve any rate $R < \frac{-b\log(1-c)+c\log(1-b)}{c}$.

Theorem 7. *Any rate* $R < \frac{-b\log(1-c)+c\log(1-b)}{c}$ *is safely* c*-achievable for* Indep_b.

Proof. Let $R < \frac{-b\log(1-c)+c\log(1-b)}{c}$ and let $c' \leq c$ be a number such that $\frac{b(1-c')}{c'(1-b)}$ is rational and $R < \frac{-b\log(1-c')+c'\log(1-b)}{c'}$. Now use b and c' to define a and d as in Example 1. We consider the channel that on input j with probability b returns a random uniformly distributed element in $\{1+(j-1)a, 2+(j-1)a \ldots, ja\} \mod d$, and with probability $1-b$ it returns a random and uniformly distributed element in $\{1, \ldots, d\}$. We see that each person sending a message, exactly corresponds to using this channel. The computation (11) from Example 1 shows that when input of this channel is uniformly distributed, the mutual information between input and output is $\frac{-b\log(1-c')+c'\log(1-b)}{c'}$. Thus, the capacity of the channel is at least this value (in fact, it is this value). We now use Shannon's noisy channel coding theorem [8,21] to get an error correcting code $\mathfrak{C} : \mathcal{X} \to \{1, \ldots, d\}^n$ for this channel, that achieves rate R and for each x fails with probability less than ϵ. Now when $X = x$ any player that is not leaking will send a message chosen uniformly at random from $\{1, \ldots, d\}$ and any player PLR_i with $L_i = 1$ chooses a message uniformly at random from $\{1+(j-1)a, 2+(j-2)a, \ldots, ja\} \mod d$, where $j = \mathfrak{C}(x)_i$ is the i'th letter in the codeword for x. This ensures that Frank will be able to guess x with probability at least $1-\epsilon$. We see that given X the random variable (A_i, L_i), is independent from $A_1, L_1, \ldots, A_{i-1}, L_{i-1}, A_{i+1}, L_{i+1}, \ldots, A_n, L_n$. Using the computation from (12) we now get that $\Pr(L_i = 1 | T = t, X = x)$ is either 0 or $c' \leq c$ as needed. □

Corollary 8. *The safe* c*-capacity for* Indep_b *is* $\frac{-b\log(1-c)+c\log(1-b)}{c}$.

Proof. Follows from Proposition 6 and Theorem 7. □

Corollary 8 shows that if you want information about something that some proportion b of the population knows, but no one wants other people to think that they know it with probability greater than c, you can still get information about the subject, and at a rate of $\frac{-b\log(1-c)+c\log(1-b)}{c}$ bits per person you ask. What if only l people in the world have the information? They are allowed to blend into a group of any size n, and observers will think that any person in the larger group is as likely as anyone else to have the information. Only the number of people with the information is known to everyone.

If they are part of a group of $n \to \infty$ people, then each person in the larger group would have the information with probability $b = \frac{l}{n}$. If we forget that exactly l people know the information, and instead assumed that all the L_is were independent with $\Pr(L_i = 1) = b$ they would be able to leak

$$\frac{-b\log(1-c)+c\log(1-b)}{c}n = \frac{-\frac{l}{n}\log(1-c)+c\log\left(1-\frac{l}{n}\right)}{c}n \to \left(\frac{\log(1-c)}{c} - \log(e)\right)l$$

bits of information, where e is the base of the natural logarithm. We will see that even in the case where the number of leakers is known and constant, we can still get this rate. First we define the distribution of $(L_1, \ldots, L_n)$ that we get in this case.

Definition 6. Let Fixed(l, n) be the random variable $(L_1, \ldots, L_n)$ that is distributed such that the set of leakers $\{\text{PLR}_i | L_i = 1\}$ is uniformly distributed over all subsets of $\{\text{PLR}_1, \ldots, \text{PLR}_n\}$ of size l.

A rate R is *safely/riskily c-achievable for* Fixed if for all $\epsilon > 0$ and all l_0, there exists a safe/risky $(n, lR, \text{Fixed}(l, n), c, \epsilon)$-protocol for some $l \geq l_0$ and some n.

The *safe/risky c-capacity for* Fixed is the supremum of all safely/riskily c-achievable rates for Fixed.

Notice that in this definition, the rate is measured in bits per leaker rather than bits per person communicating. That is because in this setup we assume that the number of people with the information is the bounded resource, and that they can find an arbitrarily large group of person to hide in.

Again, it turns out that the safe and the risky c-capacity for Fixed are actually the same, but for the proofs it will be convenient to have both definitions.

Proposition 9. *No rate* $R > \frac{-\log(1-c)}{c} - \log(e)$*, where e is the base of the natural logarithm is safely c-achievable for* Fixed.

Proof. This proof is very similar to the proof of Proposition 6.

Assume for contradiction that $R > \frac{-\log(1-c)}{c} - \log(e)$ is safely c-achievable. Consider a safe $(n, lR, \text{Fixed}(l, n), c, \epsilon)$-protocol π. We know from Theorem 5 that

$$I(X;T) \leq \frac{-\frac{l}{n}\log(1-c) + c\log\left(1-\frac{l}{n}\right)}{c} n \leq l\left(\frac{-\log(1-c)}{c} - \log(e)\right).$$

Here the second inequality follows from $\ln(1+x) \leq x$ or equivalently $\log(1+x) \leq \frac{x}{\ln(2)} = -x\log(e)$. Let $\delta := R - \left(\frac{-\log(1-c)}{c} - \log(e)\right)$. Now

$$H(X|T) = H(X) - I(X;T) \geq l\left(R - \left(\frac{-\log(1-c)}{c} - \log(e)\right)\right) = l\delta.$$

By Fano's inequality we get that the probability of error, $P_e = \Pr(D(t) \neq x)$ averages over all possible values of x is

$$P_e \geq \frac{l\delta - 1}{lR}.$$

Thus, if we choose l_0 sufficiently large and ϵ sufficiently small, we cannot have $l \geq l_0$ and $P_e \leq \epsilon$, so that there must be some value x where the probability of error $\Pr(D(T) \neq x|X = x)$ is greater than ϵ. □

Theorem 10. *Any rate* $R < \frac{-\log(1-c)}{c} - \log(e)$ *is riskily c-achievable for* Fixed.

There are two reasons that the proof of a lower bound for Indep_b given in Theorem 7 does not translate directly to a lower bound for Fixed. First, in the protocol given in the proof of Theorem 7, there is a very small risk that only the leakers send messages consistent with being leakers. This is fine when the L_i's are independent, but when the total number of leakers is known, this would completely reveal who the leakers are. This is why Theorem 10 is about risky achievability rather than safe achievability. The second problem is that the different usages of the channel are no longer independent as the number of leakers is constant. Intuitively, this should not be a problem, it should only make the channel more reliable. However, to show that this works, we would have to go through the proof of Shannon's noisy-channel coding theorem, and show that it still works. Instead, we will give a shorter but less natural proof, where we divide the players onto two groups and use Theorem 7 on each group.

Proof. Let $R < \frac{-\log(1-c)}{c} - \log(e)$, then we can find rational $b > 0$ and rational $c' < c$ and a $\delta > 0$ such that $R + \delta < \frac{-b\log(1-c') + c'\log(1-b)}{bc'}$, and let $n_0, \epsilon > 0$ be given. By Theorem 7 for any $\epsilon' > 0$ and any n_0' there exists a safe $(n, n(R+\delta), \text{Indep}_b(n), c', \epsilon')$-protocol where $n > n_0'$. Take such a protocol, where $\epsilon' > 0$ is sufficiently small and n_0' is sufficiently large. As b, and hence the denominator of b, is fixed and n can be sufficiently large, we can increase n a little to ensure that bn is an integer, while still keeping the rate at at least R. Thus we can assume that we have a $(n, nR, \text{Indep}_b(n), c', \epsilon')$-protocol, where $l := bn$ is an integer.

Now we will use this to make a risky $(2n, 2\lceil nR\rceil, \text{Fixed}(2nb), c, \epsilon)$-protocol. For such a protocol, X should be uniformly distributed on $\{1, \ldots, 2^{2\lceil nR\rceil}\}$, but instead we

can also think of X as a tuple (X_1, X_2) where the X_i are independent and each X_i is uniformly distributed on $\{1, \ldots, 2^{\lceil nR \rceil}\}$. Now we split the $2n$ players into two groups of n, and let the first group use the protocol from the proof of Theorem 7 to leak X_1, and the second group use the same protocol to leak X_2. We let Franks' guess of the value of X_1 be a function D_1 depending only of the transcript of the communication of the first group, and his guess of X_2 be a function D_2 depending only on the transcript of the second group. These functions are the same as D in the proof of Theorem 7. The total number of leakers is $2nb$, but the number of leakers in each half varies. Let S_{Indep} denote random variable that gives the number of leakers among n people, when each is leaking with probability b, independently of each other. So S_{Indep} is binomially distributed, $S_{\text{Indep}} \sim \mathrm{B}(n, b)$. Let $S_{\text{Fixed},1}$ denote the number of leakers in the first group as chosen above. Now we have.

Lemma 11. *For each k,*

$$\frac{\Pr(S_{\text{Fixed},1} = k)}{\Pr(S_{\text{Indep}} = k)} \leq 2.$$

Proof. We have

$$\Pr(S_{\text{Fixed},1} = k) = \frac{\binom{2l}{k}\binom{2n-2l}{n-k}}{\binom{2n}{n}}.$$

A simple computation shows

$$\frac{\Pr(S_{\text{Fixed},1} = k)\Pr(S_{\text{Indep}} = k+1)}{\Pr(S_{\text{Indep}} = k)\Pr(S_{\text{Fixed},1} = k+1)} = \frac{n-2l+k+1}{2l-k}\frac{l}{n-l},$$

which is > 1 for $k \geq l$ and < 1 for $k < l$. Thus, for fixed n and l the ratio $\frac{\Pr(S_{\text{Fixed},1}=k)}{\Pr(S_{\text{Indep}}=k)}$ is maximized by $k = l$. Using Stirling's formula,

$$1 \leq \frac{n!}{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n} \leq \frac{e}{\sqrt{2\pi}}$$

we get

$$\begin{aligned}\frac{\Pr(S_{\text{Fixed},1} = l)}{\Pr(S_{\text{Indep}} = l)} &= \frac{\binom{2l}{l}\binom{2n-2l}{n-l}n^n}{\binom{2n}{n}\binom{n}{l}l^l(n-l)^{n-l}} \\ &\leq \sqrt{2}\left(\frac{e}{\sqrt{2\pi}}\right)^3 \\ &< 2.\end{aligned}$$

□

Given that $S_{\text{Indep}} = k = S_{\text{Fixed},1}$, the distribution on $(L_1, \ldots, L_n)$ and transcript is the same in the protocol for Indep_b as it is for the first group in the above protocol. As Franks' guessing function is the same in the two cases, the probability of error given $S_{\text{Indep}} = k = S_{\text{Fixed},1}$ is the same in the two protocols. Let E_k denote the probability of error in the protocol for Indep_b given $S_{\text{Indep}} = k$, and let $E_{\text{Fixed},1}$ denote the probability that Franks' guess of X_1 is wrong.

$$\begin{aligned} E_{\text{Fixed},1} &= \sum_{k=1}^{n} \Pr(S_{\text{Fixed},1} = k) E_k \\ &\leq \sum_{k=1}^{n} 2 \Pr(S_{\text{Indep}} = k) E_k \\ &\leq 2\epsilon'. \end{aligned}$$

By the same argument, the probability that Frank guesses X_2 wrong is at most $2\epsilon'$, so the probability that he guesses $X = (X_1, X_2)$ is at most $4\epsilon'$. By choosing a sufficiently low ϵ' this is less than $\epsilon/2$

To compute the posterior probability $\Pr(L_i = 1 | X = x, T = t)$ that PLR_i was leaking, we have to take the entire transcript from both groups into account. Given T and X, let K denote the set of players who sent a message consistent with knowing X, and let $|K|$ denote the cardinality of K. Let S be the set of the $2l$ leaking players, and let s be a set of $2l$ players. Now

$$\Pr(S = s | X = x, T = t) = \frac{\Pr(T = t | S = s, X = x) \Pr(S = s | X = x)}{P(T = t | X = x)}.$$

