A Note on the Problem of Semantic Interpretation Agreement in Steganographic Communications

Benjamin Aziz*
School of Computing, University of Portsmouth, Portsmouth, United Kingdom
benjamin.aziz@port.ac.uk

Received: June 21, 2021; Accepted: August 3, 2021; Published: August 31, 2021

Abstract

This paper describes a solution to one of the fundamental problems in stegnagoraphy, namely that two participants can agree on a common interpretation of the communicated hidden message. Thus far, most literature in the field of steganography has focused on two problems: first, the embedding (and extraction) of hidden content and second, the analysis of suspicious messages by an external attacker. We argue here that there is a third fundamental dimension that characterises steganography, namely the ability of the three parties to interpret hidden content. We offer here one solution based on the Diffie-Hellman key-agreement calculation that leads to interpretation agreement.

Keywords: formal methods, protocols, semantics, steganography

1 Introduction

Steganography, or the art of communicating hidden content in innocent-looking messages, remains in modern times one of the active areas of Internet security and privacy research [12], and beyond [10]. The problem that steganography attempts to solve is known as the "prisoners problem", first formulated by Simmons in 1984 [25] and based on instructions to soldiers on how to use hidden (not necessarily scrambled) communications in Wold War II [15]. In this problem, two prisoners, Alice and Bob, are arrested in a prison. The prison warden, Wendy, offers to allow both Alice and Bob to exchange communications as long as those communications are in the clear (i.e. are not scrambled by encryption). Alice and Bob then use this opportunity to exchange communications that have hidden secret messages inside, in order for them to construct a plan of escape from the prison. The communication channel achieving this is known as a *subliminal channel*, since each communication has two interpretations: one known as the *cover*, which represents the obvious meaning of the communication, which Wendy can interpret, and the other a *hidden message*, which represents the secret meaning of the communication and which Wendy can not interpret but Alice and Bob can.

Whilst research in cryptographic communications has long been considered at a variety of abstraction levels and from the point of view of different security objectives, the majority of research in steganography has thus far remained focused on the computational level, defining algorithms for solving two broad problems only: First, how to embed and extract the secret message (meaning) using some cover medium. This is known as the Alice-Bob (or steganography) problem and it is primarily concerned with the *construction of stego-objects*. Second, how to analyse a cover image for probable secret content. This is known as the Wendy (or steganalysis) problem, and it is concerened with the *identification of stego-objects*.

Journal of Internet Services and Information Security (JISIS), volume: 11, number: 3 (August 2021), pp. 47-57
DOI:*10.22667/JISIS.2021.08.31.047*

*Corresponding author: School of Computing, University of Portsmouth, Portsmouth PO1 3HE, United Kingdom, Tel: +44-(0)23-9284-2265, Web: `http://azizb.myweb.port.ac.uk/`

Cachin [6] demonstrated that an information-theoretic model of steganography can be defined using relative entropy between the cover and stego-object, and that entropy determined also the bounds of the ability of Wendy to identify whether an object is stego or clean. Based on Cachin's model then, Bergmair and Katzenbeisser [5] were able to define the notion of a *context-aware steganographic system*, in which they introduced a new kind of security primitive, namely that of *the steganographic interpretation* of a cover medium, or what we will call in this paper, the steganographic semantic function.

This paper aims to provide a concrete solution to the problem of sharing a steganographic semantic function between two communicating parties in the presence of an active attacker seeking to disrupt their communications. We propose the use of a fundamental primitive, the Diffie-Hellman [9] key-sharing calculation, in order to establish an agreement on the steganographic semantic function between the communicating parties. We further demonstrate that the inherent security underlying a Diffie-Hellman key-sharing protocol also propagates to the next level of abstraction, namely the sharing of the steganographic semantic function. We establish this through a formal static analysis of the proposed solution using a process algebraic language.

The rest of the paper is structured as follows. In Section 2, we outline the problem of semantic interpretation agreement in the context of steganography. In Section 3, we discuss related work. In Section 4, we present our model for solving the problem. In Section 5, we give a formal static analysis of the proposed algorithm. Finally, we conclude the paper in Section 6 and give directions for future work.

2 The Problem

According to Bergmair and Katzenbeisser [5], the most fundamental principle underlying all steganography and steganalysis is the semantic-level understanding of objects. Unlike cryptography, which is purely concerned with the syntactic hiding of content without any attempt to hide its existence, steganography uses syntactic processing as a means for aiding Alice and Bob towards making the correct interpretation of stego-objects thereby leading to the discovery of the secret content. Wendy, on the other hand, is pushed away from that (correct) interpretation and hence away from the discovery of the secret content. Wendy utilises steganalaysis to measure its probabilistic distance from this correct interpretation in determining the real meaning of an object.

