How to Bootstrap Anonymous Communication

Sune K. Jakobsen
School of Mathematical Sciences and School of Electronic Engineering and Computer Science
Queen Mary University of London
S.K.Jakobsen@qmul.ac.uk

Claudio Orlandi
Department of Computer Science
Aarhus University
orlandi@cs.au.dk

ABSTRACT

We ask whether it is possible to anonymously communicate a large amount of data using only public (non-anonymous) communication together with a small anonymous channel. We think this is a central question in the theory of anonymous communication and to the best of our knowledge this is the first formal study in this direction.

Towards this goal, we introduce the novel concept of *anonymous steganography*: think of a leaker Lea who wants to leak a large document to Joe the journalist. Using anonymous steganography Lea can embed this document in innocent looking communication on some popular website (such as cat videos on *YouTube* or funny memes on *9GAG*). Then Lea provides Joe with a short decoding key *dk* which, *when applied to the entire website*, recovers the document while hiding the identity of Lea among the large number of users of the website. Our contributions include:

- Introducing and formally defining *anonymous steganography*,
- A construction showing that anonymous steganography is possible (which uses recent results in circuits obfuscation),
- A lower bound on the number of bits which are needed to bootstrap anonymous communication.

1. INTRODUCTION

The problem. Lea the leaker wants to leak a big document to Joe the journalist in an anonymous way.[1] Lea has a way of anonymously communicating a small number of bits to Joe, but the size of the document she wants to leak is orders of magnitudes greater than the capacity of the anonymous channel between them. In this paper we ask whether it is possible to "bootstrap" anonymous communication, in the sense that we want to construct a "large" anonymous channel using only public (non-anonymous) communication channels together with a "small" anonymous channel. We find the question to be central to the theory of anonymous communication and to the best of our knowledge this is the first formal study in this direction.

Steganography. The goal of (traditional) steganography is to hide that a certain communication is taking place, by embedding sensitive content in innocent looking traffic (such as pictures, videos, or other redundant documents). There is no doubt that steganography is a useful tool for Lea the leaker: using steganography she could send sensitive documents to Joe the journalist in such a way that even someone monitoring all internet traffic would not be able to notice that this communication is taking place. However, steganography alone cannot help Lea if she wants to make sure that *even Joe* does not learn her identity.

Anonymous Steganography. To solve this problem, we introduce a novel cryptographic primitive, which we call *anonymous steganography*. Very informally, anonymous steganography works in the following way: Lea wants to communicate a sensitive (large) message x to Joe. To do so, she embeds x in some (large) innocent looking document c which she uploads to a popular website (not necessarily in an anonymous way). Then Lea produces some (short) decoding key dk (which is a function of c and all other documents on the website – or at least a set large enough so that her identity is hidden in a large group of users, such as "all videos uploaded last week") which she then communicates to Joe using an anonymous channel. Now Joe is able to recover the original message x from the website using the decoding key dk, but at the same time Joe has no way of telling which document contains the message (and therefore which of the website users is the leaker). Intuitively, it is crucial for Lea's anonymity that Joe can only decode *the entire website*: if Joe had a way of decoding single documents (or portions) he would easily be able to pinpoint which document actually contains the leaked message.

In Section 2 we formally introduce anonymous steganography and in Section 3 we show how to construct such a scheme. Unfortunately our positive result crucially relies on heavy tools such as homomorphic encryption and indistinguishability obfuscation for circuits ($i\mathcal{O}$ for short), making it very far from being useful in practice. We leave it as a major open question to construct such schemes using simpler and more efficient cryptographic tools (perhaps even at the price of relaxing the definition of anonymity). Other open problems include studying whether the computational complexity for the leaker must depend on the size of the

[1]This naming convention is courtesy of Nadia Heninger.

ITCS'16, January 14–16, 2016, Cambridge, MA, USA.

ACM 978-1-4503-4057-1/16/01 ...$15.00.
DOI: http://dx.doi.org/10.1145/2840728.2840743.

anonymity set if the leaker is given a hash of all the documents, and whether it is possible to construct more efficient protocols if multiple leakers are leaking to Joe at once.

Lower Bounds. In the scheme described above, Lea sends Joe a decoding key dk using a (pre-existing) anonymous channel. It is a natural question to ask whether this is necessary, or if we can construct a scheme where all communication between Lea and Joe takes place over regular channels. Unfortunately the latter is too good to be true, and in Section 5 we prove that it is impossible to construct an anonymous steganography scheme unless Lea sends a key (of super-logarithmic size) to Joe. The idea behind the proof is: if the scheme is correct at some point the probability that Joe outputs x has to increase from polynomially small to 1. Joe can estimate how each message (sent by any of the users over the non-anonymous channel) affects this probability and concludes that the message which changes this probability the most must come from Lea. Hence, the messages that causes this increase has to be sent over an anonymous channel.

Related Work. Steganography and information-hiding are an important line of research in the signal-processing community. For a background on steganographic techniques see one of the textbooks on the subject e.g., [Fri09]. For a more complexity-theoretic treatment of steganography (both in the private- and in the public-key setting) see e.g. [HLA02, HAL09, AH04, BC05, FNP14] and references therein. There exist practical tools which allow whistleblowers to anonymously communicate with journalists, such as SecureDrop, which in turns relies on Tor [DMS04]. Tor is by far the most popular anonymous internet communication system, and relies on a volunteer network of relays. Several countries have tried, with different degrees of success, to block or limit Tor traffic [WL12]. While anonymous steganography is in no way practical at this stage, we find the question of whether similar ideas could be used to amplify the bandwidth offered by systems like Tor combining anonymous and non-anonymous channels extremely intriguing. *Message In A Bottle* [IKV13] is a protocol where Lea can encrypt her message under Joe's public key, embed it in an image using steganography and post the image on any blog. Joe will now monitor all blogs to see if someone left a (concealed) message for him. Interestingly [IKV13] shows that this approach is feasible in practice and because Lea can use any blog, it will be costly for e.g. a government to prevent Lea from sending the message to Joe. However, in this protocol Joe learns Lea's identity, which is what we are trying to prevent in our work. In *cryptogenography* [BJSW14, Jak14] a group of users cooperate to allow a leaker to publish a message with some reasonable degree of anonymity: here we want that anyone should be able to recover the message from the protocol transcript, but no one (even a computationally unbounded observers) should be able to determine with certainty the identity of the leaker. In other words in cryptogenography we are happy as long as the observer cannot produce evidence which proves with certainty the identity of the leaker (which could be used e.g., in a court case). In [BJSW14] the leaker can publish one bit correctly but no observer can guess the identity of the leaker with probability more than 44%. In [Jak14] instead a different setting is considered, where multiple leakers agree to publish some information while hiding their identity by blending into an arbitrarily large group. The leakers do not need perfect anonymity, but just want to ensure that for each leaker, an observer will never assign a probability greater that c to the event that that person is a leaker. It is shown that for any $\epsilon > 0$ and sufficiently large n, n leakers can publish $\left(-\frac{\log(1-c)}{c} - \log(e) - \epsilon\right) n$ bits, where e is the base of the natural logarithm. Our work is inspired by the model in [Jak14]. The main difference is that we assume the adversary has bounded computational power, so we only need one leaker and we get all but negligible anonymity. In [IKOS06] the authors investigated how an anonymous channel could be used to implement other cryptographic primitives, but not if it could be used to bootstrap a larger anonymous channel. From a technical point of view, our positive result is inspired by the clever techniques of Hubáček and Wichs [HW15] to compress communication using $i\mathcal{O}$ for circuits.

2. DEFINITIONS

Notation. We write $[x, y]$ with $x < y \in \mathbb{N}$ as a shorthand for $\{x, \ldots, y\}$ and $[x]$ as a shorthand for $[1, x]$. If v is a vector $(v_1, \ldots, v_n)$ then v_{-i} is a vector such that $(v_1, \ldots, v_{i-1}, \perp, v_{i+1}, \ldots v_n)$ and $(v_{-i}, v_i) = v$. (This notation trivially extends to multiple indices). A function is *negligible* if it goes to 0 faster than the inverse of any polynomial. We write $\mathsf{poly}(\cdot)$ and $\mathsf{negl}(\cdot)$ for a generic polynomial and negligible function respectively (When we write about asymptotic behaviour, e.g. "super-logarithmic size", "polynomial size" or "exponentially small" without explicitly mentioning which parameter we are comparing with, the implicit parameter is always the security parameter λ.). $x \leftarrow S$ denotes sampling a uniform element x from a set S. If A is an algorithm $x \leftarrow A$ is the output of A on a uniformly random tape. We highlight constants $\alpha, \beta, \ldots$, hardwired in a circuit C using the notation $C[\alpha, \beta, \ldots]$.

Anonymous Steganography. For the sake of presentation, we make the simplifying assumption that honest users sample their documents independently from the uniform distribution. It is then possible to combine our scheme with a regular steganographic scheme to obtain anonymous steganography for arbitrary sources, as discussed in Section 4.

