Journal of Network and Computer Applications 44 (2014) 183–195

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Garbled Routing (GR): A generic framework towards unification of anonymous communication systems

Shaahin Madani *, Ibrahim Khalil

School of Computer Science and Information Technology, RMIT University, Melbourne, Australia

ARTICLE INFO

Article history:
Received 29 July 2013
Received in revised form
9 May 2014
Accepted 14 May 2014
Available online 9 June 2014

Keywords:
Privacy enhancing technologies
Identity protection
Anonymous communication networks
Network architecture
Component-based design

ABSTRACT

Anonymous Communication Systems conceal the identity of the communicating parties to preserve their privacy. Various approaches exist, yet none is taking advantage of the diversity of the available solutions to offer a higher anonymity. We introduce a generic framework, a high level host architecture, that allows the mixture of various communication protocols. Our proposal relies on plugin components to offer generic message processing, and on dynamic routing schemes to offer a generic circuit establishment. The results include a potentially higher anonymity for all the networks deployed within the framework, and a pathway towards sharing user-bases and code-bases, mixing the low- and high-latency communication, and benefiting from security-by-obscurity. Blending various protocols also achieves some level of network unobservability. This paper presents the design of the generic framework, the path to its adoption, the model of two real-world systems, the analysis of various security aspects, and the experimental results.

1. Introduction

Anonymous Communication Systems (ACSs) are systems that offer collaboration between online users in order to protect their privacy. Among various available solutions, transport layer anonymisation has gained the highest popularity, mainly due to the good balance between its security and its deployment flexibility. The research activity in this domain is very rich and diverse (Danezis and Diaz, 2008; Kelly, 2009; Ren and Wu, 2010; Sampigethaya and Poovendran, 2006), and here the particular focus is on the shortcomings explained as follows.

Firstly, due to the variety of algorithms and design decisions in different systems, implementation and test of each system has been an independent practice because developers need to build everything from scratch. This translates into lower reusability as well as slower simulation, test and deployment process. Moreover, independent systems disperse both the research community and the end-users, the former hindering the development and the latter opposing the cause (Dingledine and Mathewson, 2006; Acquisti et al., 2003; Pfitzmann and Hansen, 2010). Amalgamating all the algorithms, designs, end-users and development efforts into one generic system is therefore an imperative need. The framework proposed in this paper, named the Garbled Routing (GR) Framework, addresses this need through leveraging the principles of Component-Based Design.

Secondly, practical ACSs (Dingledine et al., 2004; Freedman and Morris, 2002; Goldschlag et al., 1996; Reiter and Rubin, 1998) usually do not offer Network Unobservability (Pfitzmann and Hansen, 2010). That is, while the ACS hides the identities, it does not conceal which certain ACS the communicating parties use. This feature is desirable as it further complicates traffic analysis. GR Framework offers some level of Network Unobservability by enabling the hosted ACSs to hide amongst each other.

Thirdly, foiling timing attacks relies on the existence of dummy traffic in the network (Berthold et al., 2000b) which, in turn, imposes a significant overhead. Ideally, high-latency traffic could assist with reducing the overhead by transmitting real data. The precondition is, however, the existence of a system with the capacity for hosting different latency traffics. GR Framework facilitates such a mixture by offering an environment within which components of networks with different latencies can mix.

Fourthly, further resistance to traffic analysis may be achieved by allowing secret algorithms to operate in the system and, consequently, reducing the attacker's knowledge about the *expected behaviour* of the network routers. Such an amalgam of secret and public algorithms may also introduce new vulnerabilities which need to be thoroughly studied and analysed. Through GR Framework's design, we take a step forward by creating a framework that can potentially host secret algorithms, paving the path for future work in this area.

The schematic in Fig. 1 shows a comparison of the current practice in ACS design and how the GR Framework changes the architectural standpoint. GR Framework can be thought of as an

* Corresponding author.
E-mail addresses: shaahin.madani@rmit.edu.au (S. Madani), ibrahim.khalil@rmit.edu.au (I. Khalil).

http://dx.doi.org/10.1016/j.jnca.2014.05.005
1084-8045/

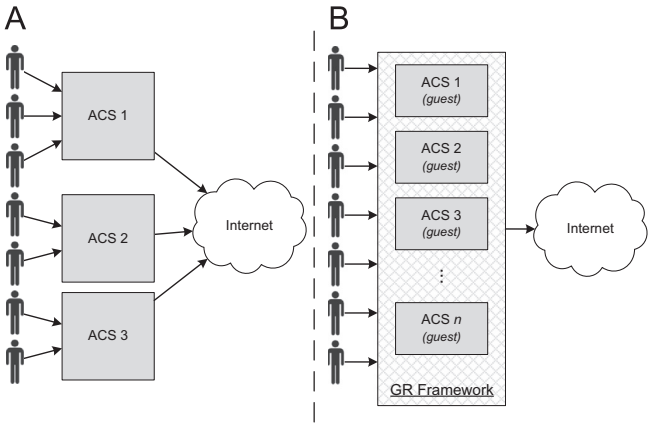

Fig. 1. Part (A) shows three Anonymous Communication Systems (ACSs) and their respective users. This is the currently existing model where user-bases and communication channels are distinguishable. Part (B) shows the GR Framework hosting the ACSs and therefore unifying the user-bases and communication channels.

anonymous system of anonymous systems. It can be seen, in a high level view, that such an architectural approach unifies the user-bases and decreases the ACS distinguishability, both of which have either actual or potential positive impacts on the degree of anonymity.