The above principle of semantics as the basis of steganography introduces an important problem: how can Alice and Bob agree on the interpretation of objects whilst they are communicating to construct a plan of escape from the prison? Whilst this agreement of interpretation has been simplified to other forms in literature, e.g. to a key-agreement problem in image-based steganograghy, it remains unsolved in its general form: *the agreement on a semantic function mapping an object to its true meaning*. Put differently, given that Alice and Bob would like to communicate some hidden content in the presence of Wendy, how do they agree a common interpretation of that content different from Wendy's interpretation? This common interpretation, in essence, becomes their interpretation of the plan of escape from the prison.

3 Related Work

Our idea in this paper is inspired by the denotational semantics approach of Scott and Strachey [23, 24], whose framework advocated the interpretation of programs as mathematical objects (e.g. λ terms [8]) using computable semantic functions. In our case, the distribution of the agents interpreting a message introduces an additional dimension to the problem: *agreement on the semantic interpretation*. Most of the work in steganography literature has focused on the problem of embedding and extracting hidden content, whether that is symmetric key-based or public key-based [14], but with little emphasis on properties of

the communication protocol itself. In fact, there is lack of research defining what the purpose of the steganographic communications is, beyond of course the old prisoners problem. Even works, such as [17], tackling the distributed nature of steganography, fail to suggest any notion of a protocol or how messages should be interpreted.

Provos and Honeyman [21] presented an early work on how to detect steganographic content on the Internet, which came on the wake of the terrorist attacks on New York in 2001, and the consequent suspicion that terrorists were using the Internet for communicating hidden content. Their work found no evidence of the use of steganography at the time, although one interesting interpretation given was that terrorists may not be using systems that the authors could find. This could also have been indicative of the terrorists communicating at a "different semantic" level. At the same time, Fisk et al. [13] defined, for the first time, a method for active wardens to intercept and eliminate TCP traffic on the Internet that may contain hidden suspicious content. Their definition of active versus passive wardens follows that of Anderson and Petitcolas [3], where a passive warden simply *detects* hidden content whereas an active warden seeks to *eliminate* that content. Neither attempts to *interpret* the content.

In recent years, steganography has found applications in Internet-related domains such as Cloud computing. For example, Sarkar and Chatterjee [22] demonstrate how secure Cloud storage can be implemented using steganogrpahy. Generally Cloud computing is seen as any other carrier medium that can be used for steganographic communications [18]. Steganogrpahy also poses challenged, in the context of the Internet [16] and the Internet-of-Things [19], as it is used for defeating forensic investigations.

4 The Model

Assume that $\mathscr{C} = \{c_1, c_2, \ldots\}$ is the set of all cover objects and $\mathscr{S} = \{s_1, s_2, \ldots\}$ the set of all secret messages. We write c_s to specify a cover object embedded with a secret message, also known as a stego-object. We define $\Psi : \mathscr{C} \times \mathscr{S} \rightarrow D_\perp$ as the space of all semantic functions that assign a meaning, $\psi(c_s) = o \in D_\perp$, to a stego-object c_s. We define meaning as semantic objects, $o_1, o_2, \ldots$ that form a semantic domain, $D_\perp$, where $\perp_D$ is the bottom element of the domain denoting undefined or unknown meaning. The special semantic function, $\psi_\perp$, maps every stego-object to this undefined meaning:

$$\forall c \in \mathscr{C}, s \in \mathscr{S} : \psi_\perp(c_s) = \perp_D$$

In other words, the above function cannot discover any meaning in any object. We assume that this is the default interpretation that all of Alice, Bob and Wendy are equipped with initially (i.e. the bootstrap state). The knowledge of how to interpret then becomes a problem of the distribution of semantic functions, or what we term the *steganography problem*, described next.

4.1 The Steganography Problem

The steganography problem can now be formulated as a semantic function agreement problem [5], in which Alice and Bob must agree on a common interpretation of all stego-objects they will communicate in the future. More formally:

$$\psi_A = \psi_B$$

For simplicity, we refer to either ψ_A or ψ_B as ψ_{AB}. There is a requirement that Alice and Bob's interpretation is different from that of Wendy's:

$$\psi_{AB} \neq \psi_W$$

although, in practice, this requirement is impossible to meet for all cases other than the initial one, i.e.:

$$(\psi_W = \psi_\perp) \Rightarrow (\psi_{AB} \neq \psi_\perp)$$

simply because Alice and Bob would have no idea of the level of knowledge and intelligence that Wendy is at in any particular point of time, for every other case of ψ_W.