We define an *anonymous steganography* scheme as a tuple of algorithms $\pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{KeyEx}, \mathsf{Dec})$. All algorithms, even when not specified, take as input the security parameter λ, and the length parameters s, ℓ, ℓ', d (s is *short*, ℓ is *long*). The syntax of the algorithm is as follows:

- $ek \leftarrow \mathsf{Gen}(1^\lambda)$ is a randomized algorithm which generates an encoding key ek.
- $c \leftarrow \mathsf{Enc}_{ek}(x)$ is a randomized algorithm which encodes a secret message $x \in \{0,1\}^{\ell'}$ into a (random looking) document $c \in \{0,1\}^{\ell}$.
- $dk \leftarrow \mathsf{KeyEx}_{ek}(t, i)$ takes as input a public vector of documents $t \in (\{0,1\}^{\ell})^d$, an index $i \in [d]$ such that $t_i = c$ and extracts a (short) decoding key $dk \in \{0,1\}^s$.
- $x' = \mathsf{Dec}_{dk}(t)$ recovers a message x' using the decoding key dk and the public vector of documents t in a deterministic way.

Remark. We chose to keep Gen separated from Enc since a single key could be used to encode multiple messages –

in a natural extension of the scheme Lea hides her secret(s) in a subset of documents $I \subset [d]$. Finally, KeyEx is a separated algorithm since it takes as input documents which are generated from honest users *after* c is published.

How to Use The Scheme. To use anonymous steganography, Lea generates the encoding key ek using Gen, waits until some honest users publish content online and then encodes her secret x using Enc_{ek} to get the c. Lea then uploads c to this website, and waits until more honest content is published. Then she chooses the set of documents she is hiding among, for example, all files uploaded to this website during that day/week. Lea then downloads all these documents t and finds the index i of her own document in this set. Finally she computes $dk \leftarrow \mathsf{KeyEx}_{ek}(t, i)$, and uses the small anonymous channel to send dk to Joe together with a pointer to t.

Properties. We require the following properties: *correctness* (meaning that $x' = x$ with overwhelming probability), *compactness* (meaning that $s < \ell'$) and *anonymity* (meaning that the receiver does not learn any information about i). Another natural requirement is *confidentiality* (meaning that one should not be able to learn the message without the decoding key dk), but it is easy to see that this follows from *anonymity.* Formal definitions follow:

Definition 1 (Correctness)**.** *We say an anonymous steganography scheme is q-*correct *if for all $\lambda \in \mathbb{N}, x \in \{0,1\}^{\ell'}$, $i \in [d], t_{-i} \in (\{0,1\}^{\ell})^{d-1}$ over $\{0,1\}^{\ell}$ the following holds:*

$$\Pr\left[\mathsf{Dec}_{dk}\left((t_{-i}, c)\right) = x\right] \geq q,$$

where $ek \leftarrow \mathsf{Gen}(1^{\lambda})$, $c \leftarrow \mathsf{Enc}_{ek}(x)$, $dk \leftarrow \mathsf{KeyEx}_{ek}((t_{-i}, c), i)$ and the probabilities are taken over all the random coin flips. We simply say that a scheme is correct *when $q \geq 1 - \mathsf{negl}(\lambda)$.*

Definition 2 (Anonymity)**.** *We define a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:*

1. *The challenger $\mathcal{C}$ generates a key $ek \leftarrow \mathsf{Gen}(1^{\lambda})$;*
2. *The adversary $\mathcal{A}$ outputs a message $x \in \{0,1\}^{\ell'}$, two indices $i_0 \neq i_1 \in [d]$, and a vector $t_{-(i_0,i_1)}$;*
3. *The challenger $\mathcal{C}$:*
   (a) *samples a bit $b \leftarrow \{0,1\}$;*
   (b) *computes $t_{i_b} \leftarrow \mathsf{Enc}_{ek}(x)$ and samples $t_{i_{1-b}} \leftarrow \{0,1\}^{\ell}$;*
   (c) *computes $dk \leftarrow \mathsf{KeyEx}_{ek}\left((t_{-(i_0,i_1)}, (t_{i_0}, t_{i_1})), i_b\right)$*
   (d) *outputs dk, t;*
4. *$\mathcal{A}$ outputs a guess bit g;*

We say π satisfies anonymity *if for all PPT $\mathcal{A}$*

$$\left|\Pr[g = b] - \frac{1}{2}\right| = \mathsf{negl}(\lambda) \ .$$

Building Blocks. We will need the following ingredients in our construction: 1) an *indistinguishability obfuscator ($i\mathcal{O}$)* $\bar{C} \leftarrow \mathcal{O}(C)$ which takes any polynomial size circuit C and outputs an obfuscated version $\bar{C}$ [GGH+13]; 2) A *compact* homomorphic encryption scheme ($\mathsf{HE.G}, \mathsf{HE.E}, \mathsf{HE.D}, \mathsf{HE.Eval}$); 3) A pseudorandom function f; 4) A vector commitment scheme ($\mathsf{VC.G}, \mathsf{VC.C}, \mathsf{VC.D}, \mathsf{VC.V}$) which allows to commit to a long string x using $\mathsf{VC.C}$, and where it is possible to decommit to individual bits of x using $\mathsf{VC.D}$. Crucially, the proof of correct decommitting π^j for any bit j has size at most polylogarithmic in $|x|$. In addition, we need that the vector commitment scheme is *somewhere statistically binding* according to the definition of Hubáček and Wichs [HW15]: in a nutshell, this means that when generating a commitment key ck it is possible to specify a special position i such that a) any commitment generated using the key ck is statistically binding for the i-th bit of x (this property is crucial to be able to verify these commitments inside circuits obfuscated using $i\mathcal{O}$) and that b) ck computationally hides the index i. Such a vector commitment scheme can be constructed from fully-homomorphic encryption [HW15]. Formal definitions of these tools can be found in the full version of this paper [**?**].

3. A PROTOCOL FOR ANONYMOUS STEGANOGRAPHY

We start with a high-level description of our protocol (in steps) before presenting the actual construction and proving that it satisfies our notion of anonymity.

First attempt. Let the encoding key ek be a key for a PRF f, and let the encoding procedure simply be a random looking "symmetric encryption" of x using this PRF. Clearly now the resulting document c is indistinguishable from other elements sampled from the uniform distribution over $\{0,1\}^{\ell}$.

In this first attempt we let the decoding key dk be the obfuscation of a circuit $C[i, ek, \gamma](t)$: The circuit contains two hard-wired secrets, the index of Lea's document $i \in [d]$ and the key for the PRF ek. It also contains the hash of the entire set of documents $\gamma = H(t)$. On input a database t the circuit checks if $\gamma = H(t)$ and if this is the case outputs x by decrypting t_i with ek.

Clearly this first attempt fails miserably since the size of the circuit is now proportional to the size of the entire database $t = d\ell$, which is even larger than the size of the secret message $|x| = \ell' \leq \ell$.

Second attempt.[2] To remove the dependency on the number of documents d, we include in the decoding key an encryption $\alpha = \mathsf{HE.E}_{pk}(i)$ of the index i (using the homomorphic encryption scheme), and an obfuscation of a (new) circuit $C[ek, sk, \gamma](\beta)$, which contains hardwired secrets ek and sk (the secret key for the homomorphic encryption scheme), as well as a hash $\gamma = H(\mathsf{HE.Eval}(\mathsf{mux}[t], \alpha))$, where the circuit $\mathsf{mux}[t](i)$ outputs t_i. The circuit C now checks that $\gamma = H(\beta)$ and if this is the case computes $t_i \leftarrow \mathsf{HE.D}_{sk}(\beta)$ using the secret key of the HE scheme, then decrypts t_i using the PRF key ek and outputs the secret message x. When Joe receives the decoding key dk, Joe constructs the circuit $\mathsf{mux}[t]$ (using the public t) and (re)-computes $\beta = \mathsf{HE.Eval}(\mathsf{mux}[t], \alpha)$. To learn the secret, he runs the obfuscated circuit on β.

In other words, we are now exploiting the compactness of the homomorphic encryption scheme to let Joe compute an encryption of the document $c = t_i$ from the public database t and the encryption of i. Since Lea the leaker can predict

[2]A different approach at this stage could be to use $i\mathcal{O}$ for Turing machines [KLW15]. Unfortunately, [KLW15] uses *complexity-leveraging* and therefore must assume *sub-exponentially hard $i\mathcal{O}$* for circuits, while the solution described next will be secure using only standard hardness.

this ciphertext[3], she can construct a circuit which only decrypts when this particular ciphertext is provided as input. However, the size of β (and therefore C) is proportional to $\mathsf{poly}(\lambda) + \ell$, thus we are still far from our goal.[4]

Third attempt. To remove the dependency from the length of the document ℓ, we construct a circuit which takes as input an encryption of a single bit j of t_i instead of the whole ciphertext. However, we also need to make sure that the circuit only decrypts these particular ciphertexts, and does not help Joe in decrypting anything else. Moreover, the circuit must perform this check in an efficient way (meaning, independent of the size of ℓ), so we cannot simply "precompute" these ℓ ciphertexts and hardwire them into C.