This is 0 if s contains players who send a message not consistent with having the information, and is constant for all other s. Thus, any two players who send a message consistent with having the information, are equally likely to have known X given T and X, so they will have $\Pr(L_i = 1 | T = t, X = x) = \frac{2l}{|K|}$. So to ensure that $\Pr(L_i = 1 | T = t, X = x) \leq c$ with high probability (for each x and random t) we only need to ensure that with high probability, $|K| \geq \frac{2l}{c}$. We see that $|K| = 2l + \text{B}\left(2n - 2l, \frac{b(1-c')}{c'(1-b)}\right)$, which have expectation $2l + (2n-2l)\frac{b(1-c')}{c'(1-b)} = \frac{2l}{c'} = \frac{2l}{c} + 2l\frac{c-c'}{cc'}$. We also see that the variance is $(2n-2l)b(1-b)$, so for sufficiently high n (and thus l) Chebyshev's inequality, shows that $|K| \geq \frac{2l}{c}$ with probability at least $1 - \epsilon/2$. Thus, for sufficiently large n'_0 and sufficiently low ϵ', the resulting protocol is a risky $(2n, 2\lceil nR \rceil, \text{Fixed}(2nb), c, \epsilon)$-protocol. □

4.1. *General $\mathfrak{L}$-Structures*

We have shown that the safe c-capacity for Fixed is at most $\frac{-\log(1-c)}{c} - \log(e)$ which is at most the risky c-capacity Fixed. To finish the proof that they are both $\frac{-\log(1-c)}{c} - \log(e)$, we only need to show that the safe capacity is not smaller than the risky. Notice that the corresponding claim is not true if we are only interested in the mutual information between X and transcript T. Here we could construct a collaborating cryptography

protocol where, with probability $1 - 10^{-100}$, we have $\Pr(L_i = 1|T = t) < b + 10^{-100}$ and yet $I(X;T) \geq 10^{100}$. To do this we need to take X to have extremely high entropy, and with a probability 10^{-100} a leaking player will send X in a message, and otherwise just send some message. On the other hand, if we require that $\Pr(L_i = 1|T = t) < b + 10^{-100}$ holds for all transcripts, then $I(X;T)$ have to be small compared to total number of players. The point of this section is to show that you cannot do something similar for reliable leakage. We will prove this in a setting that generalises Indep_b and Fixed. Remember that the difference between Indep_b and Fixed capacity is not only in the distributions on $(L_1, \ldots, L_n)$, but also in what we are trying to minimise the use of. In Indep_b we want to have as few people communicating as possible, while in Fixed we only care about the number of people who are leaking. Our general definition has to capture this difference as well.

Definition 7. An $\mathfrak{L}$-*structure* $(\mathfrak{L}, C)$ is a set $\mathfrak{L}$ of joint distributions of $(L_1, \ldots, L_n)$ (where n does not need to be the same for each element), where each L_i is distributed on $\{0, 1\}$, together with a *cost function* $C : \mathfrak{L} \to \mathbb{R}_{\geq 0}$.

Indep_b is the $\mathfrak{L}$-structure $(\mathfrak{L}_{\text{Indep}_b}, C_{\#})$, where $\mathfrak{L}_{\text{Indep}_b}$ is the set of distributions on $(L_1, \ldots, L_n)$ (over $n \in \mathbb{N}$) where for all i, $\Pr(L_i = 1) = b$ and the L_i are independent, and $C_{\#}$ is the function that sends a distribution on $(L_1, \ldots, L_n)$ to n.

Fixed is the $\mathfrak{L}$-structure $(\mathfrak{L}_{\text{Fixed}}, C_{\text{Fixed}})$ of distributions on $(L_1, \ldots, L_n)$ such that for some number l the set $\{i|L_i = 1\}$ is uniformly distributed over all subsets of $\{1, \ldots, n\}$ of size l, and C_{Fixed} sends such a distribution on $(L_1, \ldots, L_n)$ to this number l.

For an $\mathfrak{L}$-structure $(\mathfrak{L}, C)$ a rate R is safely/riskily c-achievable for $(\mathfrak{L}, C)$ if for all $\epsilon > 0$ and all $h_0 \geq 0$ there exists a safe/riskily (n, h, L, c, ϵ)-protocol with $h \geq h_0$, $h \geq C(L)R$ and $L \in \mathfrak{L}$.

The safe/risky c-capacity for $(\mathfrak{L}, C)$ is the supremum of all safely/riskily c-achievable rates for $(\mathfrak{L}, C)$.

We see that Definition 7 agrees with Definitions 5 and 6, and is much more general.

Proposition 12. *Let $(\mathfrak{L}, C)$ be an $\mathfrak{L}$-structure. The safe c-capacity for $(\mathfrak{L}, C)$ and the risky c-capacity for $(\mathfrak{L}, C)$ are non-decreasing functions of c.*

Proof. Let $c' > c$. Any safe/risky (n, h, L, c, ϵ)-protocol is a safe/risky (n, h, L, c', ϵ)-protocol, so any safe/riskily c-achievable rate for $(\mathfrak{L}, C)$ is a safe/riskily c'-achievable rate for $(\mathfrak{L}, C)$. □

Proposition 13. *Let $(\mathfrak{L}, C)$ be an $\mathfrak{L}$-structure. The safe c-capacity for $(\mathfrak{L}, C)$ is at most the risky c-capacity for $(\mathfrak{L}, C)$.*

Proof. Any safe (n, h, L, c, ϵ)-protocol is a risky (n, h, L, c, ϵ)-protocol, so any safely c-achievable rate for $(\mathfrak{L}, C)$ is riskily c-achievable for $(\mathfrak{L}, C)$. □

The opposite inequality almost holds. Before we show that, we need a lemma.

Lemma 14. *For any risky (n, h, L, c, ϵ)-protocol π, there is a risky (n, h, L, c, ϵ)-protocol π' where each message is either 0 or 1, and given previous transcript and given that the person sending the message is not leaking, there is at least probability $1/3$ of the message being 0 and at least $1/3$ of it being 1.*

Proof. To restrict to $\{0, 1\}$ we simply send one bit at a time, so now we only have to ensure that the probability of a message sent by a non-leaker being 0 is always in $[\frac{1}{3}, \frac{2}{3}]$. If the next message is 0 with probability $p < 1/3$, given that the sender is not leaking we modify the protocol (the case where $p > 2/3$ is similar). First, the player PLR_i sending the message decides if she would have sent 0 or 1 in the old protocol π. Call this message a. If $a = 0$ she chooses a number in the interval $(0, p)$ uniformly at random, if $a = 1$ she chooses a number in $(p, 1)$ uniformly at random. She then sends the bits of the number one bit at a time until

- She sends "1", or
- Given transcript until now, there is probability at least $\frac{1}{3}$ that $a = 0$

In the first case we know that $a = 1$, and we can go to the next round of π. Each time PLR_i says 0, she doubles the probability that $a = 0$, so if we are in the second case (and was not before the last message), $\Pr(a = 0|T) < \frac{2}{3}$. In this case she will simply reveal a in the next message.

Instead of choosing a real number uniformly from $(0, p)$ or $(p, 1)$, which would require access to randomness with infinite entropy, PLR_i can just in each step compute the probabilities of sending 0 or 1 given that she had chosen such a number. Thus, if for every probability p' every player has access to a coin that ends head up with probability p', they only need a finite number of coin flips to follow the above protocol. □

The following lemma says that for any risky protocol, you can always find a safe protocol that that achieves the same rate: you just need to increase the threshold c by an arbitrarily small amount.

Lemma 15. *Let $c' > c$. The safe c'-capacity for $(\mathfrak{L}, C)$ is at least the same as the risky c-capacity for $(\mathfrak{L}, C)$.*

Proof. To show this, it is enough to show that if R is a riskily c-achievable rate for $(\mathfrak{L}, C)$, then R is safely c'-achievable for $(\mathfrak{L}, C)$. Let R be a riskily c-achievable rate for $(\mathfrak{L}, C)$, and let $\epsilon' > 0$ and h'_0 be given. We want to show that there exists a safe $(n', h', L, c', \epsilon')$-protocol with $h' \geq h'_0$, $L \in \mathfrak{L}$ and $h' \geq C(L)R$.

As R is riskily c-achievable for $(\mathfrak{L}, C)$, there exists a risky (n, h, L, c, ϵ)-protocol for any $\epsilon > 0$ and some $L \in \mathfrak{L}, h \geq h'_0, h \geq C(L)R$ and n. Let π be such a protocol, where ϵ is a small number to be specified later.

We want to modify π to make it a safe protocol π'. First, by Lemma 14 we can assume that all messages send in π are in $\{0, 1\}$ and given that the sender is not leaking, it has probability at least $1/3$ of being 0 and at least probability $1/3$ of being 1.

To ensure that for no transcript t and player PLR_i we have $\Pr(L_i = 1|X = x, T = t) > c'$, we modify the protocol, such that everyone starts to pretend ignorance if the next message could result in $\Pr(L_i = 1|X = x, T^{k+1} = t^{k+1}) > c'$. Formally, we

define a protocol π' that starts of as π but if at some point the transcript is t^k and for some i and $b \in \{0, 1\}$ we have $\Pr(L_i = 1 | T^{k+1} = t^k \circ b, X = x) > c'$ all the players *pretends ignorance*, that is for the rest of the protocol they send messages as if they did not have the information and were following π. Notice that only the players who knows the information x can decide if they should pretend ignorance, but this is not a problem as the players who do not have the information, are already sending messages as if they did not have the information.

First we want to show that π' is c'-safe. As long as they do not pretend ignorance we know that $\Pr(L_i = 1 | T^k = t^k, X = x) \leq c'$ for the partial transcript t^k and all i. If at some point they start to pretend ignorance, we have $\Pr(L_i = 1 | T^k = t^k, X = x) \leq c'$ before they start, and all messages will be chosen as if no one had the information. Eve, who knows X, can compute $\Pr(L_i = 1 | T^{k+1} = t^k \circ b, X = x) > c'$ for each i and b, so she knows if everyone is pretending ignorance. Thus, Eve does not learn anything about L from listening to the rest of the communication, so we will still have $\Pr(L_i = 1 | T = t, X = x) \leq c'$ when π' terminates.

Fix $x \in \mathcal{X}$. We want to compute the probability that they pretend ignorance given $X = x$. Let $E_{par,>c'}$ denote the event that for transcript T from the execution of π, we can find some k and some i such that we have $\Pr(L_i = 1 | T^k = t^k, X = x) > c'$. That is, at some point in the execution of π, an observer would say that PLR_i was leaking with probability greater than c'. Let $E_{tot,>c}$ be that event that for the total transcript there is some i such that $\Pr(L_i = 1 | T = t, X = x) > c$. For each transcript t where $\Pr(L_i = 1 | T^k = t^k, X = x) > c'$ for some k, i, we consider that smallest k such that $\Pr(L_i = 1 | T^k = t^k, X = x) > c'$ happens for some i. For this fixed t^k let T^{-k} denote the random variable that is distributed as the rest of the transcript given that the transcript starts with t^k and $X = x$. Let S_{t^k} denote the random variable

$$S_{t^k} = \Pr\left(L_i = 1 | T = t^k \circ T^{-k}, X = x\right).$$

That is, S_{t^k} is a function of T^{-k}. We see that S_{t^k} takes values in $[0, 1]$ and $\mathbb{E}S_{t^k} = \Pr(L_i = 1 | T^k = t^k, X = x) > c'$ so by Markov's inequality on $1 - S_{t^k}$ we get

$$\Pr(1 - S_{t^k} \geq 1 - c - \epsilon_1 | X = x) \leq \frac{\mathbb{E}(1 - S_{t^k})}{1 - c - \epsilon_1} < \frac{1 - c'}{1 - c - \epsilon_1}$$

for all $\epsilon_1 > 0$. Thus, given that $E_{par,>c'}$ happens, $E_{tot,>c}$ will happen with probability at least $\frac{c'-c}{1-c} > 0$. So $\frac{c'-c}{1-c} \Pr(E_{par,>c'} | X = x) \leq \Pr(E_{tot,>c} | X = x) \leq \epsilon$, where the last inequality follows from the assumption about π.

