Stegogames

Clark Thomborson[1](0000-0002-4147-7898) and Marc Jeanmougin[2]

[1] Computer Science Department
University of Auckland, New Zealand
cthombor@cs.auckland.ac.nz
[2] GBA, Conservatoire National des Arts et Métiers
Paris, France
marc.jeanmougin@cnam.fr

Abstract. We explore the power of steganographic computation in an game-theoretic setting, where n stegocommunicants are attempting to complete a shared computation, and where a well-resourced censor is attempting to prevent the computation. For example, when collaboratively discovering the minimum value ($\min_i x_i$) in a public n-vector X, each stegocommunicant reads a randomly-selected element during each timestep. Each then transmits the index i of the smallest value they have seen to a randomly-selected collaborator. We prove that most stegocommunicants will learn the minimum value in $O(\log n)$ time, w.h.p., if at most 10% of their population is censored in any timestep. The censor in our model retains a copy of all intercepted messages, using this information to optimally select the targets of their censorship at the beginning of each timestep. Our model of stegocomputation is relevant to stegosystems in which: 1) the stegoencoding is determined by the address of the recipient, 2) the censor does not have sufficient computational resource to stegodecode more than a fixed fraction (nominally 10%) of the messages in flight, and 3) the censor cannot store any messages other than the ones it has stegodecoded.

Keywords: steganography, communication protocols, EREW PRAM

1 Introduction

Stegocommunication is similar to encrypted communication, because both involve the transmission and reception of messages under adversarial conditions.

Stegocommunication is distinguished from encrypted communication, because the former avoids revealing that messages are being transmitted, whereas the latter prevents an adversary from reading or falsifying messages.

Stegoencoding is sometimes deprecated as "weak encryption", because any stegoencoded message can be decoded, with a modest expenditure of computational resource, by an adversary who has contextual information about the message. By contrast, a strongly encrypted message can be read only by a skilled adversary who deploys massive computational resource. Furthermore, cryptographic techniques can be used to protect message integrity, whereas the integrity of a steganographic message can be attacked by any adversary who is able to decode it. However, a system's availability is adversely affected by its reliance on a cryptoprotocol, whenever a legitimate user has lost access to their key material, and whenever a cryptographic service is unavailable for an extended period of time. As we will show in this paper, a stegoprotocol can assure the successful completion of a shared computation for most of its participants; but this availability assurance comes at some expense in confidentiality. In this regard, our stegoprotocols are complementary to cryptoprotocols.

The Dolev-Yao model is widely accepted as the basis for cryptoprotocol design, because its axioms of strong cryptography and key-material secrecy are feasibly assured in many real-world situations, and because these axioms are sufficient to support a wide range of useful cryptoprotocols.

The primary contribution of this paper is an axiomatic model for stegoprotocol design. The adversary in our model is actively intercepting and interrupting, but is neither modifying nor impersonating.

We present and justify our model in Section 2. In Section 3, we illustrate our model by fully analysing a very small stegogame. In Section 4, we prove that stegocommunicants cannot conduct a secret ballot. In Section 5, we sketch a proof that stegocommunicants cannot be prevented from using a collaborative process to discover the minimum value in a public dataset of n values. In the concluding section, we summarise our findings and discuss some implications.

2 An Axiomatic Model of Stegocomputation

Axiom 1 *Each stegocommunicant can perform $O(1)$ randomized computations on $O(\log^{c_1} n)$-bit words, during each timestep, for some fixed constant c_1. Their multi-headed adversary, whom we name the Hydra, cannot predict the outputs of any stegocommunicant's private pseudorandom number generator.*

We strictly bound the computational power of stegocommunicants. We think it reasonable to assume that real-world stegocommunicants are able to take actions when cannot be predicted by their surveillants.

