Real-World Deniability in Messaging

Daniel Collins*
Purdue University
West Lafayette, United States
colli594@purdue.edu

Simone Colombo
EPFL
Lausanne, Switzerland
simone.colombo@epfl.ch

Loïs Huguenin-Dumittan*
Tune Insight
Lausanne, Switzerland
lois@tuneinsight.com

Abstract

This work explores real-world deniability in messaging. We propose a formal model that considers the entire messaging system to analyze deniability in practice. Applying this model to the Signal application and DKIM-protected email, we demonstrate that these systems do not offer practical deniability guarantees. Additionally, we analyze 140 court cases in Switzerland that use conversations on messaging applications as evidence and find that none consider deniability, providing evidence that this property does not have an impact in the legal setting. Based on these technical and legal findings, we assess whether deniability is a desirable property and the challenges and shortcomings of designing a system that is deniable in practice. We posit that systems should either offer real-world deniability or refrain from claiming to achieve it. We discuss how to choose an appropriate threat model for deniability in a given context and how to design communication systems that are deniable in practice. For Signal, we propose and discuss a simple yet effective solution: the application should enable direct modification of locally stored messages in the user interface. This position paper raises several unanswered questions, aiming to further stimulate discussion and research on real-world deniability in messaging.

Keywords

cryptography, legal analysis, deniability, real world, messaging, Signal

1 Introduction

Deniability, according to `www.dictionary.com`, is "the ability to deny something, as knowledge of or connection with an illegal activity". Despite the negative connotation, this definition captures the coarse notion agreed upon by researchers and practitioners: deniability enables a user to *plausibly deny* their involvement in executing some scheme or protocol. The Signal protocol [36, 38], which is the de-facto standard for secure messaging, claims to offer deniability. Moxie Marlinspike, one of the designers of Signal, discusses deniability in the context of Off-the-Record (OTR) [4] as follows [35]:

> "If someone receives an OTR message from you, they can be absolutely sure you sent it (rather than having been forged by some third party), but can't prove to anyone else that it was a message you wrote."

In the cryptographic literature, deniability is typically formalised as a game played between abstract entities. A protocol is deniable if a *judge* cannot differentiate between a real execution of the protocol and a simulated one. This means that any genuine execution of the protocol could have been faked, thereby not implicating a given party who is possibly being framed. For secure messaging, a judge might need to distinguish between actual cryptographic conversations (a sequence of ciphertexts) and simulated transcripts [5, 40, 41]. The judge might actively collude with one or more participants before the entire real transcript has been generated [15, 53] and/or have access to the cryptographic secrets of one or more parties [5, 8].

However, it is crucial to question whether these models (1) are accurate, and (2) are applicable in the real world, such as in a court of law [24]. In Section 2, we argue that neither of these conditions hold true in the context of secure messaging. For (1), existing models fail to consider the higher-level context in which parties execute the cryptographic protocol (e.g., a client that keeps messages in the device's memory and authenticates to a server), thereby rendering their deniability claims vacuous. For (2), the notions of cryptographic deniability do not align with how a human judge would interpret evidence in practice. While we do not question the *technical* value of prior work on the *cryptographic* nature of deniability, we highlight their limitations when considering practical deniability *in real-world scenarios*. In essence, cryptographic deniability is theoretically sound, but its practical implications are weakened by real-world contexts and human factors that these models do not address.

Motivated by these shortcomings, we propose in Section 3 a new model for deniability in secure messaging. Our model captures the fact that, in practice, messages are routed between users via a server that usually *authenticates* users. Consider two parties, Alice and Bob, where Bob incriminates Alice. In our model, the judge receives Bob's state (e.g., their entire phone or screenshots of the conversation) after allegedly communicating with Alice. The judge also has data from the server and any other relevant information available. The system is deniable if there exists a *practical* simulator who, under *application-specific constraints*, can interact with the server and produce a state that is indistinguishable from Bob's. This approach extends the classical notions, which only consider the cryptographic transcript, by providing the judge with Bob's state, which can include his entire phone, a portion of the server's state and arbitrary auxiliary data [41]. Our model broadens the purely cryptographic approach incorporating real-world evidence and higher-level components that can undermine deniability. By doing so, we aim to provide a more comprehensive model that better captures practical aspects of deniability in messaging systems. We note that, depending on what information the judge is afforded, conflicting conclusions can be drawn, and therefore care must be taken when modelling and assessing deniability.

*This work was conducted when the author was at EPFL.
A preliminary version of this work was presented at RWC 2023.

Proceedings on Privacy Enhancing Technologies 2025(1), 320–340

https://doi.org/10.56553/popets-2025-0018

In Section 4 we discuss how Signal is *not* deniable in practice by distinguishing two cases: normal authentication and authentication with sealed sender—a feature that hides a message's sender from the relaying server [34]. The first approach is not deniable as the Signal server authenticates the sender, whereas the sealed-sender feature fails to provide deniability because every message includes a sender-specific certificate. We also examine related work proposing solutions for the deniability of email with DKIM protection [12, 47]. In this context, the role of email domain servers—another form of relying server—also poses challenges to practical deniability. For both Signal and email with DKIM protection, we show how they can be captured in our model and how it identifies the limitations of deniability's applicability.

In Section 5 we analyze 140 legal cases in Switzerland that uses WhatsApp conversations as evidence. Since WhatsApp uses the same core protocol as Signal for two-party messaging, these conversations are cryptographically deniable. We find that (1) in only two cases the legitimacy of such evidence is questioned, (2) judges always accept this evidence, even if disputed, and (3) no case mentions or considers deniability. Although our findings cannot be generalized to other countries, our analysis, along with a similar one in the United States [60] shows that cryptographic deniability does not hold up in a legal setting. We also highlight additional results that show how the very idea of deniability is largely unknown in the legal world.