The rest of this paper is organised as follows. The following section presents background of the domain, followed by the definition of the threat model and the scope in Section 3. Various components of the GR Framework are detailed throughout Section 4. Strategies for the adoption of the framework alongside the sample scenarios are presented in Section 5. Various aspects of security analysis are discussed in Section 6, followed by the experimental results in Section 7. Finally, in Section 8, the open questions, limitations and future direction are discussed.

2. Background

Protecting the anonymity of participants in communication systems has been an important and active field of research with many different applications in, for example, election schemes (Carroll and Grosu, 2009; Park et al., 1994), VoIP (Karopoulos et al., 2010; Liberatore et al., 2011), and mobile services (Chen et al., 2011; Demestichas et al., 2009). In this paper the focus is on the Anonymous Communication Systems (ACSs) that provide generic communication over the Internet. The first nearly perfect high-latency ACS was proposed in 1981 by Chaum (1981). This work has since been rigorously examined and adopted in various systems known as mix systems (Sampigethaya and Poovendran, 2006). A *Mix* node performs a number of operations to the messages it receives (e.g., decryption, encryption, and padding), and then mixes many messages and sends them to the next node. Traffic analysis can be further hindered by using dummy traffic (Berthold et al., 2000b).

Onion Routing (Goldschlag et al., 1996) was later proposed to anonymise TCP-based, almost real-time, and bidirectional channels. In this design, message payloads are wrapped around with multiple layers of encryption, and relayed by *onion routers* along the circuits. Tor (Dingledine et al., 2004) offers improvements and new services such as congestion control and integrity checking, and has proved to be very popular. Web MIX (Berthold et al., 2000a) offers real-time traffic anonymisation, relying on cryptographic techniques to change coding of messages, and chop-and-slice algorithm to break large messages into smaller fixed-size slices. Crowds (Reiter and Rubin, 1998) builds random routing paths among a set of similar users to offer anonymous Web browsing. Tarzan (Freedman and Morris, 2002) routes are created through a small number of mixes while this system also leverages onion encrypted messages and dummy traffic. Mixmaster (Möller et al., 2003) is a high-latency anonymous email delivery system which is based on the design of Chaum's Mix (Chaum, 1981). For further details and information about other designs, the enthusiastic reader is encouraged to consult, e.g., Danezis and Diaz (2008), Kelly (2009), Ren and Wu (2010), and Sampigethaya and Poovendran (2006).

Introducing the basic terms assists with understanding how anonymity is defined and measured. A set of subjects with similar properties is called an *Anonymity Set* (Pfitzmann and Hansen, 2010); and a subject is *anonymous* if an attacker cannot sufficiently identify it within the Anonymity Set (Pfitzmann and Hansen, 2010). The *global anonymity* of an ACS is defined as the anonymity offered by the system to all of its users together (Pfitzmann and Hansen, 2010). Assuming other conditions to be equal, global anonymity of an ACS improves as a result of either growth in the Anonymity Set; or more even distribution of sending or receiving subjects within the set (Ren and Wu, 2010; Pfitzmann and Hansen, 2010; Dingledine and Mathewson, 2006; Acquisti et al., 2003). *Network unobservability* ensures that a user can join an ACS without the observer being able to identify which particular ACS is being used (Pfitzmann and Hansen, 2010).

The existing ACSs limit the potential for growth in the user-bases of each system as end-users must choose one ACS over the others. This has negative impact on the growth of the Anonymity Set and consequently on the *global anonymity* offered by each system. Additionally, the existing ACSs do not offer *network unobservability*, while there exists a potential to provide this feature by hiding different communication protocols amongst each other. There have been prior works on providing unobservability of anonymous connections amongst other Internet traffic, which rely on the cooperation of ISPs (Houmansadr et al., 2011; Wustrow et al., 2011) or popular routers (Karlin et al., 2011). We make no such assumption here, and build solely upon the cooperation of peers which is the inherent property of ACSs. We take the approach of creating an overlay convergence architecture that aims to bring the existing and future systems together.

Previous attempts towards blending the traffics with different latencies resulted exists, such as the Stop-and-Go-MIX (Kesdogan et al., 1998) and Alpha-mixing (Dingledine et al., 2006) that offer such mixing through time intervals in the mix nodes. Besides the additional aims, the GR Framework accommodates for such traffic mixing through the concept of Message Processors.

ACSs are closely related to censorship circumvention systems and there are many common design features. In fact, many users use ACSs solely to bypass censorship rather than enjoying anonymity. Integration of censorship circumvention systems, such as e.g., Collage (Burnett et al., 2010), to our framework could be a further application of our Framework and a mutually beneficial practice.

3. Threat model

Inspired by the model used for the practical low-latency systems, we assume a local eavesdropper and an attacker who can control and observe only some fraction of the network. Specifically, the attacker can monitor the communication to and from a user's computer; can add and remove arbitrary messages to the communication channels; and can run his own routers. However, the attacker is incapable of monitoring the entirety, and particularly the edges, of the network. This model takes into