Axiom 2 *Each stegocommunicant has a unique name g_i, which is drawn at random from a set of size $O(n^{c_2})$ for some constant $c_2 > 3$.*

Randomly-selected names are sometimes called gensyms by LISP programmers, so our notation is mnemonic. For convenience when describing our stegoprotocols, we assume that the i-th stegocommunicant is named Gia, that the j-th is named Genji (when $j \neq i$), and that the k-th is named Ganika (when $k \neq i$ and $k \neq j$).

Axiom 3 *During each timestep, each stegocommunicant can send a stegomessage to one other stegocommunicant. If the destination of the stegomessage is unspecified, it is transmitted to a randomly selected stegocommunicant – and no one, not even the Hydra, can predict this random choice. Alternatively, the stegomessage may be addressed to someone already known (by gensym) to the stegotransmitter.*

We introduce this axiom to model a globally-accessible social network with millions or billions of participants. The participants in this network have agreed to accept a small number of messages per day from unknown sources, despite the risks of receiving objectionable messages, in order to participate in a public consensus-formation process which cannot effectively be censored by any government.

Random-introductions in social-networking systems are currently available in `https://www.facebook.com/RandomFriendAdder/` and `http://kikcontacts.com/random`.

Axiom 4 *No stegocommunicant knows their index i in any compact range $1..c_3n$, for any constant $c_3 \geq 1$.*

We leave it as an open problem to develop a stegoprotocol for mapping gensyms onto $1..n$. We note that such a compact enumeration would allow stegocommunicants to map their names onto the nodes in a shuffle-exchange graph or other powerful structure for parallel computation. If it turns out to be infeasible for a compact indexing to be stegocomputed, then it would be interesting to explore the properties of a stegomodel in which gensyms are drawn at random from a set of size c_3n.

For analytic convenience, we assume the Hydra always defines (at least implicitly) a bijection of $1..n$ onto inboxes, as well as a bijection of $1..n$ onto gensyms. When we refer to the i-th stegocommunicant, we are using the Hydra's bijections. Each stegocommunicant Gia therefore knows her own name, but not her index i.

Axiom 5 *The Hydra censors at most αn of the stegocommunicants in each timestep.*

To assure this axiom in a real-world setting, stegocommunicants could make public postings on a popular, governmentally-sanctioned, social-networking system that supports random-sharing of public posts. Censors would be expected to block postings with abnormally high entropy, because these are likely to be encrypted. However messages which closely resemble normal traffic [8,5,6,7] would evade mass censorship: the censor must search each one, individually, for stegocontent.

In a practical implementation of a stegocommunication system that obeys Axiom 5, stegotransmitters could randomly-share ten of their public postings each day (or week), using the "share-to-random" facility of Axiom 3. One of these random-shares is the cover message for that day's stegotransmission, using a stegosystem that is keyed to some recent public postings by the sender. The recipient of each random-share must expend some computational resource to search through all possible stegokeyings to discover its stegocontent, if any. Note that this stegochannel is, essentially, employing a cryptographic system with a keyspace that is small enough to allow stegodecoding of individual messages by individual recipients, but is large enough to prevent the Hydra from stegodecoding more than αn messages per timestep.

Axiom 6 *The stegocommunications network delivers a stegomessage if and only if there is no contention for the recipient's inbox. The stegocommunicant associated with this inbox is unaware of the message delivery if they are currently being censored; in this case, the Hydra reads the message.*

This axiom could be assured by a communication-services provider which allocates one fixed-size inbox to each stegocommunicant. It is analytically attractive, because it makes our computational model very similar to the well-studied Exclusive Read Exclusive Write (EREW) Parallel Random Access Machine (PRAM) [3].

Axiom 7 *The case of no incoming messages is indistinguishable from the case of multiple incoming messages in an inbox, for the intended recipient and for the Hydra.*

We introduce this indistinguishability solely to simplify our analysis. In a real-world deployment, this axiom could be violated by a governmental censor who instrumented the communications fabric for traffic analysis, allowing it to know how many stegomessages were sent to each stegocommunicant in each time period. We leave it to future work to analyse the properties of our model without this simplifying assumption.