4.2 The Steganalysis Problem

The steganalysis problem, on the other hand, is divided into two problems [7, p.2]. The first, known as *passive steganalysis*, is simply to determine the probability that the real meaning of a stego-object is different from the undefined meaning:

$$P[\psi_{AB}(c_s) \neq \psi_{\perp}(c_s)]$$

This first problem is concerned only with determining whether or not there is *some* hidden meaning (content) in an object, but not what that meaning is. The higher the probability the more likely it is that there is some such hidden meaning, where $P[\psi_{AB}(c_s) \neq \psi_{\perp}(c_s)] = 1$ means that there is definitely some meaning in the object's content, and $P[\psi_{AB}(c_s) \neq \psi_{\perp}(c_s)] = 0$ means that there is no such meaning (i.e. it is simply an innocent object).

The second problem however, called *active steganalysis*, attempts to determine what that hidden meaning is (exactly or approximately). This is expressed as the probability that Wendy's interpretation will equal some semantic object, $o \in D_{\perp}$:

$$P[\psi_W(c_s) = o] \quad \textit{where,} \quad o \sqsubseteq \psi_{AB}(c_s)$$

Where o is either the object of Alice and Bob's interpretation, or any other semantic object comparable to that interpretation in some lesser sense. The probability here can be determined for any semantic object, $o = \psi_W(c_s)$, that Wendy can arrive at, comparable to the semantic object, $\psi_{AB}(c_s)$, reflecting the real meaning. When Wendy calculates the probability of $\psi_W(c_s) = \psi_{AB}(c_s)$, she is attempting to understand the *exact* meaning of the stego-object. Otherwise, if Wendy is calculating the probability $\psi_W(c_s) \sqsubset \psi_{AB}(c_s)$, then she is attempting to understand an *approximate* meaning of the stego-object[1]. The higher the probability for each interpretation, the more confident Wendy is of the meaning she is attempting to understand. When Wendy calculates the probability of non-comparable semantic objects, (i.e. where $\psi_W(c_s) \not\sqsubseteq \psi_{AB}(c_s)$), we say that Wendy has arrived at some *wrong interpretation*, totally irrelevant to the context of the communication.

4.3 Same and Different Semantic Domains

If both Alice and Bob on one hand, and Wendy on the other hand, were set up to interpret stego-objects in the same semantic domain, then they would all be at a level playing field:

$$\textit{co-dom}(\psi_{AB}) = \textit{co-dom}(\psi_W)$$

and this would result in, for all objects c_s:

$$P[\psi_{AB}(c_s) = \psi_W(c_s)] \neq 0$$

However, a more secure set-up would be when the interpretation used by Alice and Bob is in a different semantic domain to that of Wendy:

$$\textit{co-dom}(\psi_{AB}) \neq \textit{co-dom}(\psi_W)$$

in this case, we have that:

$$P[\psi_{AB}(c_s) = \psi_W(c_s)] = 0$$

[1] for cases where $o \sqsupset \psi_{AB}(c_s)$, we abstract o to $\psi_{AB}(c_s)$, since the additional meaning is of no interest.

4.4 Some Conditions for a Secure Steganographic System

We outline below a number of conditions each of which would be sufficient to guarantee a steganographic system to be completely secure.

1. $\psi_W = \psi_\perp$: with this condition, Wendy is guaranteed to always interpret stego-objects as having no hidden meaning. However, this is not a realistic condition since Wendy (whether a system or a human) is bound to eventually learn to interpret objects within some meaningful semantic domain.

2. $P[\psi_W(c_s) = o] = 0$ *where,* $o \sqsubseteq \psi_{AB}(c_s)$: with this condition, Wendy will not believe (e.g. in terms of her confidence) in any relevant interpretation of the hidden meaning, simply because she will think the probability of that is zero.

3. $range(\psi_W) \cap range(\psi_{AB}) = \{\}$: with this condition, Wendy will be interpreting stego-objects in a completely different semantic domain (non-level playing field), therefore, she is always bound to arrive at a different interpretation from Alice and Bob.

4.5 Diffie-Hellmann as a Basis for Agreement

Mitchell, Ward and Wilson [20] described a method where Bob could influence a Diffie-Hellmann key [9] it computes with Alice, while preserving the security of the key against an external observer. Recall quickly that the Diffie-Hellmann protocol consists of a pair of messages:

1. $A \rightarrow B: \quad g^a \ mod \ p$
2. $B \rightarrow A: \quad g^b \ mod \ p$

where at the end of the protocol, A computes $g^{b \times a} \ mod \ p = K$ and B computes $g^{a \times b} \ mod \ p = K$, where $a \in \mathbb{N}$ is A's secret number, $b \in \mathbb{N}$ is B's secret number and $g, p \in \mathbb{N}$ are public numbers. Each of these numbers have certain computational characteristics [9] that are outside of the scope of our argument.

