The current state of denial

Deniability on secure messaging systems

Sofía Celi
Cloudflare
cherenkov@riseup.net

Iraklis Symeonidis
University of Luxembourg
iraklis.symeonidis@uni.lu

Abstract

What is deniability? Although it might sound trivial, this question has sparked debates on the privacy community ranging from legal to technical perspective. In the context of private communications, this question is notoriously difficult to approach and analyze. From a computational perspective, to answer it, one needs to look at the broader picture in which deniability applies. In this paper, we aim to provide the notions of deniability and to make more explicit the definition of deniability given in the work of Canetti et al. [2], Dodis et al. [3], Unger [5], and Walfish [6]. We do this by studying the model and by analyzing the key features and types of deniability on private peer-to-peer communications.

What our paper also aims is to emphasize on the open questions on deniability. For example, whether the current model can be generalized to group messaging, whether metadata can be deniable, or whether both coerced participants can break deniability. Additionally, we will list the means to examine the current private messaging applications as, currently, there is limited research that examines the deployed private messaging protocols. Thus, this paper aims to provide the main highlights and directions for these focus-points as an introduction to the current study in progress. Future research aims at answering the open questions and will examine how private messaging protocols approach deniability. An extended version of this paper will be published later.

1. INTRODUCTION

Probably, one of the notions that have been most debated as a property for private messaging protocols is the notion of deniability. Deniability has been debated on two fronts. On its impact on the "real-world", by, for example, examining how this property can be applied to court cases. And, on its definition as a privacy property. On the latter, historically, there have been many definitions for deniability. To this day, the term "deniability" continues to be an unsettled one. One can argue that this unsettlement persists due to the lack of a proper model, which includes the definitions of adversaries. For this, we consider two main approaches that can be used: the model outlined by Walfish [6], and by Unger [5] as an extension to it. At a simple level, it is said that a protocol is deniable if the participants on that protocol can "deny" executing a specific task as there is no plausible evidence of it.

Even though the ultimate goal of this research is to generalize the notion of deniability to any protocol that wants to achieve it, in the current paper, we will focus only on deniability for peer-to-peer private messaging protocols. We will focus on their authentication and encryption routines, but we will ask the question if it can be applied to other routines existing in the protocol at hand. We will also ask the question of deniability as part of routine interactions: if a routine is "deniable", can other routines inherit this property, or, if the other routines are not deniable, can they break the deniability of the deniable routine.

It should be stressed that deniability has been applied mainly to authentication and encryption routines during the execution of the private messaging protocols. This has been referred to as deniable authentication and deniable encryption, respectively. It remains to be seen if deniability can be applied to other tasks that protocols want to achieve and how it will interact, such as metadata protection or message franking [4]. It also remains to be investigated how deniability interacts with other privacy properties, such as anonymity. We need to stress that in this work, we do not strive to answer these questions for deniability once and for all; but rather to outline an approach for defining deniability and provide a definition according to the approach.

2. THE SYSTEM MODEL

We describe a set of entities, and we introduce relevant parties, including adversaries, as the entities that aim to break deniability.

Participants as the sender or/and receiver(s) are the communicating parties that are exchanging messages in the private messaging protocol. *Judge* is the oracle of the model that answers a decisional question: can it distinguish simulation from evidence? Essentially, it answers if the evidence presented is a true outcome of the task that is executed during the protocol run or a simulated/forged one. It is worth noting that judges can interact with the accusers/defendants, without the first knowing their true nature. Nevertheless, judges are completely rational, in the sense that they only provide an answer based on the evidence presented. *Accuser* which is analogous to what Dodis et al. [3] referred as "informant". This entity witnesses the protocol execution and tries to convince a judge that indeed a task took place during it. It can try to corrupt the participants as a way to gather evidence. *Defendant:* which is analogous to what Dodis et al [3] referred as "misinformant". It does not witness the protocol execution but tries to convince a judge that a task occurred. It creates a simulation as evidence so the protocol can remain deniable. The defendant tries to provide fake evidence to the judge so it can deem the task deniable. It can pretend to corrupt the participants as a way to mislead the judge.

As adversaries and allies, accusers and defendants can be of a certain type of models [3]. *A semi-adapter accuser or defendant* can only corrupt/aid parties before the beginning of the protocol and during the protocol execution. It means that it can interact with the participants before the execution of the protocol in order to corrupt them, or during the executing protocol. This means that a judge, in this case, acts in an online manner by interacting and instructing them to execute actions on their behalf. An "online judge" aims to distinguish between a true accuser that is interacting with them (and providing true evidence) and a fake accuser (a defen-

dant) who fabricates evidence during protocol execution. Note that the current literature focuses on the case where only one participant is corrupted, which opens up the question of what happens if both are. *A forward-secure accuser or defendant* can only corrupt/aid the parties after the protocol execution. This means that it can interact with the transcript of the protocol that has already been executed. The attackers provide evidence to a judge after the protocol execution. The judge, in this case, acts as in an "offline way" by examining the evidence presented after it happened. Note that by "protocol execution", we mean the routine that executes certain tasks by which deniability is limited. For example, this means that if a protocol executes authentication, the parties should be able to authenticate themselves, a task that is non-deniable to themselves. However, there is no evidence of this task that an accuser can use to convince a judge as a defendant could have simulated it.