Axiom 8 *The stegocommunicants' goal is a computation of a randomised function $f(A, X)$ in polylog(n) time. The vector A has one private component per stegocommunicant. The vector X is a globally-accessible, uncensorable, write-once vector of length $O(n)$. The value of each component of the domain and range of $f()$ is encoded in a bitstring of length polylogarithmic in n. For any distribution on the domain (A, X), the stegocommunicants win the game if their computation is complete, accurate, and widely-dispersed (as defined immediately below) with high probability,* i.e. *with chance of failure $O(n^{-c})$ for fixed $c > 0$.*

1. Complete*: A value for each of the components of $f()$ is declared.*
2. Accurate*: No stegocommunicant declares an incorrect value for any component of $f()$.*
3. Widely dispersed*: At least half of the stegocommunicants declare a value for at least one component of $f()$.*

We introduce the vector A of private information to model information that is generated by individual stegocommunicants, and which they are attempting to share with other stegocommunicants.

Each element in the vector X is written only once per stegocomputation. Depending on the problem, X could be $O(n)$ words of randomly-generated data, data collected from $O(n)$ real-world sensors, or data written by stegocommunicants. For example, if Gia were initially provided with her index i, she could write her vote into cell x_i; and the problem to be solved might be to collate the votes. In an implementation, x_i could be a designated area on Gia's timeline or blog.

The completion conditions are complex. We think they are best understood by working through an illustrative example in the next section.

Axiom 9 *All stegocommunicants follow the same (randomized) stegoprotocol, and this stegoprotocol is known to the Hydra.*

This axiom distinguishes our model sharply from distributed computing models in which a fraction of the participants are untrustworthy. Furthermore, distributed computing models are usually analysed for their worst-case performance, rather than for the w.h.p. bounds of our Axiom 8.

No axiom can ever be fully assured in a real-world system. Trustworthiness axioms, such as this one, are especially problematic. The trustworthiness of any stegocommunicant may change over time, and no person or computer system is completely trustworthy – there is always some chance of faulty behaviour. In this respect, our model is inaccurate: it provides an upper-bound, rather than an unbiased estimate, of the likelihood that any real-world set of n stegocommunicants can successfully complete a stegocomputation over a censorious communication network.

We note that the result of any stegocomputation may be assessed for accuracy in a subsequent stegocomputation. We leave the development of such trustworthiness-assessment stegoprotocols to future work.

3 Private 3-Majority

In this section, we illustrate our model by analysing a stegogame on $n = 3$ stegocommunicants s_1, s_2, and s_3. Their adversary is a one-headed Hydra ($\alpha = 1/3$).

Each stegocommunicant has a private bit a_i. Their shared goal is to evaluate the majority predicate on their private bits, $f(A) = (\sum_i a_i > 1)$, after a single round ($T = 1$) of communication. Each stegocommunicant must either declare her answer, or remain silent, at the end of this round.

By our last axiom, the stegocommunicants win their game if a majority of stegocommunicants declare a correct answer, and if nobody declares an incorrect answer.

Below, we evaluate the stegocommunicants' winning probability under a plausibly-optimal stegoprotocol, when their private bits a_i are independent Bernoulli variates with $p = 0.5$. If this were a formal analysis rather than an illustration of our model, we would prove (or disprove!) our conjecture that this probability distribution is pessimal for the stegocommunicants.

In our exemplary stegoprotocol, each stegocommunicant Gia (s_i) chooses a target Genji (s_j) uniformly at random under the constraint that $j \neq i$. The body of Gia's message is her random value a_i. If Genji receives Gia's message, he reports the value $\max(a_i, a_j)$ at time $t = 1$. Otherwise Genji reports the value a_j.

In our model, a message is received by its intended recipient unless one or more of the following conditions arise:

- the sender is censored,
- the receiver is censored, or

– there are multiple message-arrivals in the receiver's inbox.