In the above, it is possible for Bob to influence a number of bits of the generated key, K. This is done by setting, *a priori*, a number of bits to certain values, then Bob would choose its secret number, b, to produce a specific K that matches those preset bits. Such influence reflects both strong and selective key control [26] and it works in normal cryptography-based scenarios only if Alice is willing to tolerate certain latency between the sending of Message 1 and her receiving Message 2, and therefore would limit Bob's ability to influence the whole key in practical terms. However, in our prisoners scenario, there is no such time limitation requirement, and therefore, Bob can delay its sending of message 2 till any point in time he can compute the right value of the secret b influencing the whole of K.

We propose the above method as an example of a solution to the problem of semantic function agreement between Alice and Bob. In this solution, assume that every semantic function, $\psi \in \Psi$, is indexed by a key, $K_\psi \in \mathbb{N}$. We can retrieve the semantic function corresponding to some key using an index operator, $\iota : \mathbb{N} \rightarrow (\mathscr{C} \times \mathscr{S} \rightarrow D_\perp)$. If Alice and Bob were now to engage in a Diffie-Hellmann handshake, they would end up agreeing a key. Furthermore, if Bob could influence this key (à la Mitchell et al. [20]), Alice and Bob would end up agreeing on a specific key, K_{AB}. In other words, they would end up agreeing on a specific semantic function $\psi_{AB} = \iota(K_{AB})$.

Algorithm 1 outlines the full steps for the proposed method, where ψ_B is the semantic function that A will use to interpret future communications received from B with, and ψ_A is vice versa. The most important outcome of this algorithm is that $\psi_B = \psi_A$.

A run of this protocol amounts to calling the ALICE procedure, which will then start the protocol. As an example, the semantic function ψ could be thought of as one that has a domain as a dictionary, therefore, a secure steganographic communication would be one where the dictionary used by Alice and

1: **procedure** ALICE
2: Choose some $a \in \mathbb{N}$
3: Compute $K = \text{BOB}(c_A)^a \ mod \ p$ ▷ where $c_A = g^a \ mod \ p$
4: Set in internal state $\psi_B =: \iota(K)$
5: **return**
6: **end procedure**
7:
8: **procedure** BOB(c_A)
9: Choose some $\psi \in \Psi$
10: Choose a specific $b \in \mathbb{N}$ such that $\iota({c_A}^b \ mod \ p) = \psi$
11: Set in internal state that $\psi_A = \psi$
12: **return** c_B ▷ where $c_B = g^b \ mod \ p$
13: **end procedure**

Algorithm 1: The DH-based Semantic Function Agreement Algorithm [p, g, ι are global parameters, whereas ψ_A, ψ_B are local state variables]

Bob is totally different than the one used by Wendy, therefore, interpreting hidden content in a different manner.

5 Formal Abstract Analysis

We present here a formal model and an abstract static analysis of Algorithm 1 using the formal language of the applied pi calculus [1]. The specification of the protocol is shown in Figure 1.

$$
\begin{array}{lll}
\textit{Alice} & \stackrel{\text{def}}{=} & \nu a.(\\
 & & \overline{c_Y}\langle (g^a \ mod \ p)\rangle.c_X(x_{XY}).\overline{run}\langle next\rangle.\mathbf{0}\{\iota({x_{XY}}^a \ mod \ p)/\psi_B\}) \\
\textit{Bob} & \stackrel{\text{def}}{=} & \nu\psi.\nu b.(\\
 & & c_Y(x'_{XY}).\textit{if } \psi = \iota({x'_{XY}}^b \ mod \ p) \textit{ then } \overline{c_X}\langle (g^b \ mod \ p)\rangle.\mathbf{0}\{\psi/\psi_A\} \textit{ else } \overline{run}\langle next\rangle.\mathbf{0}) \\
I & \stackrel{\text{def}}{=} & \nu i.(\ \bar{i}\langle \kappa_0\rangle \ \mid \ !i(\kappa).(\nu net.\bar{i}\langle \kappa \cup \{net\}\rangle \ \mid \\
 & & \quad \prod_{\forall M,N \in set(\kappa)} \overline{M}\langle N\rangle.\bar{i}\langle \kappa\rangle \ \mid \\
 & & \quad \prod_{\forall M \in set(\kappa)} M(x).\bar{i}\langle \kappa + x\rangle \ \mid \\
 & & \quad \prod_{\forall f \in \Sigma, M, N_1, \ldots, N_n \in set(\kappa)} \overline{M}\langle f(N_1, \ldots, N_n)\rangle.\bar{i}\langle \kappa + f(N_1, \ldots, N_n)\rangle \ \mid \\
 & & \quad \prod_{\forall x, M \in set(\kappa)} \{M/x\}.\bar{i}\langle \kappa\rangle)) \\
\textit{Protocol} & \stackrel{\text{def}}{=} & !run(w).\nu p'.\nu g'.\nu \iota'.(Alice\{A/X\}\{B/Y\}\{p'/p\}\{g'/g\}\{\iota'/\iota\} \ \mid \\
 & & Bob\{A/X\}\{B/Y\} \ \mid \ I\{(A, B, I, c_A, c_B, c_I, p', g')/\kappa_0\}) \ \mid \ \overline{run}\langle first\rangle
\end{array}
$$