This is where we use the vector commitment scheme: we let the decoding key include a (short) commitment key ck. We include in the obfuscated circuit a (short) commitment $\gamma = \mathsf{VC.C}_{ck}(\beta)$ (where $\beta = (\beta^1, \dots, \beta^\ell)$ is a vector of encryptions of bits) and we make sure that the circuit only helps Joe in decrypting these ℓ ciphertexts (and nothing else). In other words, we obfuscate the circuit $C[ek, sk, ck, \gamma](\beta', \pi', j)$ which first checks if $\mathsf{VC.V}_{ck}(\gamma, j, \beta', \pi') = 1$ and if this is the case it outputs the j-th bit of x from the j-th bit of the ciphertext $t_i^j \leftarrow \mathsf{HE.D}_{sk}(\beta')$.[5] We have now almost achieved our goal, since the size of the decoding key is $\mathsf{poly}(\lambda \log(d\ell))$.

Final attempt. We now have to argue that our scheme is secure. Intuitively, while it is true that the index i is only sent in encrypted form, we have a problem since the obfuscated circuit contains the secret key for the homomorphic encryption scheme, and we therefore need a final fix to be able to argue that the adversary does not learn any information about i.

The final modification to our construction is to encrypt the index i twice under two independent public keys. From these encryptions Joe computes two independent encryptions of the bit t_i^j which he inputs to the obfuscated circuits together with proofs of decommitment. The circuit now outputs $\perp$ if any of the two decommitment proofs are incorrect, otherwise the circuit computes and outputs x^j from one of the two encryptions (and ignores the second ciphertext).

Anonymity. Very informally, we can now prove that Joe cannot distinguish between the decoding keys computed using indices i_0 and i_1 in the following way: we start with the case where the decoding key contains two encryptions of i_0 (this corresponds to the game in the definition with $b = 0$). Then we define a hybrid game where we change one of the two ciphertext from being an encryption of i_0 with an encryption of i_1. In particular, since we change the ciphertext which is ignored by the obfuscated circuit, this does not change the output of the circuit at all (and we can argue indistinguishability since the obfuscated circuit does not contain the secret key for this ciphertext). We also replace the random document c_{i_1} with an encryption of x with a new key for the PRF. Finally we change the obfuscated circuit and let it recover the message x from the second ciphertext. Thanks to the SSB property of the commitment scheme it is possible to prove, in a series of hybrids, that the adversary cannot notice this change. To conclude the proof we repeat the hybrids (in inverse order) to reach a game which is identical to the definition of anonymity when $b = 1$.

Before moving on to the actual construction, we describe an attack that is implicitly ruled out by our construction: Consider an adversary that picks a random t^* and computes

$$\beta^* = \mathsf{HE.Eval}(\mathsf{mux}[(t_{-i_0}, t^*)], \alpha)$$

and feeds it to the obfuscated program. It might seem that the adversary can find b by checking if the decoding program returns x on input β^* or not. However, this does not work since it is hard to find two databases $t \neq t^*$ such that $\beta = \beta^*$ where $\beta = \mathsf{HE.Eval}(\mathsf{mux}[t], \alpha)$ and $\beta^* = \mathsf{HE.Eval}(\mathsf{mux}[t^*], \alpha)$: suppose wlog $t[1] \neq t^*[1]$. If $\mathsf{HE.D}(\alpha) = 1$ then $\mathsf{HE.D}(\beta) \neq \mathsf{HE.D}(\beta^*)$ which implies (since the HE is correct and decryption is deterministic) that $\beta \neq \beta^*$. But then one can use this as a test to distinguish encryptions of 1 from encryptions of other indices, contradicting the IND-CPA security of the HE scheme.

The Actual Construction. A complete specification of our anonymous steganography scheme follows. Note that in our construction $\ell' = \ell$.

Key Generation: On input the security parameter λ the algorithm Gen samples a random key $ek \in \{0,1\}^\lambda$ for the PRF and outputs ek.

Encoding: On input a message $x \in \{0,1\}^\ell$ and an encoding key ek the algorithm Enc outputs an encoded message $c \in \{0,1\}^\ell$ where for each bit $j \in [\ell]$, $c^j = x^j \oplus f_{ek}(j)$.

Key Extraction: On input the encoding key ek, the database of documents t, and index i such that $t_i = c$ the algorithm KeyEx outputs a decoding key dk generated as follows:

1. For all $u \in \{0,1\}$ run
$$(pk_u, sk_u) \leftarrow \mathsf{HE.G}(1^\lambda)$$
and
$$\alpha_u \leftarrow \mathsf{HE.E}_{pk_u}(i) \ .$$
2. For all $j \in [\ell], u \in \{0,1\}$ run[6]
$$\beta_u^j = \mathsf{HE.Eval}_{pk_u}(\mathsf{mux}[t,j], \alpha_u)$$
where the circuit $\mathsf{mux}[t,j](i)$ outputs the j-th bit of the i-th document t_i^j;
3. For all $u \in \{0,1\}$ run
$$ck_u \leftarrow \mathsf{VC.G}(1^\lambda, \ell, 1)$$
and
$$\gamma_u \leftarrow \mathsf{VC.C}_{ck_u}(\beta_u^1, \dots, \beta_u^\ell) \ .$$
4. Pick a random bit $\sigma \in \{0,1\}$.
5. Define the following circuit:
$C[ek, \sigma, sk_\sigma, ck_0, ck_1, \gamma_0, \gamma_1](\beta_0', \beta_1', \pi_0', \pi_1', j)$:

[3]The evaluation algorithm $\mathsf{HE.Eval}$ can always be made deterministic since we do not need circuit privacy.

[4]Note that the decryption key also contains an encryption of i which depends logarithmically on d, but we are going to ignore all logarithmic factors.

[5]This means that we need to use a symmetric encryption scheme where it is possible to recover a single bit of the plaintext from a single bit of the ciphertext. This can easily be done by encrypting x bit by bit using the PRF.

[6]Note that we consider $\mathsf{HE.Eval}$ to be a deterministic algorithm. This can always be achieved by fixing the random tape of $\mathsf{HE.Eval}$ to some constant value.

(a) if$(\forall u \in \{0,1\} : \mathsf{VC.V}_{ck_u}(\gamma_u, j, \beta'_u, \pi'_u))$

$$\text{output } \mathsf{HE.D}_{sk_\sigma}(\beta'_\sigma) \oplus f_{ek}(j);$$

(b) else output $\perp$;

6. Compute an obfuscation $\bar{C} \leftarrow \mathcal{O}(C_\sigma)$ where C_σ is a shorthand for the circuit defined before, padded to length equal to $\max_{\tau,\rho}(C, C'_{\tau,\rho})$ (where the circuits $C'_{\tau,\rho}$ are defined in the proof of security).
7. Output $dk = (pk_0, pk_1, \alpha_0, \alpha_1, ck_0, ck_1, \bar{C})$

Decoding: On input a decoding key dk and a database of document t the algorithm Dec outputs a message x' in the following way:

1. Parse $dk = (pk_0, pk_1, \alpha_0, \alpha_1, ck_0, ck_1, \bar{C})$;
2. For all $j \in [\ell], u \in \{0,1\}$ run
$$\beta^j_u = \mathsf{HE.Eval}_{pk_u}(\mathsf{mux}[t,j], \alpha_u) \ ;$$
3. For all $u \in \{0,1\}$ run
$$\gamma_u \leftarrow \mathsf{VC.C}_{ck_u}(\beta^1_u, \ldots, \beta^\ell_u) \ .$$
4. For all $j \in [\ell], u \in \{0,1\}$ compute
$$\pi^j_u \leftarrow \mathsf{VC.D}_{ck_u}((\beta^1_u, \ldots, \beta^\ell_u), j) \ ;$$
5. For all $j \in [\ell]$ output $(x')^j \leftarrow \bar{C}(\beta^j_0, \beta^j_1, \pi^j_0, \pi^j_1, j)$;

Theorem 1. *If a) f is PRF b)* $(\mathsf{VC.G}, \mathsf{VC.C}, \mathsf{VC.D}, \mathsf{VC.V})$ *is a* somewhere-statistically binding *vector commitment scheme c)* $(\mathsf{HE.G}, \mathsf{HE.E}, \mathsf{HE.D}, \mathsf{HE.Eval})$ *is a* compact *homomorphic encryption scheme and d) $\mathcal{O}$ is an* indistinguishability obfuscator *for all polynomial size circuits then the anonymous steganography scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{KeyEx}, \mathsf{Dec})$ *which satisfies Definitions 1, 2 and has*

$$|dk| = \mathsf{poly}(\lambda \log(d\ell)) \ .$$

Proof. **Correctness (Definition 1).** Correctness follows from inspection of the protocol. In particular, for each bit $j \in [\ell]$ it holds that

$$\bar{C}(\beta^j_0, \beta^j_1, \pi^j_0, \pi^j_1, j) =$$

$$C[ek, \sigma, sk_\sigma, ck_0, ck_1, \gamma_0, \gamma_1](\beta^j_0, \beta^j_1, \pi^j_0, \pi^j_1, j)$$

since the obfuscator is correct. It is also true (since the HE scheme is correct) that $\forall u \in \{0,1\}$ the ciphertext β^j_u is such that

$$\mathsf{HE.D}_{sk_u}(\beta^j_u) = \mathsf{mux}[t,j](\mathsf{HE.D}_{sk_u}(\alpha_u)) = \mathsf{mux}[t,j](i) = t^j_i.$$

Now, since $t^j_i = x^j \oplus f_{ek}(j)$ it follows that the output z of $\bar{C}$ is either $\perp$ or x^j. Finally, the circuit only outputs $\perp$ if $\exists u \in \{0,1\}$ s.t. $\mathsf{VC.V}_{ck_u}(\gamma_u, j, \beta^j_u, \pi^j_u) = 0$. But since

$$ck_u \leftarrow \mathsf{VC.G}(1^\lambda, \ell, 1),$$

$$\gamma_u \leftarrow \mathsf{VC.C}_{ck_u}(\beta^1_u, \ldots, \beta^\ell_u), \pi^j_u \leftarrow \mathsf{VC.D}_{ck_u}((\beta^1_u, \ldots, \beta^\ell_u), j)$$

then the probability that $\bar{C}$ (and therefore Dec) outputs $\perp$ is 0 since the vector commitment is correct.