Informally: when Gia is censored, her outgoing message is routed to the Hydra rather than to Gia's intended recipient. Furthermore, a censored Gia is unable to access her own inbox – because one of the heads of the Hydra is accessing this exclusive-read memory. If multiple messages arrive in Gia's inbox during a single round, this write-contention causes this inbox to be unreadable. Accordingly, our computational model is essentially an adversarial EREW PRAM, with the inboxes taking the role of memory cells in the PRAM model. However there are only n cells of memory, memory cells have wordsize polylogarithmic in n [2], and every memory cell is "owned" [4] by exactly one stegocommunicant.

In our illustrative single-round stegogame, the Hydra has just one head, so it has only two possible strategies: it may censor nobody, or it may censor one stegocommunicant. We identify two subcases in the first strategy:

1. The stegocommunicant's randomly-chosen messaging pattern is a 3-cycle. In this subcase, every stegocommunicant receives a message.
2. The messaging pattern has a 2-cycle. In this subcase, one stegocommunicant receives a message, one stegocommunicant receives no message due to inbox contention, and one stegocommunicant has an empty inbox.

In both subcases, the stegocommunicants compute the correct value if their votes are unanimous: $\sum_i a_i \in \{0, 3\}$. This event occurs with probability 1/4.

In both subcases, if $\sum_i a_i = 1$, the stegocommunicant with $a_j = 1$ reports an incorrect answer, causing the stegocommunicants to lose the game. This event occurs with probability 3/8.

In the first subcase, if $\sum_i a_i = 2$ then the stegocommunicants compute the correct value. This event occurs with probability 3/8, so the value of the game in the first subcase is $1(1/4) + 0(3/8) + 1(3/8) = 5/8$.

In the second subcase, if $\sum_i a_i = 2$ then the stegocommunicant with $a_j = 0$ reports a correct answer if and only if she receives a message. This event occurs with probability 1/3, so the value of the game in the second subcase is $1(1/4) + 0(3/8) + (1/3)(3/8) = 3/8$.

Subcase 1 arises with probability 1/4, independently of the values a_i, by the following argument. Without loss of generality s_1's target is s_2. With probability 0.5, s_2's target is s_3; and independently with probability 0.5, s_3's target is s_1.

We conclude that the stegocommunicants win the game against an uncensoring Hydra with probability $(5/8)(1/4) + (3/8)(1 - 1/4) = 14/32 = 7/16$.

We leave it to the reader to perform the (rather tedious) analysis of the Hydra's other possible strategy, of censoring one stegocommunicant, establishing the value of this game as 5/16.

4 Majority Voting with Unpublished Ballots

A Hydra with $\Omega(n)$ heads can effectively prevent majority voting with secret ballots. Formally:

Theorem 1 *For any constant censorship rate $\alpha > 0$, the predicate $(\sum_i a_i \geq n/2)$ can not be reliably stegocomputed.*

Proof. A sufficient strategy for the Hydra is to choose αn stegocommunicants at random, and to censor these stegocommunicants at all times. A censored stegocommunicant Gia does not communicate her a_i value to anyone. The uncensored stegocommunicants may estimate the total vote of the censored stegocommunicants, $\sum_{\{i:\exists k:C_k(1)=i\}} a_i$, by random sampling. However for an input ensemble in which a_i are independent Bernoulli variates with probability $p = 0.5$, the error in this estimate is $\Omega(\ln c\sqrt{\alpha n}) = \Omega((\ln(\alpha c)\sqrt{n})$ with probability $\Omega(n^{-c})$, implying that the stegocommunicants will not accurately compute the majority vote w.h.p.

5 Global Minimum-finding on a Public Vector

A Hydra with $n/10$ heads cannot prevent n stegocommunicants from discovering the value of a smallest component in their globally-readable n vector X. A suitable stegoprotocol is easily described: each stegocommunicant probes the n-vector at random, retaining the index of the smallest value it has seen so far, and sending this index to a randomly-selected recipient.