Let E_{ig} be the event that in the evaluation of π' the players pretends ignorance. The players only pretends ignorance if they are one message away from making $E_{par,>c'}$ happen. We assumed that in π each possible message get sent with probability at least $1/3$ if the sender is not leaking. As there is probability at least $1 - c'$ that he is not leaking, each possible message gets sent with probability at least $\frac{1-c'}{3}$ so $\frac{1-c'}{3} \Pr(E_{ig} | X = x) \leq \Pr(E_{par,>c'} | X = x)$. Thus,

$$\Pr(E_{ig}|X=x) \leq \frac{3}{1-c'} \Pr(E_{par,>c'}|X=x) \leq 3\epsilon \frac{(1-c)}{(c'-c)(1-c')}.$$

Let T' denote the random variable you get from running π' and T the random variable you get from running π, with a joint distribution in such a way that $(X, L, T) = (X, L, T')$ unless the players pretends ignorance. We need to show that there is a decoding function D' from the set of complete transcripts to possible values of X such that for each x, $\Pr(D'(T') = x|X = x) \geq 1 - \epsilon'$. From the assumptions about π we know that there is a function D from the set of possible transcripts to the support of X such that for each x, $\Pr(D(T) = x|X = x) \geq 1 - \epsilon$. We know that in π' and for fixed x, the players only pretends ignorance with probability at most $\frac{3\epsilon(1-c)}{(c'-c)(1-c')}$, so by setting $D' = D$ we get $\Pr(D'(T') = x|X = x) \geq 1 - \epsilon - \frac{3\epsilon(1-c)}{(c'-c)(1-c')}$. For sufficiently small ϵ (depending only on ϵ', c and c') this is less than ϵ' and we are done. □

If we add a continuity assumption, we get that the safe and the risky c capacity are the same.

Corollary 16. *Let $(\mathfrak{L}, C)$ be a $\mathcal{L}$-structure. If the safe c-capacity for $(\mathfrak{L}, C)$ as a function of c is right-continuous at c_0, or if the risky c-capacity for $(\mathfrak{L}, C)$ as a function of c is left-continuous at c_0 then the safe c_0-capacity for $(\mathfrak{L}, C)$ and the risky c_0-capacity for $(\mathfrak{L}, C)$ are the same.*

Proof. Assume that the safe c-capacity for $(\mathfrak{L}, C)$ as a function of c a right-continuous at c_0. Then Lemma 15 shows that the risky c-capacity for $(\mathfrak{L}, C)$ is at most the safe c'-capacity for $(\mathfrak{L}, C)$ for all $c' > c$. By continuity assumption, this gives us that the risky c-capacity for $(\mathfrak{L}, C)$ is at most the safe c-capacity for $(\mathfrak{L}, C)$. Proposition 13 shows the opposite inequality. The proof of the second part of the corollary is similar. □

Corollary 17. *Let $(\mathfrak{L}, C)$ be a $\mathcal{L}$-structure. The safe c-capacity for $(\mathfrak{L}, C)$ and the risky c-capacity for $(\mathfrak{L}, C)$ are the same for all but at most countably many values $c \in (0, 1)$.*

Proof. By Proposition 12, the safe c-capacity for $(\mathfrak{L}, C)$ is a monotone function, so it is continuous in all but countably many points. Now 16 implies that it is the same as the risky c-capacity for $(\mathfrak{L}, C)$ in all but countably many points. □

As promised, we can now show that for Indep_b the safe and the risky c-capacities are the same.

Corollary 18. *The safe c-capacity for Indep_b and the risky c-capacity for Indep_b are the same for all $c \in (0, 1)$.*

Proof. We know from Corollary 8 that the safe c-capacity for Indep_b is a continuous function of c. Now Corollary 16 implies that it is the same as the risky c-capacity for Indep_b. □

Corollary 19. *Let $c \in (0, 1)$. The safe c-capacity for* Fixed *and the risky c-capacity for* Fixed *are both* $\frac{-\log(1-c)}{c} - \log(e)$.

Proof. We know from Proposition 9 that the safe c-capacity for Fixed is at most $\frac{-\log(1-c)}{c} - \log(e)$, we know from Theorem 10 that the risky c' fixed capacity is at least $\frac{-\log(1-c)}{c} - \log(e)$, and from Corollary 17 that they are the same except on at most countably many values. Thus, they must both be $\frac{-\log(1-c)}{c} - \log(e)$ on all but countably many values. We know from 12 that both are monotone, so they must both be $\frac{-\log(1-c)}{c} - \log(e)$ without exceptions. □

5. Adaptive Cryptogenographic Protocols

Until now we have assumed that each player is either leaker or not a leaker. In this section we study some adaptive models where people start as non-leakers, but might start to leak at some point. Once a person is a leaker, that person will always be a leaker.

An *adaptive cryptogenography protocol* π is defined as follows: for each partial transcript t^k, each vector $L_{\cdot,k-1} = (L_{1,k-1}, \ldots, L_{n,k-1})$ describing the set of leaker when the k'th message was sent, and each secret x, π gives a probability distribution over vectors $L_{\cdot,k} \geq L_{\cdot,k-1}$ describing the set of leakers after the k'th message. Furthermore, like a collaborative cryptogenography protocol, π specifies for each partial transcript t^k

- Should the communication stop or continue, and if it should continue,
- Who is next to send a message, say PLR_i, and
- A distribution $p_?$ and a set of distributions, $\{p_x\}_{x\in\mathcal{X}}$ (the distributions $p_?$ and $\{p_x\}_{x\in\mathcal{X}}$ depend on π and t^k). Now PLR_i should choose a message using $p_?$, if $L_{i,k} = 0$ and choose a message using p_x if $L_{i,k} = 1$ and $X = x$.

Here it is natural to put some restriction on how many leakers there can be, and on what can influence whether a person becomes a leaker. We suggest two ways of putting a limitation on the total number of leakers, and three different rules for what can affect the probability that a person becomes a leaker, giving a total of six different combinations. In this section we will find the capacities for two of them.

The two ways of restricting the total number of leakers are called "b-threshold" and "b-dormant". The *b-threshold* restriction requires that the expected number of leakers at the end of the protocol is at most bn. This is a slightly unnatural requirement, but is the easiest to analyse. A more natural requirement is the *b-dormant* restriction, which say that at the beginning each player is chosen to be a "dormant" leaker with probability b, and only dormant leakers can become leakers. We can think of dormant leakers as people with the personality or the capacity to become leakers. Clearly, the b-dormant model is more restrictive than the b-threshold model, but on the other hand, the leakers can do more in the b-dormant model than in Indep_b in the static model: If you take $b = c$, the c-capacity for Indep_c is 0, but in the b-dormant model you can leak information, for example by letting each dormant leaker become leaker with probability $1/2$, and use a protocol for $\text{Indep}_{c/2}$.

The three models for how a player become a leaker are called "centrally organised", "informed choice" and "uninformed choice". In the *centrally organised* model we assume that there is someone organising which players become leakers. We assume this person have all the relevant information, t^k, $L_{\cdot,k-1}$ and x, and hence there is no restriction on the distribution of $L_{\cdot,k}$ except $L_{\cdot,k} \geq L_{\cdot,k-1}$, that is leakers cannot turn into non-leakers. In the *informed choice* we assume that even the non-leakers know x, and they may use this when deciding whether to become a leaker, but each player makes the decision on whether to become a leaker on her own. That is, the distribution of $L_{i,k}$ depends on $L_{i,k-1}$, x and t^k but given x and t^k it is independent from all the other $L_{j,k}$'s and $L_{j,k-1}$'s. Finally, there is the *uninformed choice* model where the players only learn x when they decide to become leakers. Here $L_{i,k}$ only depends on $L_{i,k-1}$ and t^k.

These give six different models that we call *adaptive model*. We use M_b to denote an adaptive model with parameter b. While "b-dormant informed choice" and "b-dormant uninformed choice" are probably the most realistic models, "b-threshold centrally organised" and "b-threshold informed choice" seem to be the easiest to analyse.

Definition 8. We let $\text{susp}_{i,k}$ denote the suspicion that $L_{i,k} = 1$, e.g.

$$\text{susp}_{i,k}\left(X, T^k\right) = -\sum_{x,t^k} \Pr\left(X = x, T^k = t^k\right) \log\left(\Pr\left(L_{i,k} = 0 | X = x, T^k, t^k\right)\right).$$

We define $L_i = L_{i,\text{length}(\pi)}$ and similarly $\text{susp}_i = \text{susp}_{i,\text{length}(\pi)}$.

Definition 9. A *risky* (n, h, M_b, c, ϵ)*-protocol* is an adaptive cryptogenography protocol satisfying the requirements of model M_b together with a function D from the set of possible transcripts to $\mathcal{X} = \{1, \ldots, 2^{\lceil h \rceil}\}$ such that when X is uniformly distributed on $\mathcal{X}$, then for any $x \in \mathcal{X}$, there is probability at least $1 - \epsilon$ that a random transcript t distributed as $T|_{X=x}$ satisfies

Reasonable doubt: $\forall i : \Pr(L_i = 1 | T = t, X = x) \leq c$, and
Reliable leakage: $D(t) = x$

A *safe* (n, h, M_b, c, ϵ)*-protocol* is a risky (n, h, M_b, c, ϵ)-protocol where $\Pr(L_i = 1 | T = t, X = x) \leq c$ for all i, t, x with $\Pr(T = t, X = x) > 0$.

A rate R is *safely/riskily c-achievable for* M_b if for all $\epsilon > 0$ and all n_0, there exists a safe/risky $(n, nR, M_b, c, \epsilon)$-protocol for some $n \geq n_0$.

The *safe/risky c-capacity for* M_b is the supremum of all safely/riskily c-achievable rates for M_b.

Theorem 20. *For $b \leq c$ and any model M_b the safe c-capacity for M_b is at most* $\frac{-b\log(1-c)}{c} - b\log(e)$.

Proof. As "b-threshold centrally organised" is the least restrictive model, we can assume that M_b is this model. Let π be an adaptive cryptogenography protocol for M_b. The function $\frac{-b\log(1-c)}{c} - b\log(e)$ is increasing in b, so we can assume that the expected number of leakers at the end of π is exactly bn, as the protocol would otherwise be an $M_{b'}$-protocol for some $b' < b$.

As when we proved Theorem 4 we can assume that the next player to send a message does not depend on the previous transcript t^{k-1}, but only on the number of messages sent. If PLR_j sends the k'th message Corollary 2 tells us that

$$I\left(X; T_k | T^{k-1}\right) \leq \text{susp}_{j,k-1}\left(X, T^k\right) - \text{susp}_{j,k-1}\left(X, T^{k-1}\right), \tag{13}$$

and Proposition 3 tells us that for $i \neq j$

$$\text{susp}_{i,k-1}\left(X, T^k\right) \geq \text{susp}_{i,k-1}\left(X, T^{k-1}\right). \tag{14}$$

If we move right hand side of (14) to the other side, add over all $i \neq j$ and add the result to (13) we get

$$I\left(X; T_k | T^{k-1}\right) \leq \sum_{i=1}^{n}\left(\text{susp}_{i,k-1}\left(X, T^k\right) - \text{susp}_{i,k-1}\left(X, T^{k-1}\right)\right) \tag{15}$$

In this adaptive model we also need to consider how it affects the suspicion that players can turn into leakers. By a small abuse of notation we let c_{i,k',x,t^k} denote $\Pr(L_{i,k'} = 1 | X = x, T^k = t^k)$, and $c_{i,k}$ denote $\Pr(L_{i,k} = 1)$. As $\frac{d}{dx}\log(x) \geq \log(e)$ for $x \leq 1$ and $L_{i,k} \geq L_{i,k-1}$ we have for all i and k,

$$\begin{aligned}
&\text{susp}_{i,k}\left(X, T^k\right) - \text{susp}_{i,k-1}\left(X, T^k\right) \\
&= -\sum_{x,t^k} \Pr\left(X = x, T^k = t^k\right)\left(\log\left(1 - c_{i,k,x,t^k}\right) - \log\left(1 - c_{i,k-1,x,t^k}\right)\right) \\
&\geq -\sum_{x,t^k} \Pr\left(X = x, T^k = t^k\right)\log(e)\left(\left(1 - c_{i,k,x,t^k}\right) - \left(1 - c_{i,k-1,x,t^k}\right)\right) \\
&= \sum_{x,t^k} \Pr\left(X = x, T^k = t^k\right)\log(e)\left(c_{i,k,x,t^k} - c_{i,k-1,x,t^k}\right) \\
&= \left(c_{i,k} - c_{i,k-1}\right)\log(e).
\end{aligned} \tag{16}$$