Anonymity (Definition 2). We prove anonymity using a series of hybrid games. We start with a game which is equivalent to the definition when $b = 0$ and we end with a game which is equivalent to the definition when $b = 1$. We prove at each step that the next hybrid is indistinguishable from the previous. Therefore, at the end we conclude that the adversary cannot distinguish whether $b = 0$ or $b = 1$.

Hybrid 0. This is the same as the definition when $b = 0$. In particular, here it holds that

$$(\alpha_0, \alpha_1) \leftarrow (\mathsf{HE.E}_{pk_0}(i_0), \mathsf{HE.E}_{pk_1}(i_0)).$$

Hybrid 1. In the first hybrid we replace $\alpha_{1-\sigma}$ with

$$\alpha_{1-\sigma} \leftarrow \mathsf{HE.E}_{pk_{1-\sigma}}(i_1)$$

Note that the circuit $C[ek, \sigma, sk_\sigma, ck_0, ck_1, \gamma_0, \gamma_1](\cdot)$ does *not* contain the secret key $sk_{1-\sigma}$, therefore any adversary that can distinguish between Hybrid 0 and 1 can be turned into an adversary which breaks the IND-CPA security of the HE scheme.

Hybrid 2. In the previous hybrids t_{i_1} is a random string from $\{0,1\}^\ell$. In this hybrid we replace t_{i_1} with an encryption of x using a new PRF key ek'. That is, for each bit $j \in [\ell]$ we set $t^j_{i_1} = x^j \oplus f_{ek'}(j)$. Clearly, any adversary that can distinguish between Hybrid 1 and Hybrid 2 can be used to break the PRF.

Hybrid 3.(τ, ρ). We now define a series of 2ℓ hybrids indexed by $\tau \in [\ell], \rho \in \{0,1\}$. In Hybrid 3.$(\tau, \rho)$ we replace the obfuscated circuit with the circuit:

$C'[\tau, \rho, ek, ek', \sigma, sk_0, sk_1, ck_0, ck_1, \gamma_0, \gamma_1](\beta'_0, \beta'_1, \pi'_0, \pi'_1, j)$:

1. if$(\exists u \in \{0,1\} : \mathsf{VC.V}_{ck_u}(\gamma_u, j, \beta'_u, \pi'_u) = 0)$ output $\perp$
2. else if$(j \geq \tau + \rho)$ output $\mathsf{HE.D}_{sk_\sigma}(\beta'_\sigma) \oplus f_{ek}(j)$;
3. else output $\mathsf{HE.D}_{sk_{1-\sigma}}(\beta'_{1-\sigma}) \oplus f_{ek'}(j)$;

We use $C'_{\tau,\rho}$ as a shorthand for a circuit defined as above, and which, if necessary, is padded to make C_σ and all the $C'_{\tau,\rho}$'s equally long.

In addition, we also replace the way the keys for the vector commitment schemes are generated. Remember that in the previous hybrids

$$\forall u \in \{0,1\} \quad ck_u \leftarrow \mathsf{VC.G}(1^\lambda, \ell, 1),$$

which are now replaced with

$$\forall u \in \{0,1\} \quad ck_u \leftarrow \mathsf{VC.G}(1^\lambda, \ell, \tau).$$

We now argue that Hybrid 3.$(1,0)$ is computationally indistinguishable from Hybrid 2 thanks to the $i\mathcal{O}$ property of the obfuscator. This holds since: 1) the keys ck_0, ck_1 are identically distributed and 2) the circuit $C'_{1,0}$ computes the same function as the circuit C obfuscated in Hybrid 2: since j is indexed starting from 1 we always have $j \geq 1 + 0$ and the branch (3) is never taken.

Next, we argue that Hybrid 3.$(\tau, 0)$ is indistinguishable from Hybrid 3.$(\tau, 1)$. First we note that the commitment keys ck_0, ck_1 are identically distributed in these two hybrids i.e., in both hybrids

$$\forall u \in \{0,1\} \quad ck_u \leftarrow \mathsf{VC.G}(1^\lambda, \ell, \tau).$$

The only difference between the two hybrids is what circuits are being obfuscated: in Hybrid 3.$(\tau, 0)$ we obfuscate $C'_{\tau,0}$ and in Hybrid 3.$(\tau, 1)$ we obfuscate $C'_{\tau,1}$. We now argue that these two circuits give the same output on every input, and therefore an adversary that can distinguish between Hybrid 3.$(\tau, 0)$ and Hybrid 3.$(\tau, 1)$ can be used to break the indistinguishability obfuscator.

It follows from inspection that the two circuits behave differently only on inputs of the form $(\beta_0', \beta_1', \pi_0', \pi_1', \tau)$. On input of this form:

- $C'_{\tau,0}$ (since $j = \tau \geq \tau$) chooses branch (2) and outputs

$$x_0^j \leftarrow \mathsf{HE.D}_{sk_\sigma}(\beta'_\sigma) \oplus f_{ek}(j),$$

- $C'_{\tau,1}$ (since $j = \tau \not\geq \tau + 1$) chooses branch (3) and outputs

$$x_1^j \leftarrow \mathsf{HE.D}_{sk_{1-\sigma}}(\beta'_{1-\sigma}) \oplus f_{ek'}(j).$$

Now, the *somewhere-statistically binding* property of the vector commitment scheme allows us to conclude that there exists only one single pair (β_0', β_1') for which $C'_{\tau,0}$ and $C'_{\tau,1}$ do not output $\perp$ (remember that in both hybrids the commitment keys ck_0, ck_1 are statistically binding on index τ), namely the pair

$$\forall u \in \{0,1\} \quad \beta_u^j = \mathsf{HE.Eval}_{pk_u}(\mathsf{mux}[t,\tau], \alpha_u)$$

which decrypts to the pair $(t_{i_0}^j, t_{i_1}^j)$ (since we changed $\alpha_{1-\sigma}$ in Hybrid 1), which in turns were defined as (since we changed $t_{i_1}^j$ in Hybrid 2)

$$(t_{i_0}^j, t_{i_1}^j) = (x^j \oplus f_{ek}(j), x^j \oplus f_{ek'}(j))$$

which implies that $x_0^j = x_1^j$ and therefore the two circuits have the exact same input/output behavior.

Finally, we argue that Hybrid 3.$(\tau, 1)$ is indistinguishable from Hybrid 3.$(\tau+1, 0)$ for all $\tau \in [\ell]$ since by definition the circuits $C'_{\tau,1}$ and $C'_{\tau+1,0}$ are identical and the only difference between these hybrids is in the way the commitment keys ck_0, ck_1 are generated. In particular, the only difference is the index on which the keys are statistically binding. Therefore, any adversary who can distinguish between 3.$(\tau, 1)$ and Hybrid 3.$(\tau + 1, 0)$ can be used to break the *index hiding property* of the vector commitment scheme.

This concludes the technical core of our proof, what is left now is to make few simple changes to go from Hybrid 3.$(\ell, 0)$ to the game from Definition 2 when $b = 1$.

Hybrid 4. In this hybrid we replace the obfuscated circuit with

$$C[ek', \sigma', sk_{\sigma'}, ck_0, ck_1, \gamma_0, \gamma_1](\cdot)$$

where $\sigma' = 1 - \sigma$. It is easy to see that the input/output behavior of this circuit is exactly the same as $C'_{\ell,1}$: since $\forall j \in [\ell] : j \not\geq \ell + 1$ the circuit $C'_{\ell,1}$ always executes branch 3) and therefore an adversary that can distinguish between Hybrid 4 and Hybrid 3.$(\ell, 0)$ can be used to break the indistinguishability obfuscator.