Figure 1: Specification of the DH-based Semantic Function Agreement Algorithm

The specification models the ALICE and BOB procedures of Algorithm 1, using two similarly named processes. The *Alice* process is instantiated in the *Protocol* process with five values; its own name, A, the name of the entity it is communicating with, B (for *Bob*), the two Diffie-Hellman numbers, p' and g', and finally, the index operator, ι', for retrieving the semantic function corresponding to some key. The *Alice* process, when instantiated, will create a new secret number, a, which then uses to create the

first Diffie-Hellman computation, $g^a \ mod \ p$. It then sends this value to the *Bob* process, and waits from that process to receive the first computation computed by *Bob*. Once that happens, *Alice* terminates by storing the semantic function indexed by ι in its internal state as the value of ψ_B, and at the same time, instantiates a new run of the protocol.

On the other hand, the *Bob* process is instantiated first with the names of the communicating entities, *A* for *Alice* and *B* for *Bob*. It then proceeds to create a couple of new names; ψ which is some chosen semantic function, and b, which is *Bob*'s secret number. After receiving the first Diffie-Hellman computation from *Alice*, the next step is important; it simply puts a condition on the selected value of both ψ and b such that $\psi = \iota({x'_{XY}}^{b} \ mod \ p)$, meaning that the selected b number will lead to the desired ψ value. If that is the case, *Bob* will proceed to send to *Alice* its first Diffie-Hellman computation, $g^b \ mod \ p$, and at the same time, register the value of ψ internally as the semantic function to use when communicating with *Alice*. Alternatively, if the above equality does not hold, nothing happens and the next run of the protocol is called.

The *Protocol* process also includes the process representing the attacker, I, running in parallel with *Alice* and *Bob*. This attacker process is a model of Dolev-Yao's most powerful attacker [11] and it is described in detail in [4, §7]. In summary, such an attacker is capable of repeatedly doing anything the language of the model allows it to do. One thing to note in this attacker is that it is instantiated with an initial knowledge of names, κ_0, which is then increased. Note that $set(\kappa)$ is the set underlying the values in the κ tuple. The *Protocol* process also includes a first signal, $\overline{run}\langle first \rangle$, to run the protocol.

A full description of the syntax and denotational semantics of the applied pi calculus is given in [4]. We give briefly here a reminder of the syntax of the language, as shown in Figure 2. This syn-

$L,M,N,T,U,V \in \mathscr{T}$	$::=$	Terms
	$a,b,c,\ldots,s \in \mathscr{N}$	Names
	$x,y,z \in \mathscr{V}$	Variables
	$f(M_1,\ldots,M_n)$	Function application
$P,Q,R \in \mathscr{P}$	$::=$	Processes
	$\mathbf{0}$	Null process
	$P \mid Q$	Parallel composition
	$P+Q$	Non-deterministic choice
	$!P$	Replication
	$\nu n.P$	Name restriction
	$if \ M=N \ then \ P \ else \ Q$	Conditional
	$M(x).P$	Input
	$\overline{M}\langle N \rangle.P$	Output
$A,B,C \in \mathscr{EP}$	$::=$	Extended processes
	P	Process
	$A \mid B$	Parallel composition
	$\nu n.A$	Name restriction
	$\nu x.A$	Variable restriction
	$\{M/x\}$	Active substitution

Figure 2: The Syntax of the Applied Pi Calculus

tax is similar to that of [2]. Functions are assumed to be taken from a finite signature, Σ, equipped with an equational theory that is used to infer when two terms are equal, $\Sigma \vdash M = L$, for example, $\texttt{decrypt}(\texttt{encrypt}(x,y),y) = x$, to indicate that decryption can reverse encryption using the right key, and $\texttt{decrypt}(\texttt{sig}(x,K),C) = x$, to indicate that a successful digital signature verification process using a

private key can result in extracting the signed term in the presence of a valid public certificate. Moreover, by default, $\Sigma \vdash M = M$. We also assume that the usual notions of free and bound names and variables of terms, processes and extended processes, as well as α-conversion all apply.