If we move right hand side of (16) to the other side, add over all i and add the result to (15) we get

$$\begin{aligned}
I\left(X; T_k | T^{k-1}\right) \leq \sum_{i=1}^{n}\Big(&\text{susp}_{i,k}\left(X, T^k\right) \\
&-\text{susp}_{i,k-1}\left(X, T^{k-1}\right) - \log(e)\left(c_{i,k} - c_{i,k-1}\right)\Big).
\end{aligned}$$

Summing this over all rounds gives us

$$I(X; T) \leq \sum_i \left(\text{susp}_i(X, T) - \text{susp}_{i,0}(X) - \log(e)\left(\Pr(L_i = 1) - \Pr(L_{i,0} = 1)\right)\right)$$

$$= \sum_i \left(\text{susp}_i(X, T) - \log(e)\Pr(L_i = 1)\right).$$

We have $\Pr(L_i = 1|X = x, T = t) \leq c$ so by the same argument as in Theorem 5 we have

$$\text{susp}_i(X = x, T = t) \leq \frac{-\log(1-c)}{c}\Pr(L_i = 1|X = x, T = t).$$

Now we make a computation very similar to the one in Theorem 5.

$$\begin{aligned}
\sum_i \text{susp}_i(X, T) &= \sum_{i,x,t} \Pr(X = x, T = t)\text{susp}_i(X = x, T = t) \\
&\leq \sum_{i,x,t} \Pr(X = x, T = t)\frac{-\log(1-c)}{c}\Pr(L_i = 1|X = x, T = t) \\
&= \sum_{i,x,t} \frac{-\log(1-c)}{c}\Pr(L_i = 1, X = x, T = t) \\
&= \sum_i \frac{-\log(1-c)}{c}\Pr(L_i = 1) \\
&\leq n\frac{-b\log(1-c)}{c}.
\end{aligned}$$

Here the last inequality follows from the assumption that $\mathbb{E}\sum_i L_i = nb$ By applying Fano's inequality as in the proof of Proposition 6, it follows that the safe c-capacity for M_b is at most $\frac{-b\log(1-c)}{c} - b\log(e)$. □

In the next two propositions we show that this upper bound also holds for risky protocols.

Proposition 21. *Let $c' > c$. The safe c'-capacity for "b-threshold centrally organised" is at least the same as the risky c-capacity for "b-threshold centrally organised".*

Proof. Let M_b be the model "b-threshold centrally organised". We use that same strategy as in the proof of Lemma 15. Assume that R is riskily c-achievable for M_b. To show the statement, it is enough to show that if R is then safely c'-achievable for M_b. Let $\epsilon' > 0$ and n'_0 be given. We need to show that there exists a safe $(n', Rn', M_b, c', \epsilon')$-protocol, where $n' \geq n'_0$. As R is riskily c-achievable for M_b, there exists a risky $(n, nR, M_b, c, \epsilon)$ protocol for any $\epsilon > 0$ and where $n \geq n'_0$. Let π be such a protocol for a small ϵ to be specified later.

We will modify π to get a safe protocol π'. By Lemma 14 we can assume that all messages send in π are in $\{0, 1\}$ and given that the sender is not leaking, it has probability at least $1/3$ of being 0 and at least probability $1/3$ of being 1.

As in the proof of Lemma 15, modify the π by forcing the players to *pretend ignorance* in some situations. Pretending ignorance means that a player sends messages as if he

was a non-leaker. Once a player starts to pretend ignorance, he will continue to do so for the rest of the protocol. Furthermore, once one leaker starts to pretend ignorance, we want all the players to pretend ignorance. To make this possible, we need to ensure that all the leakers can decide whether they should pretend ignorance, but the non-leakers does not have to know, as they are already sending messages as if they were non-leakers. If Eve is able to decide whether the players are pretending ignorance, it means that once the players start to pretend ignorance, she does not get any further information.

We require the players to pretend ignorance from round $k+1$ and onwards, if the current transcript is t^k and $\Pr(L_{i,k} = 1|T^k = t^k, X = x) \leq c'$ but $\Pr(L_{i,k} = 1|T^{k+1} = t^k \circ t_{k+1}, X = x) > c'$ for some player i and some $t_{k+1} \in \{0, 1\}$. The leakers can all compute $\Pr(L_{i,k} = 1|T^{k+1} = t^k \circ t_{k+1}, X = x)$, so the leakers know if they should start to pretend ignorance. Eve can also compute $\Pr(L_{i,k} = 1|T^{k+1} = t^k \circ t_{k+1}, X = x)$, so once the player pretend ignorance she does not learn any further information. We also modify π such that when the players start to pretend ignorance, no one turn into leakers. We can do this, because the model is centrally organised, so the probability of becoming a leaker can depend on X. Furthermore, we modify π such that if the partial transcript t^k satisfies $\Pr(L_{i,k-1} = 1|T^k = t^k, X = x) \leq c'$ but $\Pr(L_{i,k} = 1|T^k = t^k, X = x) > c'$ for some i, then no one becomes a leaker at round k or any later rounds, and everyone starts to pretend ignorance. By induction on k, these modifications ensure that $\Pr(L_i = 1|T = t, X = x) \leq c'$.

Next we need to define the function D' that takes transcripts of π' to guesses of the value X. This is simply defined to be the same as the function D for π. To show that π' is a $(n', Rn', M_b, c', \epsilon')$-protocol, we need to show that for each x the probability $\Pr(D'(T) \neq x|X = x)$ is at most ϵ'. We define $E_{par,>c'}$ to be the event that for transcript T from the execution of π, we can find some k and some i such that we have $\Pr(L_{i,k-1} = 1|T^k = t^k, X = x) > c'$ or $\Pr(L_{i,k} = 1|T^k = t^k, X = x) > c'$, $E_{tot,>c}$ to be that event that for the total transcript there is some i such that $\Pr(L_i = 1|T = t, X = x) > c$, and E_{ig} to be the event that the players start to pretend ignorance. The only situation where the players start to pretend ignorance are when there is a possible message t_{k+1} that would give $\Pr(L_{i,k-1} = 1|T^k = t^k, X = x) > c'$ (as in the proof of Lemma 15) or if we would otherwise had increase some player i's probability of being a leaker $\Pr(L_{i,k-1} = 1|T^k = t^k, X = x)$ to a probability greater than c'. In the first case there is still probability at least $\frac{1-c'}{3}$ that $E_{par,>c'}$ would have happened if the players did not pretend ignorance, and in the second case there is probability 1 that $E_{par,>c'}$ would have happened. So we still have

$$\Pr(E_{ig}|X = x) \leq \frac{3}{1-c'} \Pr(E_{par,>c'}|X = x).$$

All other computations and arguments are exactly as in the proof of Lemma 15. This gives us

$$\Pr(E_{ig}|X = x) \leq 3\epsilon \frac{(1-c)}{(c'-c)(1-c')}.$$

Now we get

$$\begin{aligned}\Pr(D'(T) \neq x | X = x) &\leq \Pr(D(T) \neq x | X = x) + \Pr(D'(T') \neq D(T) | X = x) \\ &\leq \epsilon + \Pr(E_{ig} | X = x) \\ &\leq \epsilon + 3\epsilon \frac{(1-c)}{(c'-c)(1-c')}.\end{aligned}$$

For sufficiently small ϵ, depending on ϵ', c and c', this is less than ϵ'. □

Proposition 22. *For any $b \leq c$ and any model M_b the risky c-capacity for M_b is at most $\frac{-b\log(1-c)}{c} - b\log(e)$.*

Proof. As "b-threshold centrally organised" is the most general model, we can assume that M_b is this model. By continuity of $c \mapsto \frac{-b\log(1-c)}{c} - b\log(e)$ the result follows from Theorem 20 and Proposition 21. □

Proposition 23. *For any $b \geq c$ and any adaptive model M_b the risky c-capacity is at most $\frac{-c\log(1-c)}{c} = \log(1-c)$.*

Proof. Let π be a risky (n, h, M_b, c, ϵ)-protocol. Then we must have $\Pr(L_i = 1) \leq c$, so it is also a risky $(n, h,$ "c − threshold centrally organised", $c, \epsilon)$-protocol. The Proposition now follows from Proposition 22. □

We now show that the upper bounds are tight in two of the models.

Proposition 24. *Let $b < c$. If M_b is "b-threshold centrally organised" or "b-threshold informed choice", the safe c-capacity for M_b is at least $\frac{-b\log(1-c)}{c} - b\log(e)$.*

Proof. As "b-threshold informed choice" is the most restrictive of the two models, we can assume that M_b is this model.

Let b and c be fixed, and choose any $\epsilon > 0$ and some large integer m. We will define a protocol π for the model M_b that works in m stages. At the beginning everyone are *available* and after each stage some players be *unavailable*, meaning that they will not send any more messages. Before each stage starts, everyone, even an observer who does not know X will be able to compute who should be available and who should be unavailable in that stage. Define $n' = \lfloor \frac{c-b}{2c} n \rfloor$, $b' = \frac{2bc}{(c-b)m}$ and $h = \left\lfloor \left(\frac{-b'\log(1-c) + c\log(1-b')}{c} - m^{-2} \right) n' \right\rfloor$ and let X be uniformly distributed over $\{1, \ldots, 2^h\}^m$.

If there is less than n' players available at the beginning of stage j, the protocol halts. Otherwise, each of the first n' players who are available, choose whether to become leaker independently with probability b'. Assuming that n is sufficiently big (given b, c, ϵ and m) then n' is sufficiently big and we get a safe $(n', h, \mathrm{Indep}_{b'}(n'), c, \epsilon/(2m))$-protocol from the proof of Theorem 7. We let the n' players follow this protocol to leak X_j. According to Definition 3 there is a function D of the communication, that with high probability guesses the value the leakers tried to leak. Let $\hat{X}_j$ be D of the communication

of the j'th stage. If $x_j \neq \hat{X}_j$ we let all the leakers pretend ignorance for the rest of the entire protocol, and the non-leakers stay non-leakers. At the end of the j'th stage some players will have $\Pr(L_{i,(j)} = 1 | T^{(j)} = t^{(j)}, X^j = \hat{X}^j) > 0$, where (j) denotes the round where stage j finishes. We let these players be unavailable for all the following stages, and all other available players stay available. In particular we see that all players who are leaking in a given stage, will be unavailable in all the following stages.

Eve can compute $\hat{X}$, so she can determine if the players are pretending ignorance. Hence, once they pretend ignorance, Eve will not get any further information, so we only need to prove that until they pretend ignorance, they have reasonable doubt.

Let j be the last stage in which player i sends a message where he did not pretend ignorance. As he did not pretend ignorance, we must have $x^{j-1} = \hat{X}^{j-1}$, and he must have been available in stage j, otherwise he would not have sent a message in that stage. That means that $\Pr(L_{i,(j-1)} = 1 | T^{(j-1)} = t^{(j-1)}, X^{j-1}) = x^{j-1}) = 0$ so Eve would know that he did not leak in earlier rounds. As he had probability b' of becoming a leaker at stage j and the players used a safe $(n', h, \mathrm{Indep}_{b'}(n'), c, \epsilon/(2m))$-protocol in round j, we must have $\Pr(L_{i,(j)} = 1 | T^{(j)} = t^{(j)}, X = x) \leq c$, and no further message will change this probability. Thus, the protocol ensures reasonable doubt.

Next we want to compute the rate for the protocol we have constructed. In the limit, when $n \to \infty$ much faster than $m \to \infty$ the rate is

$$
\begin{aligned}
&\lim_{m\to\infty}\lim_{n\to\infty}\frac{hm}{n} \\
&= \lim_{m\to\infty}\lim_{n\to\infty}\frac{\left\lfloor\left(\frac{-b'\log(1-c)+c\log(1-b')}{c} - m^{-2}\right)n'\right\rfloor m}{n} \\
&\geq \lim_{m\to\infty}\lim_{n\to\infty}\frac{\frac{-b'\log(1-c)+c\log(1-b')}{c}n'm - (m^{-2}n'+1)m}{n} \\
&\geq \lim_{m\to\infty}\lim_{n\to\infty}\frac{\frac{-\frac{2bc}{(c-b)m}\log(1-c)+c\log\left(1-\frac{2bc}{(c-b)m}\right)}{c}\left(\frac{c-b}{2c}n-1\right)m - \left(m^{-2}\frac{c-b}{2c}n+1\right)m}{n} \\
&= \lim_{m\to\infty}\frac{-\frac{2bc}{(c-b)m}\log(1-c)+c\log\left(1-\frac{2bc}{(c-b)m}\right)}{c}\left(\frac{c-b}{2c}\right)m - \left(m^{-2}\frac{c-b}{2c}\right)m \\
&= \lim_{m\to\infty}\frac{-b\log(1-c)+m\frac{c-b}{2}\log\left(1-\frac{2bc}{(c-b)m}\right)}{c} - \left(m^{-2}\frac{c-b}{2c}\right)m \\
&= \frac{-b\log(1-c)-bc\log(e)}{c},
\end{aligned}
$$

as we wanted.