Hybrids 5, 6, 7. In Hybrid 5 we change the distribution of both commitment keys ck_0, ck_1 to $\mathsf{VC.G}(1^\lambda, \ell, 1)$ whereas in Hybrid 4 they were both sampled as $\mathsf{VC.G}(1^\lambda, \ell, \ell)$. Indistinguishability follows from the index hiding property. In Hybrids 6 we replace t_{i_0} with a uniformly random string in $\{0,1\}^\ell$ whereas in the previous hybrid it was an encryption of x using the PRF f with key ek. Since the obfuscated circuit no longer contains ek we can use an adversary which distinguishes between Hybrids 5 and 6 to break the PRF. In Hybrid 7 we replace $\alpha_{1-\sigma'}$ (which in the previous hybrid is an encryption of i_0) with an encryption of i_1. Since the obfuscated circuit no longer contains $sk_{1-\sigma'} = sk_\sigma$ we can use an adversary which distinguishes between Hybrids 6 and 7 to break the IND-CPA property of the encryption scheme. Now Hybrid 7 is exactly as the definition of anonymity with $b = 1$ with a random bit $\sigma' = 1 - \sigma$ (which is distributed uniformly at random) and a random encoding key ek'. This concludes therefore the proof. □

Our theorem, together with the results of [HW15] implies the following.

Corollary 1. *Assuming the existence of homomorphic encryption and indistinguishability obfuscators for all polynomially sized circuits, there exists an anonymous steganography scheme.*

4. DEALING WITH GENERAL DISTRIBUTIONS

In our study we have made the simplifying assumption that the honestly generated documents are sampled independently from the uniform distribution, and we have shown under this constraint that it is possible to construct an anonymous steganography scheme. However, real media (such as video, pictures, etc.) is clearly not generated according to this distribution. We briefly review here "regular" steganography as defined by [HAL09] and discuss how it can be combined with our scheme in order to obtain anonymous steganography for realistic distributions.

Distribution of honest users. In [HAL09] the "innocent looking" traffic is modelled using a distribution $\mathcal{D}$ which samples a sequence of documents in $\{0,1\}^\ell$. Crucially, samples are not independent and we denote with $\mathcal{D}_h$, the marginal distribution of a single document conditioned on a history $h = (t_1, \ldots, t_j)$.

Steganography for $\mathcal{D}$. In [HAL09] a steganographic scheme is defined as a tuple of algorithms $(\mathsf{StG}, \mathsf{StE}, \mathsf{StD})$ such that:

- The generation algorithm $k \leftarrow \mathsf{StG}(1^\lambda, \mathcal{D})$ on input the security parameter λ and the distribution $\mathcal{D}$ samples a key k;
- The encoding algorithm $c \leftarrow \mathsf{StE}(k, h, x)$ embeds a message x in an innocent looking document c given a history of documents $h = (t_1, \ldots, t_j)$.
- The decoding algorithm $x' \leftarrow \mathsf{StD}(k, c)$ recovers a message x' using the key k and the document c;

As usual we require *correctness* (i.e., $x' = x$ except with negligible probability) and *security* with the following game:

Definition 3 (Chosen Hiddentext Attack)**.** *We define a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:*

1. *The challenger $\mathcal{C}$ generates a key $k \leftarrow \mathsf{StG}(1^\lambda, \mathcal{D})$ and samples a bit $b \leftarrow \{0,1\}$;*
2. *The adversary $\mathcal{A}$ asks polynomially many queries of the form (h, x) with $h = (t_1, \ldots, t_j)$;*
3. *The challenger $\mathcal{C}$ returns:*
   (a) *$c \leftarrow \mathcal{D}_h$ if $b = 0$;*
   (b) *$c \leftarrow \mathsf{StE}(k, h, x)$ if $b = 1$;*

4. *$\mathcal{A}$ outputs a guess bit g;*

We say π satisfies chosen hiddentext attack *if for all PPT $\mathcal{A}$ $\left|\Pr[g = b] - \frac{1}{2}\right| = \mathsf{negl}(\lambda)$.*

In [HAL09] several constructions provably secure steganography schemes are provided. In particular we are going to use the scheme MultiBit which allows to embed ℓ' bit long messages x into $\ell = \lambda \cdot \ell'$ long documents c, and has the property that each bit of x can be recovered looking only at λ bits of c.

Anonymous Steganography for $\mathcal{D}$. We slightly tweak the definition of anonymous steganography given in Section 2 to handle the distribution $\mathcal{D}$ and the history h. In particular, the interface of Gen and Enc are modified as follows (while KeyEx, Dec are left unchanged):

- $ek \leftarrow \mathsf{Gen}(1^\lambda, \mathcal{D})$ is a randomized algorithm which generates an encoding key ek on input a distribution $\mathcal{D}$.
- $c \leftarrow \mathsf{Enc}_{ek}(h, x)$ is a randomized algorithm which encodes a secret message $x \in \{0,1\}^{\ell'}$ into a (random looking) document $c \in \{0,1\}^{\ell}$, given a history $h = (t_1, \ldots, t_j)$.

The new definition of correctness is straightforward. Anonymity is now defined as

Definition 4 (Anonymity)**.** *We define a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:*

1. *The challenger $\mathcal{C}$ generates a key $ek \leftarrow \mathsf{Gen}(1^\lambda, \mathcal{D})$;*
2. *The adversary $\mathcal{A}$ outputs a message $x \in \{0,1\}^{\ell'}$, and index i_0, and a set of documents $(t_1, \ldots, t_{i_0 - 1})$;*
3. *The challenger $\mathcal{C}$ defines $h_0 = (t_1, \ldots, t_{i_0-1})$ and:*
   (a) *if $b = 0$ computes and outputs $t_{i_0} \leftarrow \mathsf{Enc}_{ek}(h_0, x)$;*
   (b) *if $b = 1$ computes and outputs $t_{i_0} \leftarrow \mathcal{D}_{h_0}$;*
4. *The adversary $\mathcal{A}$ outputs an index i_1 and a set of documents $(t_{i_0+1}, \ldots, t_{i_1-1})$;*
5. *The challenger $\mathcal{C}$ defines $h_1 = (t_1, \ldots, t_{i_1-1})$ and:*
   (a) *if $b = 1$ computes and outputs $t_{i_1} \leftarrow \mathsf{Enc}_{ek}(h_1, x)$;*
   (b) *if $b = 0$ computes and outputs $t_{i_0} \leftarrow \mathcal{D}_{h_1}$;*
6. *The adversary $\mathcal{A}$ outputs a set of documents $(t_{i_1}, \ldots, t_d)$;*
7. *The challenger $\mathcal{C}$ computes and outputs*
$$dk \leftarrow \mathsf{KeyEx}_{ek}\left((t_1, \ldots, t_d), i_b\right) \ ;$$
8. *$\mathcal{A}$ outputs a guess bit g;*

We say π satisfies anonymity *if for all PPT $\mathcal{A}$ $\left|\Pr[g = b] - \frac{1}{2}\right| = \mathsf{negl}(\lambda)$.*

To achieve this definition we only need a few simple changes to the construction from Section 3, highlighted here:

Key Generation: sample $ek \leftarrow \mathsf{StG}(1^\lambda, \mathcal{D})$ (instead of sampling a PRF key);

Encoding: compute $c \leftarrow \mathsf{StE}(k, h, x)$ (instead of using the PRF);

Key Extraction: In Step 5.(a) (the construction of the circuit), let the circuit decode using $\mathsf{StD}(k, \mathsf{HE.D}_{sk_\sigma}(\beta'_\sigma))$ (instead of using the PRF);

Decoding: Unchanged.

Correctness of the modified scheme follows from correctness of the original construction and of $(\mathsf{StG}, \mathsf{StE}, \mathsf{StD})$. The proof that the modified construction satisfies Definition 4 trivially follows from the original proof, where in Hybrid 2 and 6 we rely on the *chosen hiddentext attack* instead of using the PRF f.

5. LOWER BOUND

In this section we show that no (correct) anonymous steganography scheme can have a decoding key of size $O(\log(\lambda))$. Since the decoding key must be sent over an anonymous channel, this gives a lower bound on the number of bits which are necessary to bootstrap anonymous communication.

To show this, we find a strategy for Joe that gives him a higher probability of guessing the leaker than if he guessed uniformly at random.

Our lower bound applies to a more general class of anonymous steganography schemes than defined earlier, in particular it also applies to *reactive* schemes where the leaker can post multiple documents to the website, as a function of the documents posted by other users. We define a *reactive anonymous steganography* scheme as a tuple of algorithms $\pi = (\mathsf{Enc}, \mathsf{KeyEx}, \mathsf{Dec})$ where:

- $(t_k, state_j) \leftarrow \mathsf{Enc}_{ek}(x, t^{k-1}, state_{j-1})$ is an algorithm which takes as input a message $x \in \{0,1\}^{\ell'}$, a sequence of documents t^{k-1} (which represents the set of documents previously sent) and a state of the leaker, and outputs a new document $t_k \in \{0,1\}^{\ell}$, together with a new state.
- $dk \leftarrow \mathsf{KeyEx}_{ek}(t^d, state)$ is an algorithm which takes as input a transcript of all documents sent and the current state of the leaker and outputs a decryption key $dk \in \{0,1\}^s$.
- $x' = \mathsf{Dec}_{dk}(t^d)$ is an algorithm that given transcript t^d returns a guess x of what the secret is in a deterministic way.