In order for the syntax to make sense, it requires the definition of some sort of operational semantics. These are defined in terms of the structural congruence relation (Figure (3)) and the reaction relation (Figure (4)). In this semantics, we have replaced early input transitions, $A \xrightarrow{a(M)} A'$ as they appeared in [2, §4.4], with the late version, $A \xrightarrow{a(x)} A'$. This implies that input actions are only be instantiated through reductions, $\overline{a}\langle x\rangle.P \mid a(x).Q \xrightarrow{\tau} P \mid Q$, and active substitutions, $\{M/x\}$.

PAR-0	A	$\equiv$	$A \mid 0$
PAR-A	$A \mid (B \mid C)$	$\equiv$	$(A \mid B) \mid C$
PAR-C	$A \mid B$	$\equiv$	$B \mid A$
CH-0	A	$\equiv$	$A + A$
CH-A	$A + (B + C)$	$\equiv$	$(A + B) + C$
CH-C	$A + B$	$\equiv$	$B + A$
REPL	$!P$	$\equiv$	$P \mid !P$
NEW-0	$\nu n.0$	$\equiv$	0
NEW-C	$\nu u.\nu v.A$	$\equiv$	$\nu v.\nu u.A$
NEW-PAR	$A \mid \nu v.B$	$\equiv$	$\nu u.(A \mid B)$ *when,* $u \notin fv(A) \cup fn(A)$
ALIAS	$\nu x.\{M/x\}$	$\equiv$	0
SUBST	$\{M/x\} \mid A$	$\equiv$	$\{M/x\} \mid A\{M/x\}$
REWRITE	$\{M/x\}$	$\equiv$	$\{N/x\}$ *when,* $\Sigma \vdash M = N$

Figure 3: Rules of the structural congruence relation, $\equiv$

In [4], we defined an abstract interpretation-based static analysis for the applied pi calculus, that would produce an abstract environment, $\phi_{\mathscr{A}} : \mathscr{V}^\sharp \to \wp(\mathscr{N}^\sharp)$, which maps each abstract bound variable of the analysed process to a set of abstract names, representing terms that could substitute that variable. An abstract domain, $D^\sharp_\perp = \mathscr{V}^\sharp \to \wp(\mathscr{N}^\sharp)$, is thus formed, ordered by subset inclusion, as follows:

$$\forall \phi_{\mathscr{A}1}, \phi_{\mathscr{A}2} \in D^\sharp_\perp, x \in V^\sharp : \phi_{\mathscr{A}1} \sqsubseteq_{D^\sharp_\perp} \phi_{\mathscr{A}2} \Leftrightarrow \phi_{\mathscr{A}1}(x) \subseteq \phi_{\mathscr{A}2}(x)$$

where, $\perp_{D^\sharp_\perp} = \phi_{\mathscr{A}0}$. The outcome of any abstract analysis will be an environment $\phi_{\mathscr{A}} \in D^\sharp$. Taking $D^\sharp_\perp$ as the abstract semantic domain, one can define an abstract semantic function, $\mathscr{A}([\![A]\!])\ \rho\ \phi_{\mathscr{A}} \in D^\sharp_\perp$, which returns the result of the static analysis of A. For the full rules of $\mathscr{A}$, we refer the reader to [4].

Next, we analyse the protocol, by applying $\mathscr{A}([\![Protocol]\!])\ \{|\ |\}\ \phi_{\mathscr{A}0}$ with $\alpha_{2,2}$, which enables us to monitor two runs of the protocol (non-uniform analysis). The results are shown in Figure 5 for some of the values of the final $\phi_{\mathscr{A}}$.

From the results of this analysis as shown in Figure 5, we see that the protocol is secure as the attacker will never obtain any complete Diffie-Hellman key, and hence it will not be able to arrive at the correct semantic function using the index operator, ι'.

6 Conclusion

This paper provides a fresh new look at the problem of steganographic communications as seen from the perspective of semantic interpretations of the communicated messages [5]. We argue that the Diffie-