Both technical and legal analysis show that *cryptographic deniability is ineffective in the real world*. In Section 6 we discuss whether deniability should be a goal of messaging solutions by analyzing the issues and shortcomings that practical deniability brings. Given our model to analyze deniability and the analysis of technical and legal limitations, we claim that deniability should either not be a goal of messaging solutions or these must aim for *practical real-world deniability*. We argue that for deniability to be practical, it must be *easily accessible to all users*. Under our notion, Signal would achieve deniability if the application allowed users to *modify, insert or delete messages stored on their devices*, enabling *all* users to simulate conversations *in practice*. This approach provides concrete guarantees since, as our legal analysis shows, screenshots or compromised phones are generally considered authentic [42, 60], and are likely more tangible to a judge than an abstract simulator. If deniability is a goal, we advocate to implement message modification on the local device, in Signal and other secure messaging applications. We also discuss the risks that such editing capabilities entail.

To summarise, we make the following contributions:

- We propose a model for real-world deniability in messaging.
- We analyze the Signal application and show that it does not provide adequate deniability in practice.
- We examine solutions [47] for deniability of email with DKIM protection and show that they fail to provide deniability in some cases.
- We analyze 140 Swiss court cases that use WhatsApp chats as evidence and find no case that considers deniability.
- We argue that either deniability should be set aside as a security property or made practical. For Signal we propose a drastic solution: users should be able to modify all messages stored on their device.

This paper contains several opinionated claims about deniability. Our goal is to discuss deniability in the real world and to stimulate further discussion on this subject within the privacy-enhancing technologies community and beyond. We acknowledge the efforts made by Signal to provide a highly secure and private messaging solution, for example by not storing any user information metadata [44, 45], aiding deniability in practice. Our aim is to push these boundaries even further. Similarly, we recognize the work that both researchers and practitioners have done to improve the security of messaging systems and their deniability.

2 Background

This section introduces deniability in messaging and discusses previous research on what we call *meta-deniability*, which explores to what extent and how deniability is a desirable property.

2.1 Deniability in messaging

We assume two parties that use a secure messaging protocol supported by a public key infrastructure and a logical server that routes messages. Messaging solutions typically consist of two main components: an initial key exchange and the actual messaging protocol. Signal implements this initial key exchange using X3DH [36] or its post-quantum variant PQXDH [33][1], and messaging with the Double Ratchet algorithm [38]. The latter regularly updates keys to protect communication in the case of state exposure.

In messaging, deniability usually refers to the ability of a party to deny interaction with another party. The conceptual simplicity of deniability conceals multiple nuances, leading to a plethora of definitions (cf. [5, 11, 14, 18, 20, 27, 30, 40, 41, 54] for a non-exhaustive list and the work of Brendel et al. for a survey [5, Appendix A]).

The usefulness of deniability has been debated before, for example in the context of OTR in 2014 [29]. We discuss this matter in Section 6. While some issues and arguments that we mention have been raised before, we aim to provide a more structured, up-to-date and thorough perspective. Below we broadly survey the literature on secure messaging and argue that existing approaches to achieve deniability are not practical.

Two main flavours of deniability appear in the literature: *online* deniability [15, 52, 53] and *offline* deniability. Online deniability refers to settings where the judge can interact with parties *during* the execution of the protocol. In contrast, offline deniability applies when the judge receives information only after all relevant communication has ceased. Unger and Goldberg conjecture the incompatibility of online deniability with asynchronous key exchange protocols like X3DH [52]. Moreover, it is questionably applicable to many practical scenarios. For instance, in court, evidence pertains to *past* events. The scenario where a judge actively colludes with a party to frame another seems unreasonable if the judge is honest and futile otherwise, as the judge can anyway rule against the victim. Additionally, if the judge considers current events and can mount a (remote) shoulder surfing attack, human factors invalidate online deniability. The judge can instruct the incriminating party to frame the victim, e.g., by asking a question that only the victim

[1]The latest versions of Signal use PQXDH for new chats initiated after both clients use an appropriate version [32]. Since chats may last for years, we take into account both algorithms.

can answer. This suggests an analogy between online deniability and the Turing test [49], but we leave this parallelism for future exploration. We therefore consider offline deniability in our model: the data pertaining to an interaction between parties is presented to the judge *after* the interaction has concluded.

Fischlin [21] considers and formalizes relaxed cryptographic notions of deniability such as content deniability (denying that a given message was sent, but not necessarily that an interaction took place) and time deniability, and is thus able to reason about the deniability of messaging systems like OTR [4].

Vatandas et al. [54] analyze the deniability of the X3DH protocol. They prove that the protocol is offline deniable using knowledge of exponent-style assumptions [3] which seems inherent in their model. Fiedler and Janson [20] show that PQXDH's deniability can be proven secure under the same kinds of assumptions. Both simulators are "non-constructive", i.e., the simulator exists but it is impossible to show how it works, making their practicality unclear.

Despite these impractical assumptions, the works of Vatandas et al. and Fiedler and Janson are the only ones we are aware of, in the context of X3DH and PQXDH, to consider all cryptographic information produced by the protocol that can frame a party to the execution of the protocol itself. Their definitions of deniability consider signed key bundles as auxiliary input given to the adversary in both the real and ideal cases. These signatures seem to have been abstracted away in previous work on deniability [5, 28], limiting the guarantees of these deniability notions, as signatures frame parties to protocol executions (as discussed in Section 4.2 for DKIM-protected email) and preclude strong forms of deniability [5, 28].