To use a reactive anonymous steganography scheme, the leaker's index i is chosen uniformly at random from $\{1, \ldots, n\}$ where n is the number of players. For each k from 1 to d we generate a document t_k. If $k \not\equiv i \mod n$ we let $t_k \leftarrow \{0,1\}^{\ell}$. This corresponds to the non-leakers sending a message. When $k \equiv i \mod n$ we define $(t_k, state_j) \leftarrow \mathsf{Enc}_{ek}(x, t^{k-1}, state_{j-1})$, where $t^{k-1} = (t_1, \ldots, t_{k-1})$. Then we define $dk \leftarrow \mathsf{KeyEx}_{ek}(t^d, state)$ and $x' = \mathsf{Dec}_{dk}(t^d)$. Here dk is the message that Lea would send over the small anonymous channel.[7]

The definition of q-correctness for reactive schemes is the same as for standard schemes, but our definition of anonymity is weaker because we do not allow the adversary to choose

[7]Note that a "standard" anonymous steganography scheme can easily be turned into an reactive anonymous steganography scheme by combining Gen and Enc into one algorithm and storing ek in the *state*.

the documents for the honest users. This implies that our lower bound is stronger.

Definition 5 (Correctness). *A reactive anonymous steganography scheme is q-*correct *if for all λ and $x \in \{0,1\}^{\ell'(\lambda)}$ we have*

$$\Pr\left[\mathsf{Dec}_{dk}\left(t^d\right) = x\right] \geq q.$$

where t and dk is chosen as above and the probability is taken over all the random coin flips.

Definition 6 (Weak Anonymity). *Consider the following game between an adversary A and a challenger C*

1. *The adversary A outputs a message $x \in \{0,1\}^{\ell'}$;*
2. *The challenger C samples random $i \in [n]$, and generates t^d, dk as described above*
3. *The challenger C outputs t^d, dk*
4. *A outputs a guess g;*

We say an adversary has advantage $\epsilon(\lambda)$ *if* $\left|\Pr[g = i] - \frac{1}{n}\right| \geq \epsilon(\lambda)$. *We say a reactive anonymous steganography scheme provides* anonymity *if, for any adversary, the advantage is negligible.*

In the model we assume that the non-leakers' documents are chosen uniformly at random. This is realistic in the case where we use steganography, so that each t_k is the result of extracting information from a larger file. We could also define a more general model where the distribution of each non-leaker's documents t_k depends on the previous transcript. The proof of our impossibility results works as long as the adversary can sample from $T_k|_{T^{k-1}=t^{k-1}, i \not\equiv k \mod n}$ in polynomial time. Using this general model, we can also model the more realistic situation where the players do not take turns in sending documents, but at each step only send a document with some small probability. To do this, we just consider "no document" to be a possible value of t_k.

We could also generalise the model to let the leaker use the anonymous channel at any time, not just after all the documents have been sent. However, in such a model, the anonymous channel transmits more information than just the number of bits send over the channel: the times at which the bits are sent can be used to transmit information [IW10]. For the number of bits sent to be a fair measure of how much information is transferred over the channel, we should only allow the leaker to use the channel when Joe knows she would use the anonymous channel[8], and the leaker should only be allowed to send messages from a prefix-free code (which might depend on the transcript, but should be computable in polynomial time for Joe). Our impossibility result also holds for this more general model, however, to keep the notation simple, we will assume that the anonymous channel is only used at the end.

Finally, we could generalise the model by allowing access to public randomness. However, this does not help the players: as none of the players are controlled by the adversary, the players can generate trusted randomness themselves.

We let $T' = (T'_1, \ldots, T'_d)$ denote the random variable where each T'_i is uniformly distributed on $\{0,1\}^\ell$. In particular $T'|_{T'^k=t^k}$ is the distribution the transcript would follow if the first k documents are given by t^k and all the players were non-leakers. We let dk' be uniformly distributed on $\{0,1\}^s$. Joe can sample from both $T'|_{T'^k=t^k}$ and dk' and he can compute Dec. His strategy to guess the leaker, given a transcript t, will be to estimate $\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^k = t^k)$ for each $k \leq d$. That is, given that the transcript of the first k documents is t^k and all later documents are chosen as if the sender was not a leaker and the anonymous channel just sends random bits, what is the probability that the result is x? He can estimate this by sampling: given t^k he randomly generates t^d and dk, and then he computes Dec of this extended transcript.

If we assume that the protocol π is symmetric[9] in the messages x, then before any documents are sent, we have $\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^0 = t^0) = 2^{-\ell'}$. Assuming that after all the documents are sent, there exists a key $dk' \in \{0,1\}^s$ such that $\mathsf{Dec}_{dk'}(t^d) = x$, then for a random dk' we must have $\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^d = t^d) \geq 2^{-s}$. As $s < \ell'$ the documents in t^d must have increased the probability of decoding to x. The non-leakers' documents affect this probability, but *in expectation* they do not, so in most cases most of this increase will have to come from the leaker, that is, these probabilities would tend to be higher just after the leaker's documents than just before. Of course, a leaking player might send some documents that lowers $\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^k = t^k)$ to confuse Joe, so we need a way to add up all the changes a players does to $\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^k = t^k)$. The simplest idea would be to compute the additive difference

$$\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^k = t^k) \\ - \Pr(\mathsf{Dec}_{dk'}(T') = x | T'^{k-1} = t^{k-1})$$

and add these for each player. However, the following example shows that this strategy does not work in general.

Example 1. *Consider this protocol for two players, where one of them wants to leak one bit. We have $s = 0$, that is dk is the empty sting and will be omitted from the notation. First we define the function* Dec. *This function looks at the two first documents. If none of these are 0^ℓ, it returns the first bit of the third document. Otherwise it defines the* leader *to be the first player who send 0^ℓ. Next* Dec *looks at the first time the leader sent a document different from 0^ℓ. If this number represents a binary number less than $\frac{9}{10} \cdot 2^\ell$, then* Dec *returns the last bit of the document before, otherwise it outputs the opposite value of that bit. If the leader only sends the document 0^ℓ the output of* Dec *is just the last bit sent by the other player.*

The leaker's strategy is to become the leader. There is extremely small probability that the non-leaker sends 0^ℓ in his first document, so we will ignore this case. Otherwise the leaker sends 0^ℓ in her first document and becomes the leader. When sending her next document, she looks at the last document from the non-leaker. If it ended in 0, Joe will think there is 90% chance that 0 it is output and 10% chance that the output will be 1, and if it ended in 1 it is the other way around. If the last bit in the non-leakers document is

[8] That is, there should be a polynomial time algorithm that given previous transcript t^k and previous messages over the anonymous channel decides if the leaker sends a message over the anonymous channel.

[9] By this we mean that for random transcript T' and random dk' the result $\mathsf{Dec}_{dk'}(T')$ is uniformly distributed. In the formal proof we will show why we can make this assumption.

the bit the leakers wants to leak, she just sends the document $0^{\ell-1}1$. To Joe, this will look like the non-leaker raised the probability of this outcome from 50% to 90% and then the leaker raised it to 100%. Thus, Joe will guess that the non-leaker was the leaker.

If the last bit of the previous document was the opposite of what the leaker wanted to reveal, she will "reset" by sending 0^{ℓ}. This brings Joe's estimate that the result will be 1 back to 50%. The leaker will continue "resetting" until the non-leaker have sent a document ending in the correct bit more times than he has sent a document ending in the wrong bit. For sufficiently high d, this will happen with high probability, and then the leaker sends $0^{\ell}1$. This ensures that $\mathsf{Dec}(T)$ gives the correct value and that Joe will guess that the non-leaker was the leaker.

If the leaker wants to send many bits, the players can just repeat this protocol.

Obviously, the above protocol for revealing information is not a good protocol: it should be clear to Joe that the leader is not sending random documents.

As the additive difference does not work, Joe will instead look at the multiplicative factor

$$\frac{\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^k = t^k)}{\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^{k-1} = t^{k-1})}.$$

Definition 7. *For a transcript t the* multiplicative factor *$mf_{j,[k_0,k_1]}$ of player j over the time interval $[k_0,k_1]$ is given by*

$$mf_{j,[k_0,k_1]}(t) = \prod_{k\in[k_0,k_1]\cap(j+n\mathbb{N})} \frac{\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^k = t^k))}{\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^{k-1} = t^{k-1})}.$$

We also define

$$mf_{-j,[k_0,k_1]}(t) = \prod_{k\in[k_0,k_1]\setminus(j+n\mathbb{N})} \frac{\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^k = t^k)}{\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^{k-1} = t^{k-1})}.$$

If we use the multiplicative factor on the non-leaker in the protocol in Example 1 we see that for each document sent by the non-leaker there is probability 0.5 that his multiplicative factor increases by a factor 1.8 and probability 0.5 that it is multiplied by a factor 0.2. Thus, if the non-leaker first sends a document which decrease $\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^k = t^k)$ from 0.5 to 0.1 and later a document that increases it from 0.5 to 0.9, the two document no longer chancel each other out: they result in multiplying the multiplicative factor by 0.36.