3. FEATURES AND DEFINITIONS

In this work, we adhere to four notions of deniability. With these notions in mind, we will apply them to the system in which the private messaging protocols occur. Note that Walfish [6] outlines the first two notions. *Full simulatability* is when evidence used to prove that a task occurred can be computationally simulated without participants' involvement. This notion is derived from the security of the UC framework [1]. *Voluntary reveal* can be thought of as "undeniability" of a core task of a protocol. With the voluntary reveal, a protocol provides guarantees that a task executed can be undeniable to the involved parties. For example, an authentication protocol is undeniable to the parties that authenticate themselves during execution but not to anyone else. *Limit of the deniability task* in the sense that a protocol can decide what type of deniability it provides. *Indistinguishability* and *decidability*, can be thought of as the incapability of the judge of distinguishing between simulation and truth (true/fake evidence). However, it should be able to provide a decision (a yes/no answer) to the question, "can it be distinguished?".

3.1 Key features to deniability

Prior to defining deniability, we need to summarize its key features. *What is denied?*: As stated, what a protocol wants to deny to external entities is the task achieved by the protocol execution. The task cannot be denied by the participants executing it, but the evidence that this task occurred cannot be distinguished from being true or a simulation. *What is the evidence?*: As stated, the evidence is the outcome of either a simulation or a true execution of a task in the protocol. *What is the relationship to the judge/accuser/defendant?*: As stated, those three entities can interact with the task executed in the protocol in different ways. That can be summarized as in an "offline" or an "online" way.

3.2 Types and deniability definitions

In the specific scenario of peer-to-peer private messaging communication, a threat to deniability occurs when two participants (i.e., Alice and Bob) execute a task that is undeniable to them. An entity (that can be adversarial or not) provides evidence to a judge who decides if the task occurred. Given this model, a definition of deniability can be as follows: *A protocol is deniable if it allows participants to execute a task undeniable to each other. However, there exists no valid evidence of this task to other entities, as that could have been simulated.*

Given this definition, deniability can be in terms of how the evidence is presented to the judge, and the number of entities that provide evidence [5, 2].

Considering *deniability in terms of evidence presented*: *Offline deniability*: Anyone can forge a task after protocol execution between participants. Therefore, no transcript provided can show evidence of a past protocol execution because it could have been forged. *Online deniability*: If one of the participants colludes with an accuser (or is the accuser), it can provide evidence to a judge that a particular task is executed in the protocol. he judge cannot distinguish this evidence as real or forged.

Considering deniability in terms of a number of participants providing evidence: *Uni-deniability*: If an online judge requests evidence, a participant can act as an accuser or defendant to provide such proof. *Bi-deniability*: Both participants can act as either an accuser or a defendant to provide evidence. *Off-the-record deniability*: In a bi-deniability setting, the evidence provided to a judge is inconsistent with each other. A judge cannot distinguish which evidence provided by both participants is the genuine one, if any.

4. CONCLUSION AND OPEN PROBLEMS

As it has been stated throughout the paper, several open problems arise when defining deniability. These problems are related to "key features of deniability", "types of deniability", and to evaluating real-world protocols that claim to provide deniability. We briefly list the open problems. In terms of "open questions regarding key features of deniability", we can ask: can metadata be deniable? Can message franking achieve deniability? Also, how a deniable task impacts other tasks that a protocol executes? In terms of "open questions regarding types of deniability", can the definitions and types of deniability be extended to a group chat setting? Can a two-flow non-interactive protocol achieve online deniability? Also, can bi-deniability be applied to deniable authentication?. In terms of "open questions regarding the evaluation of real-world protocols that claim to provide deniability", we can ask: how a deniable task impacts the other properties provided by other tasks during protocol execution. Moreover, how this model of deniability can be used to evaluate real-world protocols.

Although the theoretical study of deniability is decades old, there are still open questions that need to be answered. Real-world protocols provide deniability that is often poorly defined. But on this paper, we outlined a definition of deniability that includes the types of adversaries that it has, that might be of aid when evaluating real-world protocols.

5. REFERENCES

[1] CANETTI, R. Universally composable security: a new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science* (2001), pp. 136–145.

[2] CANETTI, R., PARK, S., AND POBURINNAYA, O. Fully deniable interactive encryption. https://eprint.iacr.org/2018/1244. Accessed March, 2020.

[3] DODIS, Y., KATZ, J., SMITH, A., AND WALFISH, S. Composability and on-line deniability of authentication. In *Theory of Cryptography* (Berlin, Heidelberg, 2009), O. Reingold, Ed., Springer Berlin Heidelberg, pp. 146–162.

[4] FACEBOOK. Messenger secret conversations technical whitepaper. https://bit.ly/2WiuqSc, 2016. Accessed March, 2020.

[5] UNGER, N. Deniable key exchanges for secure messaging. https://bit.ly/2WgGcfC. Accessed March, 2020.

[6] WALFISH, S. Enhanced security models for network protocols. https://cs.nyu.edu/media/publications/walfish_shabsi.pdf. Accessed March, 2020.