Although interesting from a cryptographic perspective, these approaches effectively restrict the transcript to the *cryptographic* material that parties exchange *during the protocol execution*. This limitation affects the *practicality* of deniability, since it ignores a plethora of additional cryptographic and non-cryptographic information that can frame a party to a protocol execution. Even in definitions where auxiliary input is given to the judge, this data is either ignored in the analysis or only instantiated with protocol-level cryptographic material (as for X3DH/PQXDH above), and no framework is provided to reason about the system as a whole. Modern messaging solutions such as Signal rely on a complex ecosystem of communication protocols, cryptographic solutions, and asynchronous multi-device management. This stack provides a modern and user-friendly messaging experience, but comes at the cost of deniability.

2.2 Meta-deniability for messaging

This section discusses related work on deniability in messaging from a *user perspective*. Assuming that most current messaging solutions achieve some sort of cryptographic deniability, these works focus on how deniability is perceived in the real world and the missing steps to achieve practical deniability, or *human*, deniability.

Reitinger et al.'s survey study [42]. In a recent work [42] (concurrent to the first preliminary version of this work available here), Reitinger et al. conducted a user study involving US-based participants to investigate the requirements for achieving practical deniability given a messaging protocol featuring cryptographic deniability. Participants acted as a jury in a (role-played) courtroom and were shown a screenshot of a Signal conversation suggesting a politician took a bribe. They had to decide whether the defendant was guilty or not in different scenarios:

(1) *No defense*: The defense claims that the screenshot is fake.
(2) *Expert*: Renowned cryptographers explain to the jury that the protocol is deniable and therefore there is no proof that the message, and thus the screenshot, is legitimate. The study's authors apply this scenario using two approaches, namely, with jargon or friendly parlance. In the former, the experts argue using technical terms, while in the latter they use less technical and more understandable language for non-experts.
(3) *Tool*: Any user of the messaging application can edit anything in the chat, e.g., insert or modify messages or modify delivery times. The authors define three sub-scenarios: (3.1) the tool exists and the participant is given a fake screenshot; (3.2) anyone can use the tool and the participant gets the same fake screenshot; (3.3) the tool is available to the participant, who can actively modify the screenshot (Fig. 3 in Section A shows the tool that the participant can use).

The study shows that while 71% of the participants would decide *guilty* in the no defense scenario (1), this number drops to at most 26% in the expert (2) or tool (3) cases. Additionally, the study reveals that the active forgery tool is the most convincing scenario, i.e., participants who can use the tool are significantly more likely to believe that screenshots can be forged. These results show that cryptographic deniability is effectively not accepted or understood. While cryptographers might assume that without cryptographic evidence all accusations must be deniable, this study suggests otherwise: denying authorship of a screenshot does not convince people that it is fake, even in the absence of cryptographic or other evidence. People's perception of deniability does not necessarily align with the formal cryptographic perspective. On the other hand, the study shows that the presence of a practical forging tool increases the likelihood of achieving plausible deniability.

Yadav et al.'s survey study [60]. Yadav et al. conducted a study on deniability by analyzing three components: support of deniability in the messaging application, social acceptance of deniability and legal recognition of deniability. Like the study of Reitinger et al. all the participants of this study were based in the United States.

Their analysis of social acceptance of deniability reveals that participants tended to trust conversations from a messaging application more than oral claims made by a participant in a conversation. This implies that deniability is not socially accepted: if it were, participants would place the same amount of trust in both oral and digital claims. The lack of social acceptance is likely influenced by lack of knowledge about this property. The study shows that only 0.6% of participants accurately interpreted OTR's deniability definition, while 64.8% believed they understood it actually did not. Additionally, 32% of participants found the deniability definition self-contradictory, highlighting confusion and misinterpretation.

The survey indicates that most users prefer non-repudiation, i.e., non-deniable digital communications. Approximately 60% of users desire only non-repudiation, whereas 12.7% and 4.5% prefer deniability or anonymity, respectively. The remaining 22.6% seek some combination of these properties. Participants cited instances

where non-repudiation was a requirement (98%) and emphasized its importance (82%), while deniability was less frequently needed (60.94%) or deemed highly important (23.18%).

This study provides evidence that cryptographic deniability has not been considered in US court cases involving WhatsApp chats as evidence. Out of 228 cases analyzed, none presented an argument for cryptographic deniability. Instead, judges demanded concrete evidence rather than accepting claims of message forgery. There is a need for court cases that present valid technical arguments for deniability in real-world scenarios to determine whether deniability will gain legal acceptance.

2.3 Additional related work

Celi and Symeonidis [7] presented a talk on deniability at HotPETs 2020, with a particular focus on messaging. The authors discussed a number of open questions regarding deniability in messaging, most of which remain unanswered today. To the best of our knowledge, no extended version of this work has been published.

Different works propose methods to balance non-repudiation and deniability in various settings, such as message franking [2, 25, 31, 50] and the recently introduced concept of retroactive avowal [55].

This work draws inspiration from other research that model legal and real-world scenarios using cryptography. Cohen and Nissim [10] formalize aspects of the European privacy law. Garg et al. [23] formalize the right to be forgotten. Frankle et al. [22] propose using zero-knowledge proofs to enable public audits of warrants issued confidentially by intelligence courts. Scheffler and Varia [43] study the guarantees of cryptographic protocols against compelled disclosure through self-incrimination.

3 Our model

This section introduces our deniability model, which captures deniability *in practice*. We depict our model in Fig. 1, and adopt the real/ideal paradigm: the judge must differentiate between the ideal and real worlds. Our model captures the interaction between two parties, Alice and Bob, and aims for offline deniability. In this scenario, Bob, the incriminating party, hands over his state to the judge *after* the protocol execution to frame the victim, Alice. We assume that both parties execute the protocol *honestly*, meaning that Bob does not deviate from the protocol to incriminate Alice. This does not mean that Alice and Bob have to send everything through their own actions: e.g., Bob can induce Alice to send a ciphertext over the wire if the protocol implements read receipts [37, 56].