Theorem 2 *If $\alpha \leq 0.1$, then $\min_i(x_i)$ can be reliably stegocomputed.*

Proof sketch. Every stegocommunicant probes at random into the n-vector X during each timestep, discovering a global minimum with probability $\geq 1/n$; this bound is tight when the global minimum is unique. Every stegocommunicant informs a randomly-selected stegocommunicant of the index of the minimal value it has seen to date. We use a discrete-state branching process [1] to model the spread of knowledge about the global minimum.

6 Discussion

We have exhibited a model of stegocommunication which supports proofs of reliable computation on a EREW PRAM model with adversarial message interceptions and interruptions.

We have proven that the adversary can prevent stegocommunicants from reliably computing the majority function on their private "votes". We note that an approximate private-vote could be stegocomputed by random-sampling. Furthermore, a majority public-vote could be stegodecided unless the voting is close.

We have also proven that the adversary cannot prevent stegocommunicants from discovering the minimum value in a public n-vector. This computational power would allow stegocommunicants to form a public consensus on the "best stegoprotocol" to be used in the next round of stegocomputation – if they had a prior agreement on the metric to be used when comparing two stegoprotocols.

We note that our model bears some resemblance to models of fault-tolerant distributed systems. However such models generally have a Byzantine trust model, such that any communicant may be untrustworthy. Furthermore the models generally lack a probabilistic support, but instead are analysed for worst-case behaviour: the algorithms are required to deliver correct results under a bounded-fault assumption e.g. that no more than 1/3 of the Byzantine generals are untrustworthy. Under such models of distributed computation, runtimes are typically polynomial in n. By contrast, our model assumes n trustworthy stegocommunicants who have only polylog time to complete their computation.

Our primary contribution in this article is an axiomatised model of stegocomputation which is simple enough to be analytic, while remaining realistic enough to guide the design of reliable stegosystems of practical use. Some foreseeable practical uses of stegocomputation are "white-hat", for example the reliable distribution of digital certificates in a global public-key infrastructure – when one or more governments are actively attempting to prevent this distribution. Reliable stegocomputation would also be important to "black-hats", for example criminal gangs may someday use a stegogame to coordinate their criminal activity, if no crime-fighting agency has sufficient powers of censorship to prevent such coordination.

References

1. Allen, L.J.: Continuous-time and discrete-state branching processes. In: Stochastic Population and Epidemic Models, Mathematical Biosciences Institute Lecture Series, vol. 1.3, pp. 1–12. Springer International Publishing (2015)
2. Bellantoni, S.J.: Parallel random access machines with bounded memory wordsize. Information and Computation 91(2), 259 – 273 (1991)
3. Borodin, A., von zur Gathen, J., Hopcroft, J.: Fast parallel matrix and GCD computations. Information and Control 52(3), 241–256 (1982)
4. Fernau, H., Lange, K., Reinhardt, K.: Advocating ownership. In: Chandru, V., Vinay, V. (eds.) Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Computer Science, vol. 1180, pp. 286–297. Springer Berlin Heidelberg (1996)
5. Hopper, N., von Ahn, L., Langford, J.: Provably secure steganography. IEEE Transactions on Computers 58(5), 662–676 (2009)
6. Liśkiewicz, M., Reischuk, R., Wölfel, U.: Grey-box steganography. Theoretical Computer Science 505, 27–41 (2013)
7. Takebe, H., Tanaka, K.: Grey-box public-key steganography. In: Chan, T.H., Lau, L., Trevisan, L. (eds.) Theory and Applications of Models of Computation, Lecture Notes in Computer Science, vol. 7876, pp. 294–305. Springer Berlin Heidelberg (2013)
8. Wayner, P.: Mimic functions. Cryptologia XVI(3), 193–214 (1992)