COMM	$\overline{a}\langle x\rangle.P \mid a(x).Q \xrightarrow{\tau} P \mid Q$		
THEN	$\mathit{if}\ M = M\ \mathit{then}\ P\ \mathit{else}\ Q \xrightarrow{\tau} P$		
ELSE	$\mathit{if}\ M = N\ \mathit{then}\ P\ \mathit{else}\ Q \xrightarrow{\tau} Q$		
	for any ground terms M and N such that $\Sigma \nvdash M = N$		
LATE-IN	$a(x).P \xrightarrow{a(x)} P$		
OUT-TERM	$\overline{a}\langle M\rangle.P \xrightarrow{\overline{a}\langle M\rangle} P$		
LIM-SCOPE	$A \xrightarrow{a\pi'} A'$		
	where, $a\pi' \in \{a(x), a\langle M\rangle, \tau\}$		
	and, $u \notin n(a\pi')$	$\Rightarrow$	$\nu u.A \xrightarrow{a\pi'} \nu u.A'$
PAR	$A \xrightarrow{a\pi} A' \wedge$		
	$bv(a\pi) \cap fv(B) = bn(a\pi) \cap fn(B) = \{\}$	$\Rightarrow$	$A \mid B \xrightarrow{a\pi} A' \mid B$
OPEN-CHANNEL	$A \xrightarrow{\overline{a}\langle b\rangle} A' \wedge b \neq a$	$\Rightarrow$	$\nu b.A \xrightarrow{\nu b.\overline{a}\langle b\rangle} A'$
OPEN-VARIABLE	$A \xrightarrow{\nu\, u_1 \ldots \nu\, u_k.\overline{a}\langle M\rangle} A' \wedge$		
	$x \in fv(M) \backslash \{u_1, \ldots, u_k\} \wedge$		
	x can be derived from		
	$(\nu u_1 \ldots \nu u_k.\{M/z\} \mid A')$	$\Rightarrow$	$\nu x.A \xrightarrow{\nu x.\nu\, u_1 \ldots \nu\, u_k.\overline{a}\langle M\rangle} A'$
STRUCT	$A \equiv B \wedge B \xrightarrow{a\pi} B' \wedge B' \equiv A'$	$\Rightarrow$	$A \xrightarrow{a\pi} A'$

Figure 4: Rules of the refined late labelled transition relation, $\xrightarrow{a\pi}$

$$
\begin{array}{l}
\kappa_1 \mapsto \{(A, B, I, c_A, c_B, c_I, p_1', g_1'), net, ({g_1'}^{b_1} \bmod p_1'), ({g_1'}^{a_1} \bmod p_1')\} \\
\kappa_2 \mapsto \{(A, B, I, c_A, c_B, c_I, p_2', g_2'), net, ({g_2'}^{b_2} \bmod p_2'), ({g_2'}^{a_2} \bmod p_2')\} \\
x_{AB1} \mapsto \{({g_1'}^{b_1} \bmod p_1')\} \\
x_{AB2} \mapsto \{({g_2'}^{b_2} \bmod p_2')\} \\
x'_{AB1} \mapsto \{({g_1'}^{a_1} \bmod p_1')\} \\
x'_{AB2} \mapsto \{({g_2'}^{a_2} \bmod p_2')\} \\
\psi_{A1} \mapsto \{\psi_1\} \\
\psi_{A2} \mapsto \{\psi_2\} \\
\left.\begin{array}{l} \psi_{B1} \mapsto \{\iota_1'({x_{AB1}}^{a_1} \bmod p_1')\} \\ \psi_{B2} \mapsto \{\iota_2'({x_{AB2}}^{a_2} \bmod p_2')\} \end{array}\right\} \quad \mathit{if}\ \psi_1 = \iota_1'({x'_{AB1}}^{b_1} \bmod p_1')\ \mathit{and}\ \psi = \iota_2'({x'_{AB2}}^{b_2} \bmod p_2') \\
w_1 \mapsto \{\mathit{first}\} \\
w_2 \mapsto \{\mathit{next}\} \\
\kappa_{0_1} \mapsto \{(A, B, I, c_A, c_B, c_I, p_1', g_1')\} \\
\kappa_{0_2} \mapsto \{(A, B, I, c_A, c_B, c_I, p_2', g_2')\}
\end{array}
$$

Figure 5: Results of the non-uniform analysis of the Diffie-Hellman protocol.

Hellmann operator [9], and similar to the case of cryptography, is also a fundamental operator in the case of steganography. We outline a method of how Diffie-Hellmann key calculation can be used to influence a shared key, which is then used as a basis for future semantic interpretations of the communicated messages. Future work will focus on validating the new idea through the implementation of a prototype tool based on the outlined method. We also plan to specify more rigorously the algorithm, and produce a formal model that can be used as a blueprint for any future implementations. In fact, this is a major area of research that we consider currently missing in steganography-related research, namely the specification and definition of protocol-level properties, akin to what exists in cryptography.

References

[1] M. Abadi, B. Blanchet, and C. Fournet. The applied pi calculus: Mobile values, new names, and secure communication. *Journal of the ACM*, 65(1):1–41, October 2018.

[2] M. Abadi and C. Fournet. Mobile Values, New Names, and Secure Communication. In *Proc. of the 28th ACM Symposium on Principles of Programming Languages (POPL'01), London, UK*, pages 104–115. ACM, January 2001.

[3] R. J. Anderson and F. A. Petitcolas. On the limits of steganography. *IEEE Journal on selected areas in communications*, 16(4):474–481, May 1998.

[4] B. Aziz. A static analysis of the applied pi calculus. Technical Report DTR06-15, Imperial College London, 2006.