In the real world (Fig. 1, left), Alice and Bob execute the protocol using an external server. The server is a logical entity that can represent several physical machines in a centralized or distributed setting. Both parties input an initial state (st_A and st_B) that contains, for example, their long-term private key and the counterpart's public key. We do not force a type for a party's state: it can be anything, e.g., the entire phone. The real world returns Alice's public key pk_A, Bob's secret key sk_B (which is possibly blank, i.e., $\perp$), a function of the server's internal state $f(\mathsf{st}_S)$, and Bob's final state st'_B. The function f models different types of leakage from the server. For example, a subpoena might force the maintainers to reveal some of the server's data, in which case f outputs only a partial view of the server's data. Alternatively, if the server is completely breached, f is the identity function.

The ideal world (Fig. 1, right) encompasses three actors: an oracle, a simulator (represented by the person with the virtual reality headset in Fig. 1) and a server. The server is an exact replica of the one in the real world. The simulator inputs the victim's public key, i.e., Alice's pk_A, and interacts with the oracle to produce a simulation of Bob's final state. The goal of the simulator, in collaboration with the oracle, is to demonstrate that Bob can reach a simulated final state $\widetilde{\mathsf{st}}_B$ starting from an initial state st_B *without interacting with Alice* (i.e., without knowing st_A). The simulated state $\widetilde{\mathsf{st}}_B$ must be indistinguishable (Definition 1) from the real final state st'_B, which Bob returns in the real world. To this end, the oracle inputs Bob's initial state st_B, interacts with the simulator and the server and, outputs $\widetilde{\mathsf{st}}_B$ at the end of the experiment. The simulator does not directly output the simulated state: state evolution is mediated through the oracle, which complies with the simulator's instructions while considering practical constraints.

The ideal world outputs Alice's public key pk_A, Bob's secret key sk_B, some leakage of the server $f(\widetilde{\mathsf{st}}_S)$ for some function f, and Bob's simulated state $\widetilde{\mathsf{st}}_B$. As Definition 1 states, the judge J, with the help of some auxiliary information—that is, any prior or contextual information the judge might have—must distinguish between the outputs of the two worlds.

Definition 1. *We say that a system using a server S, coerced to reveal information that the function f represents, is* deniable *if, for any victim A and any incriminating party B, there exists a simulator with access to an oracle $\mathcal{O}$, such that for any judge J with auxiliary information* aux*, the following holds:*

$$\left|\Pr\left[J(\mathsf{aux}, \mathsf{pk}_A, \mathsf{sk}_B, f(\mathsf{st}_S), \mathsf{st}'_B) \Rightarrow 1\right] - \Pr\left[J(\mathsf{aux}, \mathsf{pk}_A, \mathsf{sk}_B, f(\widetilde{\mathsf{st}}_S), \widetilde{\mathsf{st}}_B) \Rightarrow 1\right]\right| \leq \nu,$$

where ν is a value that models the in dubio pro reo *principle: if the judge is* unsure*, to some degree of uncertainty, then they rule in favour of the victim A. The exact value of ν depends on the context in which our definition is used.*

Remark 1. *Our definition captures honest protocol executions between Alice and Bob. Achieving deniability in the malicious setting is very difficult if not impossible: Bob can circumvent deniability such as with remote algorithm substitution attacks [1] or remote attestation [26]. Remote attestation leverages hardware-based trusted execution environments, which are widely available in today's mobile phones, to produce a publicly verifiable and non-deniable transcript from the execution of a deniable protocol; the victim does not even detect the loss of deniability, as she runs the original deniable protocol.*

Remark 2. *Unlike many cryptographic models that formally define variable types, our model requires these variables to be defined before analyzing practical deniability. In many cases, the accuser (Bob) can determine his state, e.g., by surrendering his entire phone to the judge. Similarly, Bob and/or the judge often control the function f that models leakage from the server, e.g., the judge can subpoena the server or the service can disclose useful metadata to the judge. The variables of the model must therefore be defined* before *applying it to a specific system, taking into account the* worst case *that the threat model encompasses.*

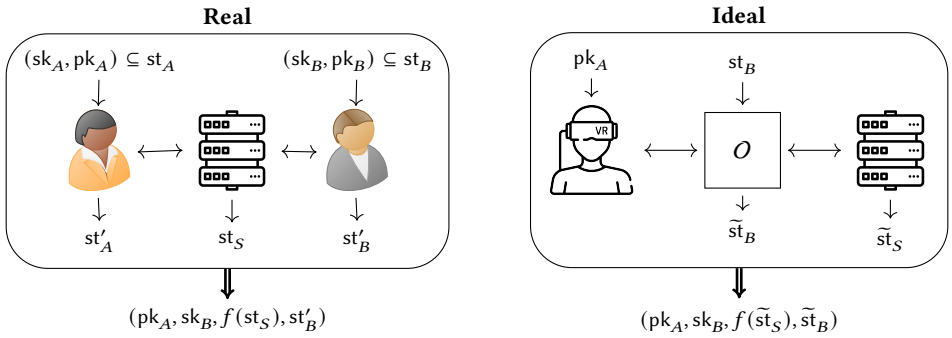

Figure 1: Our model for real world deniability in a two-parties protocol. The simulator is depicted by the virtual reality device. The application-specific oracle O models the ability of the simulator to modify Bob's state. The judge, aided by some auxiliary material aux**, must distinguish between the output of the real and the ideal worlds (see Definition 1).**