Finally, we want to compute the error probability. We divide the errors into two types. A type one error is an error where either $\hat{X}_j \neq X_j$. A type two error is the case where the protocol halts because there are less than n' available players left.

By construction, the probability of getting a type one error in stage j is at most $\frac{\epsilon}{2m}$. From the proof of Theorem 7, we see that if the players never pretend ignorance, then the number of players who would become unavailable in stage j is binomially

distributed with parameters n' and $b' + (1 - b')\frac{a}{d}$ where a and d are parameters from that proof. For any $\delta > 0$ we can choose a and d such that $\frac{a}{d} \leq \frac{b'(1-c)}{c(1-b')} + \delta$. Then $b' + (1-b')\frac{a}{d} \leq \frac{b'}{c} + \delta$. Thus, the total number of players who would become unavailable if there were enough players and they never pretended ignorance would be binomially distributed with parameters mn' and $p \leq \frac{b'}{c} + \delta = \frac{2b}{(c-b)m} + \delta$ and thus have expectation

$$\begin{aligned} mn'p &\leq m\left\lfloor \frac{c-b}{2c}n \right\rfloor \left(\frac{2b}{(c-b)m} + \delta\right) \\ &\leq m\frac{c-b}{2c}n\left(\frac{2b}{(c-b)m} + \delta\right) \\ &\leq \frac{b}{c}n + \delta nm \end{aligned}$$

By choosing δ to be sufficiently small depending on m, and n sufficiently big Chebyshev's inequality shows that with probability greater than $1 - \frac{\epsilon}{2}$ the total number of players who become unavailable is at most $\frac{c+b}{2c}n \leq n - \lfloor \frac{c-b}{2c}n \rfloor$ and hence there will be n' available players left for the last stage. Thus, the probability of a type two error would be less than $\frac{\epsilon}{2}$ so the total probability of error is less than ϵ. □

Theorem 25. *If M_b is "b-threshold centrally organised" or "b-threshold informed choice" the c-capacity for M_b is* $\frac{-\min(b,c)\log(1-c)}{c} - \min(b, c)\log(e)$.

Proof. For $b < c$ is follows from Proposition 22, Proposition 24 and the fact that the risky capacity must be at least the same as the safe. For $b \geq c$ is follows from the case $b < c$, Proposition 23 and the fact that the c-capacity must be non-decreasing in c. □

For an illustration of this theorem, see Fig. 2.

Notice that while our upper bounds for the six models are the same and are the same for the safe and risky case, the c-capacities might be different between the models, and the safe c-capacity might even be different from the risky c-capacity from some models. Similarly, even though our upper bound for all the models does not depend on b as long as $b \geq c$, we conjecture that in the b-dormant models the c-capacity for M_c is less than the c-capacity for M_1.

6. The Original Cryptogenography Problem

Brody et al. [4] studied the following cryptogenographic problem. We flip a coin, and tell the result to one out of n people. The $n - 1$ other people do not know who got the information. Formally that means we take $L = (L_1, \ldots, L_n)$ to be the random variable that is uniformly distributed over all $\{0, 1\}$-vectors $(l_1, \ldots, l_n)$ containing exactly one 1 and take X to be uniformly distributed over $\{0, 1\}$ independently from L. We let the group of n people use any collaborating cryptogenography protocol, and afterwards we let Frank guess the result of the coin flip (his guess depends only on the transcript) and then let Eve guess who was leaking (her guess can depend on both transcript and Franks'

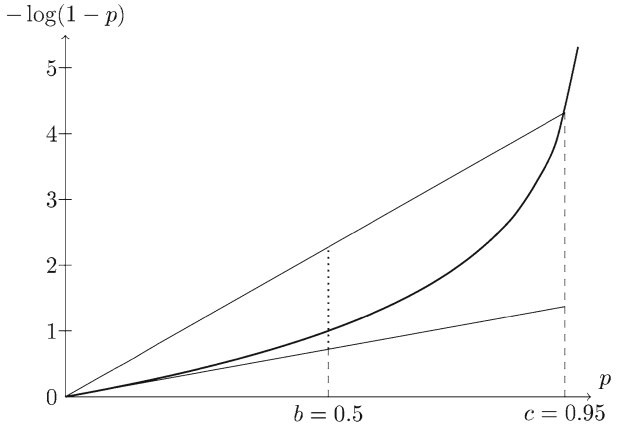

Fig. 2. This figure illustrates the advantage of the adaptive models "b-threshold centrally organised" and "b-threshold informed choice" compared to the non-adaptive model. Most of the figure is as Fig. 1. The new line is the tangent to $p \mapsto -\log(1-p)$ at $p = 0$. For $b \leq c$ the safe/risky c-capacity for M_b is given as the length of the dotted line. In the case, the advantage in using these two adaptive models over the static one, is given by the difference between the lower line and the curve. When $b > c$ in the static model, there is not even reasonable doubt from the beginning, and the capacity is $-\infty$. In these two adaptive models, the capacity is the same as for $b = c$.

guess). Eve wins if she guesses the leaker or if Frank does not guess the result of the coin flip. Otherwise, Frank and the n people communicating wins. We assume that both Frank and Eve make their guesses to maximise the probability that they win, rather than maximise the probability of being correct.[6]

For $n = 2$, Brody et al. [4] showed that the group would win with probability $\frac{1}{3}$ but no protocol can ensure winning with probability above $\frac{3}{8}$. Doerr and Künnemann [12] later improved this upper bound to 0.3672 and the lower bound to 0.3384. For general values of n, Brody et al. [4] showed that the probability that the group wins is always below $3/4$ and for sufficiently high n it is at least 0.5644. In this section we will generalise the problem to a situation where more people are leaking and X contains more information. It is obvious how to generalise X to more information, we simply take X to be uniformly distributed on $\{1, \ldots, 2^{\lceil h \rceil}\}$. It is less obvious how to generalise to more leakers. When more people are leaking, it would be unreasonable to require Eve to guess all the leakers. If this was the rule, one of the leaking players could just reveal himself as a leaker and say what X is, while the rest of the leakers behave exactly as the non-leakers. Instead, we let Eve guess one person and if that person is leaking, she wins.

[6] For example if $\Pr(L_1 = 1, X = 0 | T = t) = 0.97$, $\Pr(L_1 = 1, X = 1 | T = t) = 0.01$ and $\Pr(L_2 = 1, X = 1 | T = t) = 0.02$ then it is most likely that $X = 0$. However, Frank will guess that $X = 1$. If Frank instead guessed $X = 0$ then Eve would guess that PLR_1 is leaking and then Eve would be certain to win. Once Frank have guessed $X = 1$, Eve will guess that PLR_2 is leaking even though it is much more likely that PLR_1 is leaking. This is because, given that Frank is correct, it is more likely that PLR_2 is leaking, and Eve does not care if she guesses correct when Frank is wrong.

Definition 10. For fixed values of h, number of leakers l and number of communicating players $n > l$ and a collaborating cryptogenography protocol π, we let $\mathrm{Succ}(h, l, n, \pi)$ denote the probability that after the players communicate using protocol π, Frank will guess the correct value of X but Eve's guess will not be a leaker, assuming that Frank and Eve each guess using the strategy that maximise their own chance of winning. We define

$$\mathrm{Succ}(h, l, n) = \sup_{\pi}(\mathrm{Succ}(h, l, n, \pi)),$$

where the supremum is over all collaborating cryptogenography protocols π. Finally, we define

$$\mathrm{Succ}(h, l) = \lim_{n\to\infty} \mathrm{Succ}(h, l, n).$$

In this section we will investigate the asymptotic behaviour of $\mathrm{Succ}(h, l)$ when at least one of l and h tends to infinity. First some propositions.

Proposition 26. *The probability that the communicating players win the game does not change if Eve is told the value of X before they start to communicate.*

Proof. If Frank guesses the correct value of X, Eve was going to assume that that was the correct value anyway (as she wants to maximise the probability that she is correct given that Frank was correct), and if Frank guesses wrong, she would win anyway. □

In the rest of this section, we will assume that Eve knows the value of X.

Proposition 27. $\mathrm{Succ}(h, l, n)$ *and* $\mathrm{Succ}(h, l)$ *are non-increasing in* h.

Proof. Let $h > h'$ and let π be a protocol for parameters h, l, n and let the secret be denoted X. We construct a protocol π' with parameters h', l, n and secret denoted by X'. In the first round of π', PLR_1 announces $h - h'$ independent and uniformly chosen bits Y, and from then on, everyone follows protocol π for $X = X' \circ Y$. It is clear the $\mathrm{Succ}(h, l, n, \pi) \leq \mathrm{Succ}(h', l, n, \pi')$. □

Proposition 28. $\mathrm{Succ}(h, l, n)$ *is non-decreasing in* n.

Proof. We use the elimination strategy used in [4]. Let $n' > n$ and let π be a protocol for parameters h, l, n. We now construct a sequence of protocols π'_k for parameters h, l, n'. In the protocol π'_k each non-leaking player thinks of a uniformly chosen number in $\{1, \ldots, k\}$. First everyone who thought of the number 1 announces that and they are out, then everyone who thought of the number 2 and so on, until only n players a left. If two or more player thought of the same number, we might end up with less than n players left. In that case the leakers just announce themselves. If we are left with exactly n players, we know that the l leakers are still among them, and we have no further information about who they are. They then use protocol π, and win with probability

$\mathrm{Succ}(h, l, n)$. As $k \to \infty$, the probability that two players thought of the same number tends to 0, so $\mathrm{Succ}(h, l, n', \pi'_k) \to \mathrm{Succ}(h, l, n, \pi)$. □

Theorem 29. *For all* $p \in (0, 1)$,

$$\liminf_{l\to\infty} \mathrm{Succ}\left(\left\lceil\left(\frac{-\log(p)}{1-p} - \log(e)\right) l\right\rceil, l\right) \geq p.$$

Proof. We know from Corollary 19 that the safe c-capacity for Fixed is $\frac{-\log(1-c)}{c} - \log(e)$. If we let $\epsilon > 0$, and use this Corollary for $c = 1 - p + \epsilon/2$ we get that for sufficiently high l, n and $h = \left\lceil\left(\frac{-\log(p)}{1-p} - \log(e)\right) l\right\rceil$ there is a protocol π that will make Frank's probability of guessing wrong at most $\epsilon/2$, and seen from Eve's perspective, no one is leaking with probability greater than $1 - p + \epsilon/2$. By the union bound, the probability that Frank is wrong or Eve is correct[7] is at most $\epsilon/2 + 1 - p + \epsilon/2$, thus the communicating players win with probability at least $p - \epsilon$. □

In particular we have the following corollary.

Corollary 30. *Let* $l \to \infty$ *and* $h = h(l)$ *be a function of* l *with* $h = o(l)$. *Then* $\mathrm{Succ}(h, l) \to 1$.