For fixed k_0 and non-leaking player j the sequence

$$mf_{j,[k_0,k_0]}(T), mf_{j,[k_0,k_0+1]}(T), \dots$$

is a martingale. Furthermore, if we consider the first k_1-2 documents to be fixed and player 1 sends a document at time k_1-1 and player 2 at time k_1, then player 1's document can affect the distribution of

$$mf_{2,[k_0,k_1]}(T')|_{T'^{k_1-1}=t^{k_1-1}}$$

but no matter what document t_{k_1-1} player 1 sends,

$$mf_{2,[k_0,k_1]}(T')|_{T'^{k_1-1}=t^{k_1-1}}$$

will have expectation

$$mf_{2,[k_0,k_1-1]}(t^{k_1-1}).$$

Similar statements holds for the sum of additive differences, but the advantage of the multiplicative factor is that it is non-negative. For example, as the multiplicative factor starts at 1 there is probability at most 0.1 that it will ever be at least 10. Thus, while the leaker's multiplicative factor has to be large in most cases, all the non-leakers will with high probability have small multiplicative factors. The same does not hold for the sum of additive differences, because as Example 1 shows, you can have a probability arbitrarily close to 1 that a non-leaker's sum of additive differences increases to 0.4 (or any other positive number) as long as there is a small probability that it decreases to negative values of large absolute value.

Proposition 2. *For j and k_0, k_1 we have:*

$$\mathbb{E}_{T'|T^{k_1-1}=t^{k_1-1}} mf_{j,[k_0,k_1]}(T) = mf_{j,[k_0,k_1-1]}(t^{k_1-1})$$

Proof. For $k_1 \not\equiv j \mod n$ we have

$$mf_{j,[k_0,k_1]}(t) = mf_{j,[k_0,k_1-1]}(t^{k_1-1})$$

for any t so the statement is trivially true. For $k_1 \equiv j \mod n$ it follows from Bayes' Theorem. □

Proposition 3. *For fixed $m_0 > 2$ and x and random T there is probability at most $\frac{4d}{m_0}$ that there exists $j \neq i$ and k_0 such that $mf_{j,[k_0,d]}(T)$ or $mf_{-i,[k_0,d]}(T)$ is at least $\frac{m_0}{2}$.*

Proof. For a fixed value of k_0, and a non-leaker j we have $\mathbb{E}\left(mf_{j,[k_0,d]}(T)\right) = 1$. As

$$mf_{j,[k_0,d]}(t) \geq 0$$

this implies that

$$\Pr\left(mf_{j,[k_0,d]}(T) \geq \frac{m_0}{2}\Big|T\right) \leq \frac{2}{m_0}.$$

Similarly for $mf_{-i,[k_0,d]}$. If player j does not send the k_0'th document we have

$$mf_{j,[k_0,d]}(t) = mf_{j,[k_0-1,d]}(t),$$

so for fixed t there are only d different values (not counting 1) of $mf_{j,[k_0,d]}(t)$ with $j \neq i$ and $k_0 \leq d$. By the union bound, the probability that one of the $mf_{j,[k_0,d]}(t)$'s or one of the $mf_{-i,[k_0,d]}(t)$'s are above $\frac{m_0}{2}$ is at most $\frac{4d}{m_0}$. □

For fixed value of k Joe can estimate $\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^k = t^k)$ with a small *additive* error, by sampling $T'^d|_{T'^k=t^k}$ and dk'. However, when the probability is small, there might still be a large *multiplicative* error. Joe can only do polynomially many samples, so when $\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^k = t^k)$ is less than polynomially small Joe will most likely estimate it to be 0.[10] Instead, the idea is to estimate the multiplicative factor starting from some time k_0 such that $\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^k = t^k)$ is not too small for any $k \geq k_0$. The following proposition is useful when choosing k_0 and choosing how many samples we make.

[10] This is the reason that anonymous steganography with small anonymous channel works at all: we keep $\Pr(\mathsf{Dec}_{dk'}(T') = x | T'^k = t^k)$ exponentially small until Lea uses the anonymous channel. When Lea then uses the anonymous channel to send dk, the probability of x being the output increases from exponentially small to 1.

Definition 8. *In the following we say that Joe's estimate of* $\Pr(\mathsf{Dec}_{dk'}(T') = x|T'^k = t^k)$ *is* bad *if* $\Pr(\mathsf{Dec}_{dk'}(T') = x|T'^k = t^k) \geq \frac{\epsilon^2}{2^{s+7}d^2}$ *but his estimate is not in the interval*

$$\left[\left(1-\frac{1}{2d}\right)\Pr(\mathsf{Dec}_{dk'}(T') = x|T'^k = t^k)\right.$$
$$\left., \left(1+\frac{1}{2d}\right)\Pr(\mathsf{Dec}_{dk'}(T') = x|T'^k = t^k)\right].$$

Proposition 4. *Assume that Joe makes* $\frac{3\cdot 2^{s+9}d^4}{\epsilon^2}\log\left(\frac{4d}{\epsilon}\right)$ *samples of* $\mathsf{Dec}_{dk'}(T')|_{T'^k=t^k}$ *to estimate* $\Pr(\mathsf{Dec}_{dk'}(T') = x|T'^k = t^k)$.

No matter the true value of $\Pr(\mathsf{Dec}_{dk'}(T') = x|T'^k = t^k) \geq \frac{\epsilon^2}{2^{s+7}d^2}$*, there is probability at most* $\frac{\epsilon}{2d}$ *that his estimate is bad.*

Proof. Follows from the multiplicative Chernoff bound. □

Now we are ready to prove the impossibility result.

Theorem 5. *Let* ϵ *be a function in* λ *such that* $\frac{1}{\epsilon}$ *is bounded by a polynomial, and let* π *be a* $q(\lambda)$*-correct reactive anonymous steganography scheme with* $s(\lambda) = O(\log(\lambda))$, $\ell' \geq s + 7 + 2\log_2(d) - 2\log_2(\epsilon)$. *Now there is a probabilistic polynomial time Turing machine* A *that takes input* t *and* x *and outputs the leaker identity with probability*

$$q(\lambda) + \frac{1-q(\lambda)}{n(\lambda)} - \epsilon(\lambda)$$

Notice that we cannot do better than $q + \frac{1-q}{n}$. The players could use a protocol where with probability q the leaker reveals herself and the information and otherwise no-one reveals any information. This protocol succeeds with probability q, and when is does, Joe will guess the leaker. With probability $1-q$ it does not succeed, and Joe has probability $\frac{1}{n}$ of guessing the leaker. In total Joe will guess the leaker with probability $q + \frac{1-q}{n}$.

Proof. Let π be a reactive anonymous steganography scheme. We assume that for random T' and dk' the random variable $\mathsf{Dec}_{dk'}(T')$ is uniformly distributed[11] on $\{0,1\}^{\ell'}$ and we will just let Joe send $0^{\ell'}$ in the anonymity game.

Let $m_0 = \frac{8d}{\epsilon}$. Consider a random transcript t. If for some k_0 and some non-leaker j we have $mf_{j,[k_0,d]} \geq \frac{m_0}{2}$ or $mf_{-i,[k_0,d]} \geq \frac{m_0}{2}$ we set $E = 1$.

First Joe will estimate $\Pr(\mathsf{Dec}_{dk'}(T') = 0^{\ell'}|T'^k = t^k)$ for all k using

$$\frac{3\cdot 2^{s+9}d^4}{\epsilon^2}\log\left(\frac{4d}{\epsilon}\right)$$

samples for each k. Set $E = 1$ if at least one of these estimates is bad. In all cases where E has not been defined yet we set $E = 0$. By the above propositions and the union bound, $\Pr(E = 1) \leq \epsilon(\lambda)$.

[11]If this is not the case, we can define a reactive anonymous scheme $\widetilde{\pi}$ where this is the case: just let X' be uniformly distributed on $\{0,1\}^{\ell'}$, let $\widetilde{\mathsf{Enc}}(x, t^k, state) = \mathsf{Enc}(x \oplus X', t^k, state)$ and $\widetilde{\mathsf{Dec}}_{dk}(t) = X' \oplus \mathsf{Dec}_{dk}(t)$, where $\oplus$ is bitwise addition modulo 2. To use $\widetilde{\pi}$ we would need ℓ' bits of public randomness to give us X'. To get this, we can just increase ℓ by ℓ' and let X' be the last ℓ' bits of the first document.