Oracle NO-AUTH(op, payload)

```
if op = get then
    send st̃_B to simulator
elseif op = set then
    st̃_B ← payload
elseif op = forward then
    send payload to server
    send server's answer to simulator
elseif op = finish then return st̃_B
else return ⊥
```

Oracle AUTH(op, payload)

```
if op = get then return ⊥
elseif op = set ∧ Vf(st̃_B, payload) then
    st̃_B ← update st̃_B according to payload
elseif op = forward then
    authenticate to server using st̃_B
    send payload to server
    send server's answer to simulator
elseif op = finish then return st̃_B
else return ⊥
```

Figure 2: Two oracles for our deniability model. The NO-AUTH **oracle models a system *without* authentication,** AUTH **models *signature*-based authentication.**

3.1 Model parameters

Our model provides a large degree of flexibility and can capture many different scenarios. To this end, we discuss below how different parameters of Definition 1 may be chosen before describing and assigning variables in two example scenarios. In some cases, there may be some overlap between variables, e.g. Bob's state may contain his secret key.

Alice's public key (pk_A). This should typically correspond to Alice's long-term public key that she registers with some public key infrastructure, for example the Signal server.

Bob's secret key (sk_B). This typically corresponds to Bob's long-term secret key, but if Bob chooses to not disclose this to the judge, it may be empty ($\perp$).

Server state leakage ($f(\mathsf{st}_S)$). As we consider offline deniability, st_S corresponds to the state of the server after the relevant interactions between Alice and Bob have taken place. The leakage function f depends on the what the server is willing or compelled to disclose and what the judge has access to. In a legal setting, for example, WhatsApp discloses significantly more information than Signal [19, 57] In some settings, this could capture information that has been leaked from the server to, e.g., the general public through whistleblowing or compromise. In some cases, the judge may not have access to any pertinent information from the server. Although our definition is tailored for the case where Alice and Bob are communicating via a server, direct communication could be captured by considering the 'trivial' server that leaks nothing.

Bob's final state (st'_B). As with Bob's secret key, this can depend on Bob's level of cooperation with the judge (or indeed what they are compelled to disclose), and depends on context. For example, this could comprise of Bob's message database, their cryptographic state, the entire contents of their phone or something in between.

Auxiliary data (aux). This corresponds to any additional information that the judge may have access to, e.g., testimonies, or more generally information about Alice (e.g. their public profile), Bob and their conversation.

The judge (J). The judge corresponds to any entity that wishes to evaluate the deniability of a given execution. In a courtroom, the most classic setting, More broadly, this can capture entities like the general public (or particular individuals) who are determining whether some interaction was real or not (cf. Theorem 3.2). In principle, the judge could be anyone.

The oracle (O). The oracle models the simulator's practical effectiveness in simulating Bob's state and communicating with the server: the initial state might be encrypted with some unknown secret key and the server might require the client to authenticate with it. It captures the level of control Bob has over his client. For example, at one extreme, if Bob has rooted his phone, he can extract all secrets and messages contained within, making the oracle very permissive. At the other extreme, if Bob knows nothing about technology and is limited to using the client to try to forge messages, the oracle is much more restrictive.

Fig. 2 shows two instantiations of the oracle O which capture many realistic settings: at the top without authentication and at the bottom with authentication. Both oracles input an operation $\mathsf{op} \in \{\mathsf{get}, \mathsf{set}, \mathsf{forward}, \mathsf{finish}\}$ and a corresponding payload that the simulator wishes to send to the server. The oracle that mimics authentication checks the payload's legitimacy using the verification function Vf before updating the state. The payload, for example, can be a message encrypted with TLS, which is decrypted and stored in the state only if it comes from the correct server. The simulator interacts with the oracle to simulate Bob's state and communicates with the server while enforcing the authentication mechanism.

AUTH captures the interaction of the vast majority of users who use a given messaging application as intended and execute the protocol honestly. NO-AUTH, on the other hand, captures a technically-savvy or resourceful user who is able to modify their state directly. The behaviour of the oracle also may vary over time, e.g. in the case of KeyForge for DKIM (cf. Section 4.2).

Real-world examples. We provide two examples to highlight how the modelling process can occur. They also highlight how the judge's role can change depending on where and against whom deniability must hold; we expand further on this aspect in Section 6.2.

Example 3.1. Suppose Bob accuses Alice of sending a particular message in a classic courtroom setting with a single judge. Alice and Bob communicate using Signal and they do not know each other, i.e., they have never had any contact before. In this case the variables in the model are defined as follows:

- pk_A: Alice's public key in Signal.
- sk_B: Bob's secret key in Signal.
- $f(\mathsf{st}_S)$: Alice and Bob's last connection data to Signal and the data of the creation of Alice and Bob's contacts, all in Unix timestamp.
- st'_B: Bob's phone, since he decides to surrender it to the judge when accusing Alice.
- aux: since Alice and Bob do not know each other, we can assume that $\mathsf{aux} \leftarrow \varepsilon$, where ε represents the empty string.

The judge J is the judge of the courtroom in which Bob accuses Alice. The justification behind these choices of variables is mostly self-evident, except for the choice of f, which we chose as these timestamps comprise the only information that Signal claims to and has previously disclosed in subpoena requests.[2]

Example 3.2. Suppose Bob takes a screenshot of what he claims to be his conversation with Alice and posts the image on a public bulletin board. The conversation is on Signal and contains some insults that Alice uttered towards Bob. Shortly after, Carol posts a comment to the image where she says that she once received the same insults from Alice once. In this case the variables in the model can be defined as follows:

- pk_A: Alice's public key in Signal.
- sk_B: $\perp$, since Bob's secret key is most likely not given to the judge (see below).
- $f(\mathsf{st}_S)$: In this case $f(\mathsf{st}_S) \leftarrow \emptyset$ since no one sent a subpoena to Signal.
- st'_B: The screenshot that Bob publishes on the bulletin board.
- aux: The auxiliary information contains at least the comments that Carol posts on the bulletin board after Bob's screenshot. Additional information might exist, depending on the people acting as judges.