Proof. Let $h(l) = o(l)$ be a function. For each l, we have $\mathrm{Succ}(h(l), l) \in [0, 1]$, so we only need to show that for any $\epsilon > 0$ there exists l_0 such that for all $l \geq l_0$ we have $\mathrm{Succ}(h(l), l) \geq 1 - 2\epsilon$. If we put $p = 1 - \epsilon$ in Theorem 29 we get

$$\liminf_{l\to\infty} \mathrm{Succ}\left(\left\lceil\left(\frac{-\log(1-\epsilon)}{\epsilon} - \log(e)\right) l\right\rceil, l\right) \geq 1 - \epsilon.$$

This means that there is some l_1 such that for all $l \geq l_1$ we have

$$\mathrm{Succ}\left(\left\lceil\left(\frac{-\log(1-\epsilon)}{\epsilon} - \log(e)\right) l\right\rceil, l\right) \geq 1 - 2\epsilon.$$

As $h(l) = o(l)$, there must be some l_2 such that $h(l) \leq \left(\frac{-\log(1-\epsilon)}{\epsilon} - \log(e)\right) l$ for all $l \geq l_2$. Now define $l_0 = \max(l_1, l_2)$. For all $l \geq l_0$ we have

$$\begin{aligned}\mathrm{Succ}(h(l), l) &\geq \mathrm{Succ}\left(\left\lceil\left(\frac{-\log(1-\epsilon)}{\epsilon} - \log(e)\right) l\right\rceil, l\right) \\ &\geq 1 - 2\epsilon.\end{aligned}$$

Here the first inequality uses Proposition 27 and $l \geq l_0 \geq l_2$ and the second inequality uses that $l \geq l_0 \geq l_1$. □

[7] Here we assume that Frank guesses on the most likely value of X, and we allow Eve to use any strategy. It could be that Frank could do better, but he is guaranteed at least this probability of winning.

Definition 11. Let the distribution of $(X, L_1, \ldots, L_n)$ be given and let π be a protocol with transcript T and π' a protocol with transcript T'. For a transcript t of π let μ_t denote the distribution $(X, L_1, \ldots, L_n)|_{T=t}$, and similar for transcripts t' of π'. We say that π and π' are *equivalent for* $(X, L_1, \ldots, L_n)$ (or just *equivalent* when it is clear what the distribution of $(X, L_1, \ldots, L_n)$ is) if the distribution of μ_T is the same as the distribution of $\mu_{T'}$.

Notice that for fixed t, μ_t is a distribution of $(X, L_1, \ldots, L_n)$, so μ_T is a random variable those values are themselves distributions over $(X, L_1, \ldots, L_n)$. For π and π' to be equivalent, we require the probability that the posterior distribution of $(X, L_1, \ldots, L_n)$ is μ to be the same for both π and π'. For two different distributions of $(X, L_1, \ldots, L_n)$ with the same support, π and π' are equivalent for one of them if and only if they are equivalent for the other distribution. Thus, when the support of $(X, L_1, \ldots, L_n)$ is clear, we can simply say equivalent.

Proposition 31. *If π and π' are equivalent collaborating cryptogenography protocols, then* $\mathrm{Succ}(h, l, n, \pi) = \mathrm{Succ}(h, l, n, \pi')$.

The next lemma show that we can ensure that before any player crosses probability c of having the bit, seen from Eve's perspective, that player lands on this probability.

Lemma 32. *Let π be any collaborating cryptogenography protocol, let $(X, L_1, \ldots, L_n)$ have any distribution and let $c \in (0, 1)$. Then there exists an equivalent collaborating cryptogenography protocol π' such that when we use it on $(X, L_1, \ldots, L_n)$ and let T' denote its transcript, it satisfies: For all $x \in \mathcal{X}$, all* PLR_i *and all non-empty partial transcripts t'^k, if*

$$\Pr(L_i = 1 | T'^k = t'^k, X = x) > c.$$

then there is a $k' < k$ such that

$$\Pr(L_i = 1 | T'^{k'} = t'^{k'}, X = x) = c$$

Proof. Let π, $(X, L_1, \ldots, L_n)$ and c be given, and assume that $(x, i) = (x_0, i_0)$ is a counterexample to the requirement from the lemma. We will then construct a protocol π' such that (x_0, i_0) is not a counterexample for π', and any (x, i) that satisfied the requirement for π also satisfies it for π'. By induction, this is enough to prove the lemma.

We can assume that the messages in π are sent one bit at a time. We say a partial transcript t^k is problematic if

$$\Pr(L_{i_0} = 1 | T^k = t^k, X = x_0) < c$$

but

$$\Pr(L_{i_0} = 1 | T^{k+1} = t^k \circ m, X = x_0) > c.$$

for some bit value m. Without loss of generality, assume that $m = 1$. Let $p = \Pr(T_{k+1} = 1 | T^k = t'^k)$.

We will use the c-notation from Sect. 3, so for example

$$c_{t^k,x_0} = \Pr(L_i = 1 | T^k = t^k, X = x_0).$$

Now

$$c > c_{t^k,x_0} = pc_{t^k \circ 1,x_0} + (1-p)c_{t^k \circ 0,x_0}$$

so $c_{t^k \circ 0,x_0} < c$. Let $q \in (p, 1)$ be the number such that

$$c = qc_{t^k \circ 1,x_0} + (1-q)c_{t^k \circ 0,x_0}.$$

Now we modify π. First, the player PLR_j, who is going to send to $k+1$'th message in π, decides if she would have sent 0 or 1 in π. If she would have sent 1 she sends the bits 11. If she would have sent 0 she sends 10 with probability $\frac{p(1-q)}{q(1-p)} \in (0, 1)$, and otherwise she sends 00. In all cases she sends the bits one at a time. They then continue the protocol π as if only the last of the two bits had been sent. If we let T' denote the transcript of the protocol with this modification, we get

$$c_{T'^{k+1}=t^k \circ 0,x_0} = c_{T^{k+1}=t^k \circ 0,x_0} < c$$

and

$$\begin{aligned} c_{T'^{k+1}=t^k \circ 1,x_0} &= \frac{pc_{T^{k+1}=t^k \circ 1,x_0} + (1-p)\frac{p(1-q)}{q(1-p)}c_{T^{k+1}=t^k \circ 0,x_0}}{p + (1-p)\frac{p(1-q)}{q(1-p)}} \\ &= qc_{T^{k+1}=t^k \circ 0,x_0} + (1-q)c_{T^{k+1}=t^k \circ 1,x_0} \\ &= c. \end{aligned}$$

So if PLR_j sends 11 or 10 in the modified protocol, we land on probability c. Let π' be the protocol we get from π by doing this modification for each problematic partial transcript t^k in π. It is clear that π and π' are equivalent, and that any (x, i) that satisfied the requirement before also does so afterwards. □

Lemma 33. *For any $c \in (0, 1)$ and any h, l, n, π, we have* $\text{Succ}(h, l, n, \pi) \leq 1 - \frac{ch + l\log(1-c) + lc\log(e) - c}{h}$.

Proof. As $\text{Succ}(h, l, n)$ is non-decreasing in n, we can assume that $n > \frac{l}{c}$, so that $\Pr(L_i = 1) < c$ at the beginning. By Lemma 32 and Proposition 31 we can assume that π satisfies the requirement for π' in 32.

Let π' be the protocol that starts of as π, but where the players start to pretend ignorance (as in the proof of Lemma 15) if $\Pr(L_i = 1 | T^k = t^k, X = x) = c$ for some i, current

transcript t^k and the true value x of X. This ensures that $\Pr(L_i = 1|T' = t, X = x) \leq c$ for all i and t. Let T' be the transcript of π'. From Theorem 5 we get

$$I(X; T') \leq \left(-\frac{\log(1-c)}{c} - \log(e)\right) l$$

We let Frank guess as he would if we used protocol π. By Fano's inequality, (3), Frank's probability of being wrong when he only sees the transcript of π' is

$$\begin{aligned} P_e &\geq \frac{H(X|T') - 1}{\log(|\mathcal{X}|)} \\ &= \frac{H(X) - I(X; T') - 1}{\log(|\mathcal{X}|)} \\ &\geq \frac{h - l\left(\frac{-\log(1-c)}{c} - \log(e)\right) - 1}{h} \end{aligned}$$

In the cases where Frank is wrong in π' there are two possibilities: Either the players did not pretend ignorance, in which case Frank would also be wrong if they used protocol π, or they did pretend ignorance so $\Pr(L_i = 1|T^k = t^k, X = x) = c$ for some i and some smallest k. When this first happens Eve can just ignore all further messages in π and guess that PLR_i is leaking. This way she is wins with probability at least c. Thus, all the situations in π' where Frank guesses wrong, correspond to situations in π where Eve would win with probability at least c. So Eve's probability of winning when the players are using protocol π is at least

$$cP_e \geq \frac{ch + l\log(1-c) + lc\log(e) - c}{h}$$

□

Corollary 34. *For fixed l we have*

$$\lim_{h\to\infty} \text{Succ}(h, l) = 0.$$

Proof. By Lemma 33 we have $\text{Succ}(h, l) \leq 1 - \frac{ch+l\log(1-c)+lc\log(e)-c}{h}$ for each $c \in (0, 1)$. Setting $c = 1 - \epsilon$ we get

$$\limsup_{h\to\infty} \text{Succ}(h, l) \leq \limsup_{h\to\infty} 1 - \frac{ch + l\log(1-c) + lc\log(e) - c}{h} = \epsilon.$$

As $\text{Succ}(h, l) \in [0, 1]$ and the above holds for all $\epsilon > 0$ we have $\lim_{n\to\infty} \text{Succ}(h, l) = 0$. □

Theorem 35. *Let $r > 0$ be a real number. Now*

$$\limsup_{l\to\infty} \mathrm{Succ}(\lfloor r\log(e)l\rfloor, l) \leq \frac{\log(r+1)}{r\log(e)}$$

Proof. Set $c = \frac{r}{r+1}$ and $h = \lfloor r\log(e)l\rfloor$ in Lemma 33. Then Eve's probability of winning is at least

$$\frac{r\lfloor r\log(e)l\rfloor - l(r+1)\log(r+1) + lr\log(e) - r}{\lfloor r\log(e)l\rfloor(r+1)}$$

As l tends to infinity, this tends to

$$\frac{r^2\log(e) - (r+1)\log(r+1) + r\log(e)}{r\log(e)(r+1)} = 1 - \frac{\log(r+1)}{r\log(e)}$$

as wanted. □

In particular we have the following corollary.

Corollary 36. *Let $h \to \infty$ and let $l = l(h)$ be a function of h with $l(h) = o(h)$. Then* $\mathrm{Succ}(h, l) \to 0$.

Proof. Let $l(h) = o(h)$ be a function. As $\mathrm{Succ}(h, l(h)) \in [0, 1]$ for all h, we only need to show that for all $\epsilon > 0$ there exists a h_0 such that for all $h \geq h_0$ we have $\mathrm{Succ}(h, l(h)) \leq 2\epsilon$. We see that $\frac{\log(r+1)}{r\log(e)} \to 0$ as $r \to \infty$, so we can find a number r such that $\frac{\log(r+1)}{r\log(e)} \leq \epsilon$. By Theorem 35 we have

$$\limsup_{l\to\infty} \mathrm{Succ}(\lfloor r\log(e)l\rfloor, l) \leq \frac{\log(r+1)}{r\log(e)} \leq \epsilon$$

So there exists a number l_1 such that for all $l \geq l_1$ we have

$$\mathrm{Succ}(\lfloor r\log(e)l\rfloor, l) \leq 2\epsilon.$$

As $l(h) = o(h)$ there is a h_1 such that for all $h \geq h_1$ we have $h \geq r\log(e)l(h)$. By Corollary 34 $\lim_{h\to\infty} \mathrm{Succ}(h, l) = 0$ for each value l. So for each value l there is a $h_2(l)$ such that for all $h \geq h_2(l)$ we have $\mathrm{Succ}(h, l) \leq 2\epsilon$. Define $h_2 = \max_{l<l_1} h_2(l)$. Now define $h_0 = \max(h_1, h_2)$. We want to show that for any $h \geq h_0$ we have $\mathrm{Succ}(h, l(h)) \leq 2\epsilon$. If $l(h) < l_1$, then $h \geq h_0 \geq h_2 \geq h_2(l(h))$, so $\mathrm{Succ}(h, l(h)) \leq 2\epsilon$. If $l(h) \geq l_1$ then

$$\begin{aligned}\mathrm{Succ}(h, l(h)) &\leq \mathrm{Succ}(\lfloor r\log(e)l\rfloor, l)\\ &\leq 2\epsilon.\end{aligned}$$

Here the first inequality follows from $h \geq h_0 \geq h_1$ and Proposition 27, and the second inequality follows from $l \geq l_1$. □

7. Hiding Among Innocents

Until now we have assumed, that even the players who are not trying to leak information will collaborate. In this section we will show that we do not need the non-leakers to collaborate. As long as some people are communicating innocently, and that communication is sufficiently non-deterministic, we can use these people as if they were collaborating.