Now let k_0 be the smallest number such that for all $k \geq k_0$ Joe's estimate of $\Pr(\mathsf{Dec}_{dk'}(T') = 0^{\ell'}|T'^k = t^k)$ is at least $\frac{\epsilon^2}{2^{s+7}d^2}$. The idea would be to estimate the multiplication factors $mf_{j,[k_0+1,d]}$. However, the problem is that $\Pr(\mathsf{Dec}_{dk'}(T') = 0^{\ell'}|T'^{k_0} = t^{k_0})$ could be large (even 1) even though $\Pr(\mathsf{Dec}_{dk'}(T') = 0^{\ell'}|T'^{k_0-1} = t^{k_0-1})$ is small, so the players might not reveal any information after the $k_0 - 1$'th document. Thus, Joe needs to include the $k_0 - 1$'th document in his estimate of the multiplication factors, but his estimate of $\Pr(\mathsf{Dec}_{dk'}(T') = 0^{\ell'}|T'^{k_0-1} = t^{k_0-1})$ might be off by a large constant factor. To solve this problem, we define

$$mf_j = \begin{cases} mf_{j,[k_0+1,d]}, & \text{if } j \not\equiv_n k_0 - 1 \mod n \\ mf_{j,[k_0+1,d]}\dfrac{\Pr(\mathsf{Dec}_{dk'}(T')=0^{\ell'}|T'^{k_0}=t^{k_0})}{(1-\frac{1}{2d})^{-1}\frac{\epsilon^2}{2^{s+7}d^2}}, & \text{if } j \equiv_n k_0 - 1 \mod n \end{cases}$$

that is, we pretend that $\Pr(\mathsf{Dec}_{dk'}(T') = 0^{\ell'}|T'^{k_0} = t^{k_0}) = (1-\frac{1}{2d})^{-1}\frac{\epsilon^2}{2^{s+7}d^2}$ and then use $mf_{j,[k_0,d]}$. We define mf_{-i} the similar way. Joe's estimate of $\Pr(\mathsf{Dec}(T) = X|T'^{k_0-1} = t^{k_0-1})$ less that $\frac{\epsilon^2}{2^{s+7}d^2}$, otherwise k_0 would have been lower (here we are using the assumption $h \geq s + 7 + 2\log_2(d) - 2\log_2(\epsilon)$. Without this, k_0 could be 1). Thus, if this estimate it not bad we must have

$$\Pr(\mathsf{Dec}_{dk'}(T') = 0^{\ell'}|T'^{k_0-1} = t^{k_0-1}) \leq \left(1-\frac{1}{2d}\right)^{-1}\frac{\epsilon^2}{2^{s+7}d^2}$$

So if $E = 0$ then $mf_j \leq mf_{j,[k_0,d]} \leq \frac{m_0}{2}$. Similar for mf_{-i}.

If $E = 0$ then $mf_j \leq \frac{m_0}{2}$ for all $j \neq i$ and $mf_{-i} \leq \frac{m_0}{2}$. Furthermore, as all of Joe's estimate are good, his estimate of mf_j is off by at most a factor $\left(1-\frac{1}{2d}\right)^{-d} < 2$. Now we define Joe's guess: if exactly one of his estimated mf_j's are above m_0 he guesses that this player j is the leaker. Otherwise he chooses his guess uniformly at random from all the players. There are two ways $\Pr(\mathsf{Dec}_{dk'}(T') = 0^{\ell'}|T'^k = t^k)$ can increase as k increases[12]: by the leaker sending documents or by a non-leaker sending documents. In the cases where $E = 0$ and Joe's estimate of mf_i is less than m_0 we know that the contribution from the leaker's documents is a factor less than $2m_0$. As $E = 0$ we also know that the total contribution from all the non-leakers is at most a factor $\frac{m_0}{2}$. So when only dk' has not been revealed to Joe we have

$$\begin{aligned}\Pr(\mathsf{Dec}_{dk'}(T) = X|T = t^d) <& \frac{\epsilon^2}{2^{s+7}d^2}2m_0\frac{m_0}{2} \\ \leq& \frac{\epsilon^2}{2^{s+6}d^2}m_0^2 \\ =& 2^{-s}\end{aligned}$$

[12]If we allow the leaker to send anonymous bits before the end of the open communication, this is a third way $\Pr(\mathsf{Dec}_{dk'}(T') = 0^{\ell'}|T'^k = t^k)$ can increase. However, if the times where the anonymous channel is used are predictable by Joe, he can still sample as if the anonymous bits where random. This way, each anonymous bits makes $\Pr(\mathsf{Dec}_{dk'}(T') = 0^{\ell'}|T'^k = t^k)$ increase by at most a factor 2. If the leaker can only send s anonymous bit in total this only moves a factor 2^s increase in $\Pr(\mathsf{Dec}_{dk'}(T') = 0^{\ell'}|T'^k = t^k)$ from a later point in the proof to here.

As the only randomness left to be revealed[13] is dk' which is uniformly distributed on a set of size 2^{-s}, we know that

$$\Pr(\mathsf{Dec}_{dk'}(T) = 0^{\ell'} | T = t^d)$$

is a multiple of 2^{-s}. This implies

$$\Pr(\mathsf{Dec}_{dk'}(T) = 0^{\ell'} | T = t^d) = 0$$

In other words, if $\mathsf{Dec}_{dk}(T) = 0^{\ell'}$ and $E = 0$ then A must output i. Furthermore, in all other cases where $E = 0$ Joe will either guess the leaker correctly (because Joe's estimate of mf_i is sufficiently high) or guess uniformly among all the players. The probability that Joe is correct is now

$$\begin{aligned}
\Pr(g = i) \geq& \Pr(\mathsf{Dec}_{dk}(T) = 0^{\ell'}, E = 0) \\
&+ \frac{\Pr(\mathsf{Dec}_{dk}(T) \neq 0^{\ell'}, E = 0)}{n} \\
=& \Pr(\mathsf{Dec}_{dk}(T) = 0^{\ell'}) - \Pr(\mathsf{Dec}_{dk}(T) = 0^{\ell'}, E = 1) \\
&+ \frac{\Pr(\mathsf{Dec}_{dk}(T) \neq 0^{\ell'})}{n} \\
&- \frac{\Pr(\mathsf{Dec}_{dk}(T) \neq 0^{\ell'}, E = 1)}{n} \\
\geq& q + \frac{1-q}{n} - \Pr(E = 1) \geq q + \frac{1-q}{n} - \epsilon. \qquad \square
\end{aligned}$$

Finally we can conclude that:

Corollary 2. *If π is a reactive anonymous steganography scheme with $s = O(\log(\lambda))$, d polynomial in λ and $\frac{\ell'}{\log(\lambda)} \to \infty$ that ensures weak anonymity, then the probability of correctness q tends to 0 as $\lambda \to \infty$.*

Proof. Let π be as in the assumption and define

$$\epsilon = \max\left(\lambda^{-1}, 2^{-\frac{s+7+2\log_2(d)-\ell'}{2}}\right)$$

By assumption, $s = O(\log(\lambda))$, $\log(d) = O(\log(\lambda))$, and $\frac{\ell'}{\log(\lambda)} \to \infty$, so $\epsilon \to 0$. The parameters satisfy the assumptions in Theorem 5 so there is an adversary that can guess the leaker with probability

$$q + \frac{1-q}{n} - \epsilon = \frac{1}{n} + \frac{n-1}{n}q - \epsilon \geq \frac{1}{n} + \frac{q(n-1) - n\epsilon}{n}.$$

As π ensures anonymity, $\frac{q(n-1)-n\epsilon}{n}$ must be negligible and as $\epsilon \to 0$ we must have $q \to 0$. $\square$

Acknowledgements. This research was supported by the Danish National Research Foundation and The National Science Foundation of China (grant 61361136003) for the Sino-Danish Center for the Theory of Interactive Computation and from the Center for Research in Foundations of Electronic Markets (CFEM) and COST Action IC1306.

[13]Here we are using that Dec is deterministic. However, allowing it to be non-deterministic does not help: we could just increase ℓ and let Dec use the extra bits in each document as randomness instead of using a random tape.

6. REFERENCES

[AH04] Luis von Ahn and Nicholas J. Hopper. Public-key steganography. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 323–341, 2004.

[BC05] Michael Backes and Christian Cachin. Public-key steganography with active attacks. In *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings*, pages 210–226, 2005.

[BJSW14] Joshua Brody, Sune Jakobsen, Dominik Scheder, and Peter Winkler. Cryptogenography. In *ITCS*, 2014.

[DMS04] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[FNP14] Nelly Fazio, Antonio Nicolosi, and Irippuge Milinda Perera. Broadcast steganography. In *Topics in Cryptology - CT-RSA 2014 - The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25-28, 2014. Proceedings*, pages 64–84, 2014.

[Fri09] Jessica Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 40–49, 2013.

[HAL09] Nicholas J. Hopper, Luis von Ahn, and John Langford. Provably secure steganography. *IEEE Trans. Computers*, 58(5):662–676, 2009.

[HLA02] Nicholas J. Hopper, John Langford, and Luis von Ahn. Provably secure steganography. In *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, pages 77–92, 2002.

[HW15] Pavel Hubáček and Daniel Wichs. On the communication complexity of secure function evaluation with long output. In *ITCS*, 2015.

[IKOS06] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography from Anonymity. *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)-Volume 00*, pages 239–248, October 2006.

[IKV13] Luca Invernizzi, Christopher Kruegel, and Giovanni Vigna. Message in a bottle: Sailing past censorship. In *Proceedings of the Annual*

Computer Security Applications Conference (ACSAC). ACM, 2013.

[IW10] Russell Impagliazzo and Ryan Williams. Communication complexity with synchronized clocks. In *Computational Complexity (CCC), 2010 IEEE 25th Annual Conference on*, pages 259–269. IEEE, 2010.

[Jak14] Sune K. Jakobsen. Information theoretical cryptogenography. In *ICALP (1)*, pages 676–688, 2014.

[KLW15] Venkata Koppula, Allison Bishop Lewko, and Brent Waters. Indistinguishability obfuscation for turing machines with unbounded memory. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 419–428, 2015.

[WL12] Philipp Winter and Stefan Lindskog. How the Great Firewall of China is blocking Tor. In *Proceedings of the USENIX Workshop on Free and Open Communications on the Internet (FOCI 2012)*, August 2012.