The judge J is in this case the group of people that consult the public board, thereby seeing Bob's screenshot and Carol's comment.

3.2 Limitations and pitfalls

Although our model is able to capture many scenarios, it nonetheless has drawbacks. Firstly, the model only directly captures two-party communication, so cannot be readily used for group messaging or in multi-party settings. In addition, it only captures offline deniability and honest executions (which we nonetheless argue above is reasonable).

Our model leaves many parameters and variables undefined. Whilst this allows it to capture a variety of settings, care must be taken when it is used. In particular the model requires domain knowledge and fine-tuning to be used effectively, but we see this as inherent in assessing deniability in the real world. Whilst the model is not fully formal, it nonetheless provides a framework for thinking and reasoning about deniability in the real world.

Moreover, care must be taken when drawing conclusions from the model. Suppose that Alice and Bob are messaging, and Bob is able to modify his received messages without leaving any trace of tampering from the perspective of the judge (e.g., by modifying the set of stored messages on another device and then syncing). Then, Bob has successfully framed Alice, and so the conclusions drawn from the model in the real world would be incorrect (even if they are logically consistent). Thus, when using the model, one must ensure that the variables given to the judge are as numerous as possible. In this example, additional auxiliary data or server leakage could catch Bob's tampering, or alternatively ensuring that Bob's state is comprehensive (as here he excluded critical information). If, due to contextual constraints, that some information cannot be feasibly given to the judge, then it is important to be cautious and determine what kind of additional information *could* exist, and what impact that would have on deniability. For this reason, we advocate for the *in dubio pro reo* principle to be applied when interpreting deniability in our model, as it is usually done in the legal setting.

4 Technical case studies

This section analyzes two real-world communication solutions and their deniability properties: the Signal application and KeyForge, the latter being system that achieves a form of deniability for email

[2] See https://signal.org/bigbrother/.

communication while maintaining the protections that DKIM provides [47]. We describe both systems and we apply our deniability model (Section 3) to them. In departing from messaging to discuss another communication medium, we show that practical deniability remains hard to achieve even with practical countermeasures against non-repudiation.

4.1 The Signal application

We focus on Signal because it offers the best security and privacy guarantees among deployed secure messaging solutions. Moreover, the cryptographic algorithms underlying Signal are deniable under some notions of cryptographic deniability [36, 38, 54] and much of the code is open source. We distinguish two cases: normal authentication and authentication with the sealed sender feature [34]. To conduct this analysis we analyze the Signal server source code [46].

4.1.1 Normal authentication. To send a message, the sender's client issues a POST request to the server's `/v1/messages/{receiver}` endpoint with a payload that contains the encrypted message and the receiver's identifier. The server authenticates the sender using Dropwizard [16] basic authentication, which authenticates the sender's identity and a password (a secret shared with the Signal server). After authentication, the server performs a few extra checks (e.g., rate limiting) and creates an "envelope" that contains the authenticated sender's identity, the encrypted message and a timestamp. The server forwards this envelope to the receiver or stores it until the receiver is online.

Deniability in practice. The description above implies that a message *delivered* by a legitimate and honest Signal client originates from the claimed sender, if the Signal server operates honestly. Thus if someone inspects the receiver's device, they can be confident that the message indeed comes from the sender. If the server is honest and not compromised, the only deniability argument for the sender is to argue that the receiver tampered with the stored messages, i.e., the receiver edited the received messages in the local database. Depending on the device, modifying the database can be challenging (e.g., it requires basic technical knowledge on the desktop client as the database encryption key is stored in clear[3], while advanced knowledge is required for mobile applications). Alice's deniability claim depends therefore only on Bob's technical knowledge. Depending on the situation, Bob can deny having the ability to tamper with the local Signal database and, in general, this argument is not equally available to every potential victim.

Model variables. To capture Signal, variables in our model can be set as follows.

- pk_A: Alice's long-term public key in Signal.
- sk_B: Bob's long-term secret key or $\perp$, depending on the power of the judge.
- $f(\mathsf{st}_S)$: $\emptyset$ or two timestamps (depending on whether a subpoena was sent to Signal or not, cf. Theorem 3.1). This assumes that that the Signal server behaves honestly.
- st'_B, aux and the judge: These are highly context-dependent, depending on how much Bob cooperates, how much information the judge has access to, and who or what exactly the judge is, respectively.
- Oracle O: Depending on Bob's (technical) ability to modify his state, either NO-AUTH (if he can do so) or AUTH (if he cannot). In particular, AUTH captures the fact that Signal uses authentication as described above.

This also applies when considering sealed sender in the text below.

In our model

If there is authentication (AUTH oracle in Fig. 2), Signal is not deniable as Bob's state (comprising the received messages) is updated only upon receiving a message from the Signal server that certifies Alice sent it. As described above, this occurs only if the server authenticates Alice before she sends a message, which the simulator cannot do as it does not know Alice's credentials. In particular, in the AUTH oracle in Fig. 2, the verification function on line 2 fails because payload does not not contain the necessary credentials, i.e., Alice's, to update Bob's simulated state $\widetilde{\mathsf{st}}_B$. This results in two different states st'_B and $\widetilde{\mathsf{st}}_B$, giving the judge a distinguishing advantage of probability one in Definition 1. Note the judge requires no auxiliary information for this attack to succeed.
We formalize Bob's modification of the local database using the same AUTH oracle. The payload that the simulator provides to the oracle includes the database encryption key and the verification function Vf returns true, enabling the oracle to modify Bob's simulated state, i.e, the messages' database. If the oracle NO-AUTH is used, the simulator can modify Bob's state to make it consistent with some messages received from Alice. Hence the deniability claim relies solely on the receiver's ability to tamper with the local database (barring any incriminating evidence otherwise available to the judge).