[5] R. Bergmair and S. Katzenbeisser. Content-aware steganography: about lazy prisoners and narrow-minded wardens. In *Proc. of the 8th International Workshop on Information Hiding (IH'06), Alexandria, Virginia, USA*, volume 4437 of *Lecture Notes in Computer Science*, pages 109–123. Springer, July 2006.

[6] C. Cachin. An information-theoretic model for steganography. *Information and Computation*, 192(1):41–56, July 2004.

[7] R. Chandramouli. Mathematical approach to steganalysis. In *Proc. of Security and Watermarking of Multimedia Contents IV, San Jose, California, USA*, volume 4675, pages 14–25. SPIE, April 2002.

[8] A. Church. A set of postulates for the foundation of logic. *Annals of Mathematics*, 33(2):346–366, April 1932.

[9] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, Nov. 1976.

[10] X. Ding, Y. Xie, P. Li, M. Cui, and J. Chen. Image steganography based on artificial immune in mobile edge computing with internet of things. *IEEE Access*, 8:136186–136197, July 2020.

[11] D. Dolev and A. Yao. On the security of public key protocols. In *Proc. of the 22nd Annual Symposium on Foundations of Computer Science (SFCS'81), Washington, DC, United States*, pages 350–357. IEEE Computer Society, October 1981.

[12] S. Fathi-Kazerooni and R. Rojas-Cessa. Gan tunnel: Network traffic steganography by using gans to counter internet traffic classifiers. *IEEE Access*, 8:125345–125359, July 2020.

[13] G. Fisk, M. Fisk, C. Papadopoulos, and J. Neil. Eliminating steganography in internet traffic with active wardens. In *Proc. of the 5th International Workshop on Information Hiding (IH'02), Noordwijkerhout, The Netherlands*, volume 2578 of *Lecture Notes in Computer Science*, pages 18–35. Springer, December 2002.

[14] N. J. Hopper. Toward a theory of steganography. Technical report, CARNEGIE-MELLON UNIVERSITY, 2004.

[15] D. Kahn. *Code-Breakers*. The MacMillan Company, 1967.

[16] G. C. Kessler. Anti-forensics and the digital investigator. Technical report, School of Computer and Information Science, Edith Cowan University, Perth, 2007.

[17] X. Liao, Q.-y. Wen, and S. Shi. Distributed steganography. In *Proc. of the 7 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP'11), Dalian, China*, pages 153–156. IEEE, October 2011.

[18] W. Mazurczyk and K. Szczypiorski. Is cloud computing steganography-proof? In *Proc. of the 3th International Conference on Multimedia Information Networking and Security (MINES'11), Shanghai, China*, pages 441–442. IEEE, 2011.

[19] R. Meng, Q. Cui, Z. Zhou, Z. Fu, and X. Sun. A steganography algorithm based on cyclegan for covert communication in the internet of things. *IEEE Access*, 7:90574–90584, June 2019.

[20] C. J. Mitchell, M. Ward, and P. Wilson. Key control in key agreement protocols. *Electronics Letters*, 34(10):980–981, May 1998.

[21] N. Provos and P. Honeyman. Detecting steganographic content on the internet. In *Proc. of the Network and Distributed System Security Symposium (NDSS'02), San Diego, California, USA*. The Internet Society, February 2002.

[22] M. K. Sarkar and T. Chatterjee. Enhancing data storage security in cloud computing through steganography. *International Journal on Network Security*, 5(1):13, January 2014.

[23] D. S. Scott. *Outline of a Mathematical Theory of Computation*. Univ. Oxf. Computing Lab, Programming Research Gp., March 1970.

[24] D. S. Scott and C. Strachey. *Towards a Mathematical Semantics for Computer Languages*. Univ. Oxf. Computing Lab, Programming Research Gp., April 1971.

[25] G. J. Simmons. The prisoners' problem and the subliminal channel. In *Proc. of the 3rd Annual International Cryptology Conference (Crypto'84), Santa Barbara, California, USA*, pages 51–67. Springer, Boston, MA, 1984.

[26] A. Yadav and A. Mathuria. Towards formal analysis of key control in group key agreement protocols. In *Proc. of the 2nd International Conference (SPACE'12), Chennai, India*, volume 7644 of *Lecture Notes in Computer Science*, pages 77–93. Springer, November 2012.

Author Biography

Dr Benjamin Aziz is a Senior Lecturer at the School of Computing, University of Portsmouth. Benjamin holds PhD degree in formal verification of computer security from Dublin City University (2003) and has research interests and experience in the field of computer and information security, with over 150 publications related to areas such as security engineering of large-scale systems, IoT and SDN security, formal methods, requirements engineering and digital forensics. He is on board several program committees for international conferences and working groups, including ERCIM's FMICS, STM, Cloud Security Alliance and IFIP WG11.3.