Formally, we model the innocent communication by an innocent communication protocol. While protocols usually are designed to compute some function, innocent communication protocols is a way of describing what is already going on. An *innocent communication protocol* ι is a protocol that for each possible partial transcript s^k and each player i gives a finite set $\mathcal{A}_{i,s^k}$ of possible messages that that person can send in the next round, and a probability distribution on that set. In innocent communication protocols every person sends a message in each round. This assumption is not a restriction: if we have a protocol where only one player at a time sends messages, we can turn it into an innocent communication protocol, by requiring that all the other players send the message "no message" with probability 1. We will only be interested in innocent communication protocols that continue for infinitely many rounds. This assumption is of course unrealistic but in practice we only need it to be long.

Let S denote the random variable that is the infinite transcript we get from running ι, and let S^k denote the partial transcript of the first k rounds. For a player PLR_j and a partial transcript s^k of the first k rounds of ι we define

$$p_{max,j}(s^k) = \max_a(\Pr(A_{j,s^k} = a) | S^k = s^k),$$

where A_{j,s^k} is the message sent by PLR_j in round $k+1$. We say that ι is *informative* if for a random transcript S and for each player $\prod_{k\in\mathbb{N}} p_{max,j}(S^k) = 0$ with probability 1. In other words, if at each round in the protocol you try to guess what message PLR_j will send in the next round, then with probability 1 you will eventually fail. Notice that the model for innocent communication here is equivalent to what is used in [18], and the definition of informative is almost the same as the definition of *always informative* in [18] when one player is communicating.[8]

We say that a collaborating cryptogenography protocol π is *revealing* if there is a partial transcript t^k and a player PLR_j that is to send the next message A when the transcript is t^k and a message a such that PLR_j will send message a with positive probability if $L_j = 1$ but not if $L_j = 0$. If this is not the case, we say that π is *non-revealing*.[9] The point in cryptogenography is to hide who is sending the information, so we are only interested in non-revealing protocols.

The following is the main theorem of this section.

[8] The difference is that in [18], $\prod_{k\in\mathbb{N}} p_{max,i}(T^k)$ have to go to 0 exponentially fast.

[9] A non-revealing protocol can also reveal who the leakers are. For example, if it is known that exactly one person is leaking and all but one person sends a message that could not have been sent by a leaker. However, if $\Pr(L = (0, \ldots, 0)) > 0$ then a non-revealing protocol will never reveal anyone as a leaker.

Theorem 37. *Let π be a non-revealing collaborating cryptogenography protocol, and let ι be an informative communication protocol. Then there exists a protocol ι^π that is equivalent to π, but where the non-leakers follow the protocol ι.*

Proof. The idea is to construct the protocol ι^π and at the same time an interpretation function i that maps transcripts s of ι^π to transcripts t of π. We want them to satisfy the following.

1. For each partial transcript s^k of ι^π and each player PLR_j, ι^π gives a probability distribution, depending only on X, L_j, s^k and j that PLR_j will use to choose his next message.
2. If $L_j = 0$ then PLR_j choose her messages in ι^π using the same distributions as in ι.
3. The interpretation function i maps (infinite) transcripts s of ι^π to either transcripts t of π or to "error". The probability of error is 0.
4. If T denotes the transcript of π and S denotes the transcript of ι^π, then given that $i(S)$ is not error, $(X, L_1, \ldots, L_n, i(S))$ is distributed as $(X, L_1, \ldots, L_n, T)$.
5. For each transcript t of π, the random variable $(X, L_1, \ldots, L_n)$ is independent from S given $i(S) = t$.

Here the second requirement ensures that non-leakers can follow the protocol without knowing X or π. In fact, unlike in the collaborating communication protocol, they might be thinking that everyone is just having an innocent conversation. Thus in ι^π we refer to the non-leakers as *innocents*. Notice the important assumption that first the innocent communication protocol ι is defined and *then* we create a protocol ι^π for leaking information on top of that. This corresponds to assuming that the non-leaking players either do not care about the leak, or that they are oblivious to the protocol. If ι was allowed to depend what the leakers do, the non-leaking players could try to prevent the leak, and it would be a very different problem.

The fourth of the above requirements tells us that ι^π reveals at least as much about $(X, L_1, \ldots, L_n)$ as π and the last requirement says that we do not learn anything more. This ensures that Frank and Eve, who both know ι^π, learn exactly as much from the transcript of ι^π as they would from the transcript of π.

Proposition 38. *If ι^π satisfies the above requirements, then ι^π and π are equivalent.*

Proof. Recall Definition 11, of equivalence. i gives error with probability 0, so we can ignore all those cases. By requirement 4, $i(S)$ has the same distribution as T, and by requirement 4 and 5 the distribution μ_s of $(X, L_1, \ldots, L_n)$ given $S = s$ equals the distribution $\mu_{i(s)}$. Thus, μ_S, $\mu_{i(S)}$ and μ_T have the same distribution. □

Before we construct the protocol ι^π we will define a function i' that maps partial transcripts $s^{k'}$ of ι^π to tuples $(t^k, [y, z))$ where t^k is a partial transcript of π, and $[y, z) \subset [0, 1)$ is a half-open interval. When $i'(s^{k'}) = (t^k, [y, z))$, we refer to t^k as the interpretation of $s^{k'}$. Loosely speaking, the point of the interval is that not all messages in ι are sufficiently unlikely that they can correspond to a message in π, so instead

of interpreting them as a message in π, we store the information by remembering an interval. For an infinite transcript s, the function i' will satisfy

1. $i'(\lambda) = (\lambda, [0, 1))$, where λ is the empty string
2. If $i'(s^{k'}) = (t^k, [y, z))$ then either
   - $i'(s^{k'+1}) = (t^k \circ m, [0, 1))$ for some message m in π, or
   - $i'(s^{k'+1}) = (t^k, [y', z'))$, where $[y', z') \subseteq [y, z)$
3. If $i'(s^{k'}) = (t^k, [y, z))$ and t^k is a complete transcript for π, then $y = 0$, $z = 1$ and $i'(s^{k''}) = (t^k, [0, 1))$ for all $k'' \geq k'$

Thus every time we reveal one more round from the transcript s, we will either learn one message in π from the interpretation of s, or the interval gets smaller or stays the same. If the interpretation of $s^{k'}$ is t^k, we let $j(s^{k'})$ and $j(t^k)$ denote the index of the player to send the next message in π when the current transcript is t^k. When it is clear what $s^{k'}$ is, we write j instead of $j(s^{k'})$. If $i'(s^{k'}) = (t^k, [y, z))$ and $i'(s^{k'+1}) = (t^k \circ m, [0, 1))$ we say that at time k' player $j(s^{k'})$ finished sending the message m in π and at time $k' + 1$ player $j(s^{k'+1})$ starts sending a new message in π.

For each partial transcript t^k of π, we let $\mathcal{A}_{t^k}$ denote the set of possible next messages. We assume that all set of messages, both in π and ι, have an ordering, for example the lexicographical order. Algorithm 1 gives a pseudo code for i', but we will also define it in the main text.

Algorithm 1 i'.

```
1: procedure I'(s^{k'})
2:     t ← λ                          ▷ λ denotes the empty string, t a partial transcript
3:     k ← 0
4:     y ← 0
5:     z ← 1
6:     for r from 1 to k' do
7:         y' ← y + (z − y) Pr(S_{j(t),r} < s_{j(t),r} | S^{r−1} = s^{r−1})
8:         z' ← y + (z − y) Pr(S_{j(t),r} ≤ s_{j(t),r} | S^{r−1} = s^{r−1})
9:         y ← y'
10:        z ← z'
11:        if ∃a ∈ A_t : Pr(T_{k+1} < a | T^k = t, L_{j(t)} = 0) ≤ y,
12: z ≤ Pr(T_{k+1} ≤ a | T^k = t, L_{j(t)} = 0) then
13:            t ← t ∘ a
14:            k ← k + 1
15:            y ← 0
16:            z ← 1
17:        end if
18:    end for
19:    return (t, [y, z))
20: end procedure
```

Define a function $f : [0, 1) \to \mathcal{A}_{t^k}$ such that

$$f^{-1}(a) = [\Pr(T_{k+1} < a | T^k = t^k, L_j = 0), \Pr(T_{k+1} \leq a | T^k = t^k, L_j = 0)).$$

By definition of innocent communication protocol, each message in ι is chosen from a finite set, but to explain the point of the function f, imagine for now that ι said that in the next round PLR_j should send a random real uniformly from in $[0, 1)$. We could now interpret that as the message $f(x) \in \mathcal{A}_{t^k}$ in π. Then ι^π would say that if PLR_j was innocent he should send a number uniformly from $[0, 1)$ and if he was leaking, he should first choose $a \in \mathcal{A}_{t^k}$ using the distribution specified by π, and then send a number chosen uniformly at random from $f^{-1}(a)$. More generally, if ι said that PLR_j should choose his next message M from some continuous distribution on $\mathbb{R}$, we could take the quantile function given $L_j = 0$ of the message

$$m \mapsto \Pr(M < m | L_j = 0)$$

to turn it into a message that is uniform on $[0, 1)$ given $L_j = 0$. Unfortunately, there are only finitely many possible messages for PLR_j to send in each round, so instead of getting a number out of the quantile function, we define a similar function to get an interval. Let $i'(s^{k'}) = (t^k, [y, z))$ and let $\mathcal{M}_{j,s^{k'}}$ denote the set of possible messages that PLR_j can send in round $k' + 1$ when transcript is $s^{k'}$ and choose some ordering on the set. Define $g : [y, z) \to \mathcal{M}_{j,s^{k'}}$ by

$$g^{-1}(m) = \{y + (z - y)t | t \in [\Pr(M < m | L_j = 0), \Pr(M \leq m | L_j = 0))\}.$$

Thus instead of getting a number in $[0, 1)$ out of $m \in \mathcal{M}_{j,s^{k'}}$, we get an interval $g^{-1}(m)$, whose length is proportional to the probability that an innocent player would send that message. If $g^{-1}(m) \subset f^{-1}(a)$ for some $a \in \mathcal{A}_{t^k}$ we say that PLR_j sends a in π and define $i'(s^{k'+1}) = (t^k \circ a, [0, 1))$. Otherwise, PLR_j is not done sending his message and we define $i'(s^{k'+1}) = (t^k, g^{-1}(m))$. Algorithm 1 gives a pseudo code for computing i'. Here $s_{j,r}$ denotes the message in ι sent by player j in round r.

Now if for some k' we have $i'(s^{k'}) = (t, [0, 1))$ where t is a complete transcript of π we define $i'(s^{k''}) = (t, [0, 1))$ for all $k'' > k'$ and $i(s) = t$. If for some s no such k' exists, we define $i(s)$ to give "error".

Next we define the protocol ι^π. Any non-leaking player chooses his messages as given by ι and when the current transcript is $s^{k'}$ all players except $\text{PLR}_{j(s^{k'})}$ also choose their messages as in ι. When a leaking player, $\text{PLR}_{j(s^{k'})}$, starts sending a message in π, he first choose the message $a \in \mathcal{A}_{t^k}$ using the distribution given by π (this distribution depends on $X = x$). Next he chooses a number α randomly and uniform in $f^{-1}(a)$. Until he has sent his message in π he will now send messages m such that $\alpha \in g^{-1}(m)$. This uniquely specifies which messages m to send (notice that g will depend on current transcript in ι^π, so m is not necessarily the same for every round). When we get to a transcript $s^{k'}$ that is interpreted as a complete transcript t of π all the players will just follow ι. Figure 3 gives an example of how one message in π is send by using ι^π.