Finally, a malicious client can prove that they indeed received the message from a Signal server by using DECO [61], which enables clients to prove that data received via TLS originated from a particular server. For this and other reasons, which we present in Section 3, we exclude malicious clients from our analysis.

4.1.2 Authentication with sealed sender. Authentication with sealed sender conceals the identity of the sender from everyone except the receiver, *including* the Signal server. This protects the sender's identity at the *application level*, but does not hide other sensitive information, such as IP addresses. By default, this option is only enabled between mutual contacts and works as follows. Periodically, clients retrieve a certificate signed by Signal that contains (1) the client's identity, (2) the client's phone number and (3) an expiration date. In addition, each party derives a *delivery token* from their profile key and registers it with Signal. To send a message, the sender's client encrypts the certificate, the message and the sender's identity. The client forwards the resulting ciphertext, the

[3]Joshua Lund from Signal comments on this on the SignalCommunity forum: "The database key was never intended to be a secret. At-rest encryption is not something that Signal Desktop is currently trying to provide or has ever claimed to provide. Full-disk encryption can be enabled at the OS level on most desktop platforms" (https://community.signalusers.org/t/vulnerabilities/4548/7). On Linux, the database encryption key is stored in `~/config/Signal/config.json` and the database in `~/config/Signal/sql/db.sqlite`.

receiver's identity, and the delivery token to the server using the same endpoint as in normal authentication. The server checks that the token corresponds to the intended recipient, and forwards the ciphertext when the receiver is online; the server does not authenticate the sender. Finally, the receiver decrypts the ciphertext and verifies that the certificate is valid and corresponds to the claimed sender's identity.

Deniability in practice. This setting offers brighter prospects for deniability than normal authentication. As the server never authenticates the sender, receiving a message from Alice that was correctly forwarded by the server does not mean she actually sent it. However, the fact that the message includes the sender's certificate hinders their capability to deny. Let's say the victim Alice claims that Bob (1) forged a message coming from her, then (2) made the Signal server forward that message to himself and finally (3) his client successfully delivered it. It means that Bob managed to recover one of Alice's certificates that is valid for that specific time frame. In turn, that implies that he (or an accomplice) recently received a message from Alice and extracted the certificate to forge a valid message. Therefore, in practice, one can be very confident that Alice exchanged messages with Bob recently, which weakens plausible deniability. Alice can still deny that she sent a particular message, but the claim relies solely on the fact that Bob had the technical knowledge to extract the certificate and forge a valid message or, as in normal authentication, the capacity of tampering with the messages stored on his client.

In our model

Transposing this argument in our model, we see that Signal with sealed sender is not deniable in the authenticated setting ($\mathcal{O}$ = AUTH) as Bob's state will accept Alice's message only if a legitimate certificate was provided. However, the simulator cannot obtain Alice's certificate from the server as it authenticate as Alice: the verification function Vf returns false and Bob's state is not updated. In the setting where Bob can freely modify the set of received messages ($\mathcal{O}$ = NO-AUTH), Signal with sealed sender is deniable.

4.1.3 Summary and discussion. The above discussion shows that plausible deniability is practically non-existent in Signal unless the receiver has good technical capabilities. To summarise:

- Using normal authentication, if the client delivers a message from Alice, then it surely came from Alice because the server authenticated with her.
- Using sealed-sender-based authentication, if Bob's client delivers a message from Alice, then with good probability Bob received a message from Alice recently. Bob could forge a message coming from Alice by asking for Alice's certificate from one of Alice's contacts, but we deem this to be relatively impractical.
- In both modes of authentication, deniability is plausible only if the receiver has good technical knowledge to modify the local database of received messages, i.e., to modify the state of the incriminating party ($\widetilde{\mathsf{st}}_B$ in our model). If the incriminating party lacks technical knowledge, the judge will hardly believe the victim's denial of participation.

We conclude that, in most cases, the Signal application does not provide plausible deniability.

It is worth noting that the deniability claim becomes stronger as the system becomes easier to breach, and vice versa. For instance, in a perfectly "secure" system (e.g., impenetrable clients and interactions only through authenticated clients as in our model with $\mathcal{O}$ = AUTH), Signal is not deniable. This creates an undesirable situation: certain aspects of security clash with deniability. We propose that the best way to address this problem is to either give up on deniability or to incorporate "cracking" as a controlled and desirable feature. With cracking we refer to the usually illicit act of circumventing or overcoming security measures, such as accessing the protected Signal's database and modify its content, or compromising Signal's server to send a message under a spoofed identity. For example, Signal could include an edit/add button that allows the user to modify received messages or add new ones. The user interface could look like the interactive forging tool that Reitinger et al. used for their study (Fig. 3). While this could lead to other problems as discussed in Section 6, we claim that if we want deniability in Signal and other messaging application, then we should aim for human and practical deniability instead of just cryptographic deniability, which, as we show, does not translate into real-world deniability. Overall, this form of deniability can only be guaranteed if a practical, user-friendly simulator exists, which is currently not available. We continue the discussion about how to achieve practical deniability and possible shortcomings in Section 6.