We see that if in π a leaking player's distribution of a is exactly the same as a non-leaking players, then the distribution of the number α chosen by the leaking player in uniform on $[0, 1)$. By the definition of g, the probability that a leaking player sends a particular message m in ι^π is exactly the probability given by ι, and thus the same as a non-leaking player. Using this reasoning in the opposite direction, this tells us that we

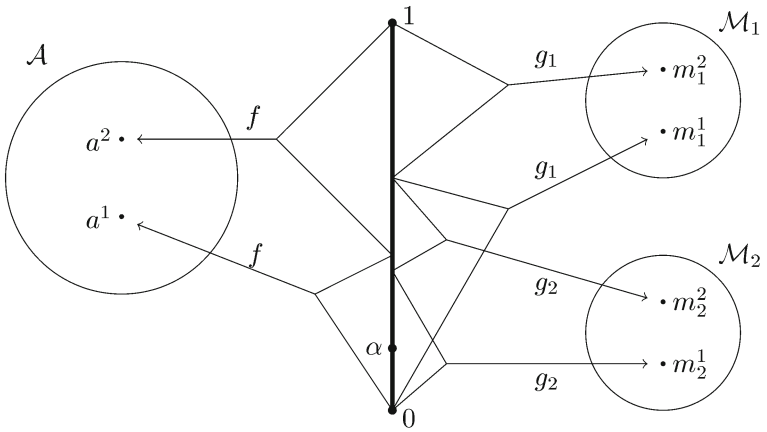

Fig. 3. Example of how to construct a part of ι^π. In this figure we see an example of how construct a part of ι^π. In π, the next player to send a message is PLR_j. The message A_1 should come from $\mathcal{A} = \{a^1, a^2\}$. We have $\Pr(A_1 = a^1 | L_j = 0) = 0.4$, so $f : [0, 1) \to \mathcal{A}$ maps $x \in [0, 0.4)$ to a^1, and $x \in [0.4, 1)$ to a^2. Now $L_j = 1$, so PLR_j first chooses a message from $\mathcal{A}$ to send, this happens to be a^1, and then a number α chosen randomly and uniformly from $f^{-1}(a^1)$. In ι, the next message M_1 that PLR_j sends should be from $\mathcal{M}_1 = \{m_1^1, m_1^2\}$. If PLR_j was innocent and was following the protocol ι, we would have $\Pr(M_1 = m_1^1) = 0.6$, so $g_1 : [0, 1) \to \mathcal{M}_1$ maps $x \in [0, 0.6)$ to m_1^1 and the rest to m_1^2. As $\alpha \in [0, 0.6)$, PLR_j now sends the message m_1^1. We see that $g_1^{-1}(m_1^1)$ overlaps with both $f^{-1}(a^1)$ and $f^{-1}(a^2)$, so an observer cannot yet determine which message in π PLR_i was sending, so PLR_j has not sent his message yet. His next message M_2 should be choosen from $\mathcal{M}_2 = \{m_2^1, m_2^2\}$, and again it happens that if he was following ι then $\Pr(M_2 = m_2^1) = 0.6$, so $g_2 : [0, 0.6) \to \mathcal{M}_2$ maps $x \in [0, 0.36)$ to m_2^1 and the rest to m_2^2. As $\alpha \in [0, 0.36)$, PLR_j sends the message m_2^1, and now $g_2^{-1}(m_2^1) \subset f^{-1}(a^1)$, so now an observer can see that PLR_j was sending the message a^1 in π, and PLR_j is done sending his message in π.

can assume that even the innocents, when starting sending a message in π, choose a uniformly distributed $\alpha \in [0, 1)$ and send the message m such that $\alpha \in g(m)$, until they have sent the message in π. They may not do that, but the probability of any transcript is the same as if they did.

Finally, we need to check that ι^π satisfies the 5 requirements. The first two follows from the construction. To show the third, we need to show that for a random transcript s of ι^π there will with probability 1 exist a k' such that $i'(s^{k'}) = (t, [0, 1))$ where t is a complete transcript for π. As π only has finitely many rounds, it is enough to show that for each message of π we start sending in ι^π, there is probability 1 that we will finish sending it. Assume that $i'(s^{k'}) = (t^k, [0, 1))$ for some k', where t^k is an incomplete transcript of π, but for all $k'' > k'$ the interpretation of $s^{k''}$ is still t^k. If $\text{PLR}_{j(s^{k'})}$ is innocent, everyone will be following ι, so by the assumption that ι is informative, the set of transcripts where the length of the interval does not go to 0 has probability 0. As stated earlier we can assume that when sending a message in π, even the innocents start by choosing a random number α uniformly from $[0, 1)$. As f only jumps in finitely many points, there is probability 0 that $\text{PLR}_{j(s^{k'})}$ chooses one of these points. If he does not, and the length of the interval goes to 0, he will eventually send his message in π. Thus, there is probability 0 that a non-leaker does not send his message. A leaker

chooses his random $\alpha \in [0, 1)$ using a different distribution, but we can divide $[0, 1)$ into a finite set of intervals (given by $f^{-1}(a)$) such that it is uniform on each of these intervals. This tells us that given $s^{k'}$ there is a constant K such that, as long as $\text{PLR}_{j(s^{k'})}$ is still sending the same message in π, any continuation of the transcript is at most K times more likely when $\text{PLR}_{j(s^{k'})}$ is leaking as when he is not leaking. Thus, there is still probability $K \cdot 0 = 0$ that he will not finish his message in π.

For the fourth requirement, we observe that any leaking player is actually choosing messages in π following the distribution given by π, and then making sure that the message send in ι^π will be interpreted as the message he wanted to send in π. The innocent players are not doing this, but we have seen that the distribution on the message they send in ι^π are the same as if they did. Thus, requirement 4 holds. Finally, we see that given $i(S) = t$ a player not sending a message in π always follows ι and a player sending a message in π can be thought of as haven chosen an α uniformly from $f^{-1}(a)$ where a is the next message in transcript t. This is independent from $(X, L_1, \ldots, L_n)$ and thus the last requirement follows. □

To implement the protocol ι^π the leaking players do not have to choose all the infinitely many digits in a random number α. Instead, they can just for each message compute the probability that they would send each message, given that they had chosen an α. We also see that if $i(S)$ does not give an error, then there is some k such that S^k determines $i(S)$. If we let K be the random variable that is ∞ when we have error and otherwise gives the smallest value k such that S^k determines $i(S)$, then we know that $\Pr(K = \infty) = 0$. So all the probability mass of K is on the integers, hence for any $\epsilon > 0$ there must exists some k_0 such that $\Pr(K \geq k_0) < \epsilon$. That is, $i(S^{k_0})$ gives a total transcript with probability greater than $1 - \epsilon$

In order to find the protocol ι^π you need have a description of the protocol ι. This is a strong assumption: even if you are able to communicate innocently, it does not mean that you are aware of the distribution you use to pick your random messages. In steganography, the weaker assumption that you have a random oracle that takes history and player index as input and gives a message following the innocent distribution as output, is sometimes enough [18]. However, it is not clear if this weaker assumption is enough for doing cryptogenography. While it may not be possible to find ι for all kinds of innocent communications, there are situations where we can approximate ι very well. For example, if a person posts blog posts, we can consider the message to be only the parity of the minutes in the sending time. This value will probably, for most people, be close to uniformly distributed on $\{0, 1\}$.

8. Open Problems

In this paper we only considered how much information l players can leak in an asymptotic sense, where l tends to infinity, and the proof of the achievability results is not constructive. We have not tried to find any explicit protocols that work well for fixed specific values of l and tolerance of errors ϵ, but that would be an interesting possibility for further research. We assumed that both Eve and Frank knew the true distribution q

of $(X, L_1, \ldots, L_n)$. It might be interesting to consider the problem where their beliefs, q_E and q_F are different from q and from each other.

We have only found the c-capacity for Fixed and for Indep_b. It would be interesting to find a way to compute the capacity of more general $\mathfrak{L}$-structures. It would also be interesting so find the safe and the risky capacities in the last four adaptive models, in particular to see if there in this model can be a difference between the safe and the risky capacities.

In the setup we considered here, there are two types of players. Some know the information that we want to leak and some do not. We could also imagine that some people know who knows the information, without knowing the information itself, and some could know who knows who knows the information and so on. We could also have people who would only know X if it belongs to some set S, and otherwise only know that $X \notin S$. It is known from the game theory literature that all of this can be described by having a joint distribution $(X, P_1, \ldots, P_n)$ where X is the information we want to leak and P_i is a random variable known to player i [1].

A different generalisation would be to have players who try to prevent the leakage by sending misleading information. Such players would also not want to be discovered. If Frank notices that someone is sending misleading information, he could just ignore all the messages sent by that person.

Acknowledgements

I would like to thank my supervisors, Peter Keevash and Søren Riis for valuable discussions about cryptogenography. I also want to thank Jalaj Upadhyay for pointing me to [18].

References

[1] R.J. Aumann. Interactive epistemology I: Knowledge. *Int. J. Game Theory*, **28**(3), 263–300 (1999)
[2] M. Backes, A. Kate, P. Manoharan, S. Meiser, E. Mohammadi, AnoA: A framework for analyzing anonymous communication protocols, in *2013 IEEE 26th Computer Security Foundations Symposium (CSF)* (IEEE, 2013), pp. 163–178
[3] R. Bagai, H. Lu, R. Li, B. Tang, An accurate system-wide anonymity metric for probabilistic attacks, in *Proceedings of the 11th Privacy Enhancing Technologies Symposium (PETS 2011)* (2011)
[4] J. Brody, S. Jakobsen, D. Scheder, P. Winkler, Cryptogenography, in *ITCS* (2014)
[5] D. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, **24**(2), 84–90 (1981).
[6] D. Chaum, The dining cryptographers problem: Unconditional sender and recipient untraceability. *J. Cryptol.***1**, 65–75 (1988)
[7] S. Clauß, S. Schiffner, Structuring anonymity metrics, in *Proceedings of the Second ACM Workshop on Digital Identity Management, DIM '06* (ACM, New York, NY, 2006), pp. 55–62
[8] T.M. Cover, J.A. Thomas. *Elements of Information Theory* (Wiley-Interscience, New York, NY, 1991)

[9] G. Danezis, C. Diaz, A survey of anonymous communication channels. Technical Report MSR-TR-2008-35, Microsoft Research (2008)
[10] C. Diaz, S. Seys, J. Claessens, B. Preneel, Towards measuring anonymity. In R. Dingledine, P. Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*, LNCS 2482 (Springer, 2002)
[11] R. Dingledine, N. Mathewson, P. Syverson, Tor: The second-generation onion router, in *Proceedings of the 13th USENIX Security Symposium* (2004)
[12] B. Doerr, M. Künnemann, Improved protocols and hardness results for the two-player cryptogenography problem (2016). arXiv:1603.06113
[13] C. Dwork, A. Roth, The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.9*(3-4), 211–407 (2014)
[14] M. Edman, F. Sivrikaya, B. Yener, A combinatorial approach to measuring anonymity, in *2007 IEEE Intelligence and Security Informatics* , pp. 356–363 (2007)
[15] Freehaven. Anonymity bibliography. http://www.freehaven.net/anonbib/.
[16] B. Gierlichs, C. Troncoso, C. Diaz, B. Preneel, I. Verbauwhede, Revisiting a combinatorial approach toward measuring anonymity. In V. Atluri, M. Winslett, editors, *WPES '08: Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society* (ACM, Alexandria, VA, 2008), pp. 111–116
[17] D.M. Goldschlag, M.G. Reed, P.F. Syverson, Hiding routing information. In R. Anderson, editor, *Proceedings of Information Hiding: First International Workshop*, LNCS 1174 (Springer, 1996), pp. 137–150
[18] N.J. Hopper, *Toward a Theory of Steganography*. PhD thesis, Carnegie Mellon University (2004)
[19] E. Kushilevitz, N. Nisan, *Communication Complexity* (Cambridge University Press, New York, NY, 1997)
[20] A. Serjantov, G. Danezis, Towards an information theoretic metric for anonymity. In R. Dingledine, P. Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*, LNCS 2482 (Springer, 2002)
[21] C.E. Shannon, ACM SIGMOBILE Mobile Computing and Communications Review. *A mathematical theory of communication*, **5**(1), 3–55 (2001)
[22] G. Tóth, Z. Hornák, F. Vajda, Measuring anonymity revisited. In S. Liimatainen, T. Virtanen, editors, *Proceedings of the Ninth Nordic Workshop on Secure IT Systems* (2004), pp. 85–90
[23] J. Van Den Hooff, D. Lazar, M. Zaharia, N. Zeldovich, Vuvuzela: Scalable private messaging resistant to traffic analysis, in *Proceedings of the 25th Symposium on Operating Systems Principles* (ACM, 2015), pp. 137–152
[24] A. Chi-Chih Yao, Some complexity questions related to distributive computing (preliminary report), in *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing, STOC '79* (ACM, New York, NY, 1979), pp. 209–213
[25] Y. Zhu, R. Bettati, Anonymity vs. information leakage in anonymity systems, in *ICDCS'05* (2005), pp. 514–524