4.2 DKIM and KeyForge

Another interesting case study for deniability is email-based communication. Here, we assume that emails are not authenticated with PGP or S/MIME: in such cases the email is cryptographically signed by the author and thus the fact that it was sent is publicly verifiable and undeniable.

We focus on email protected with DomainKeys Identified Mail (DKIM) [12], which authenticates the source of an email by verifying the domain name from which it originated. If an email is sent from `first.com` to `second.com`, the server of `first.com` cryptographically signs the email (including message and headers) and adds the signature to the email's headers. When the email reaches `second.com`, the server verifies the signature using `first.com`'s public key and delivers the email only if the verification is successful. If the DKIM signing keys are kept private, the signature proves that the message was not altered during transit and that it was sent by an authorized sender, thereby preventing malicious activities such as spoofing, phishing and spam.

As the leak of Hillary Clinton's emails [59] shows (see Section 6 for further discussion), DKIM protection devastatingly impacts email deniability. The digital signature provides a cryptographic assurance that the sender is who they claim to be, unless email accounts are breached or secret keys are leaked, which we deem

unlikely[4]. Using our model (Section 3), DKIM-protected email is not deniable since we need an oracle like AUTH to model this setting: the only way to update Bob's state is to have control over the signing keys of all domains between the sender and the recipient, unless another breach occurs.

To remedy this situation, Specter et al. propose KeyForge [47], which enforces that DKIM signing keys are released after a predefined delay, which the authors recommend to set at 15 minutes. To this end, Specter et al. formalize and instantiate a new cryptographic primitive called forward forgeable signatures. One instantiation of such signatures is KeyForge, which builds on a so-called hierarchical signature scheme. This approach preserves the protection of DKIM during the delay, but enables anyone to forge DKIM signatures after the delay, thereby achieving a form of deniability. However, during the delay, unless Alice's account is compromised, Bob cannot spoof Alice's identity to authenticate to the server, precluding deniability.

To address this, the authors of KeyForge propose KeyForge$^+$ [47], which includes an additional protocol called forge-on-request. This protocol enables parties like Bob to request forged emails from any domain, such as Alice's domain, under the only condition that the recipient of the forged email is the requester, i.e., Bob in this case. In other words, Bob can request to forge an email from Alice only if the recipient is Bob himself. For group emails, Bob can request a forged email from any domain. This forge-on-request protocol enables the simulator to spoof Alice's identity and authenticate to the server.

Model variables. To capture plain DKIM [12], KeyForge and KeyForge$^+$ [47], we can choose model variables as follows:

- pk_A: This corresponds to Alice's domain's public key.
- sk_B: Bob's domain's secret key or $\perp$, depending on the power of the judge, albeit it should not affect its decision.
- $f(\mathsf{st}_S)$: This depends on context - see the discussion below for how server leakage can prevent deniability.
- st'_B, aux and the judge: These are context-dependent as before. In the case of the Clinton e-mail leaks, the judge was the general public and the press (note the similarity between this scenario and Theorem 3.2).
- Oracle O: For plain DKIM, the oracle behaves like AUTH, given that the signing key is never leaked. For KeyForge for a given DKIM signing key, the oracle behaves as in AUTH until the signing key is released, after which it behaves like NO-AUTH. For KeyForge$^+$, when the signing key is not released, it behaves like NO-AUTH if a request is made with forge-on-request (note we assume parties honestly execute the protocol, so such a request should succeed), and otherwise as in KeyForge.

In our model

Forging DKIM signatures is not enough to achieve practical deniability. For Alice, the victim, to defend herself against Bob, the accuser, Bob must be able to send an email on Alice's behalf, which is not immediately possible in KeyForge, even if Alice's email is registered on a different domain. This limitation is also visible in the AUTH oracle: on line 5, the oracle uses Bob's simulated state ($\widetilde{\mathsf{st}_B}$). This is resolved in KeyForge$^+$ as described above. That is, until the signing key is public, the oracle cannot forge a DKIM signature and therefore Bob's state is not updated (the Vf function on line 2 of AUTH in Fig. 2 returns false). When the signing key becomes publicly available, anyone can forge DKIM signatures: the Vf function returns true and the oracle can update Bob's state.
The combination of KeyForge and KeyForge$^+$ shifts the oracle from AUTH to NO-AUTH after a predefined delay, thereby increasing the deniability properties of DKIM-protected email. However, our model demonstrates that the server plays a crucial role in real-world deniability. The judge inputs a function of the real and simulated server states (the former in the real world, the latter in the ideal one), where the function f models different leakages from the server. The work of Specter et al. does not account for this leakage: the server could log Bob's request of a forgery of Alice's messages differently than how it logs a real sending procedure from Alice, which would help the judge to distinguish the real and ideal worlds. Additionally, the server could use two different databases to store forgeries and legitimate emails. The judge can subpoena these databases to differentiate between the two worlds [6]. This highlights the significant role that the service's server (formally represented by the function f) plays in our model.

4.2.1 Summary and discussion. In the above discussion we saw that plausible deniability for DKIM-protected email is non-existent and how KeyForge and KeyForge$^+$ partially solve this problem. At a high level, the procedure is similar to what we propose for Signal: use "cracking" as a controlled and desirable feature. In this case this involves releasing the private signing keys after a predefined amount of time, which breaks the authentication that DKIM provides. However, as with Signal, email servers must play the game: if they log too much information about the exchanged messages and the judge obtains this information, deniability is lost.

Unlike our proposed approach for Signal, i.e., giving the ability to edit messages on the user interface level to all users, KeyForge$^+$ requires important changes to the current email infrastructure in order to be deployed, as acknowledged by the authors. This forge-on-request protocol must also be easy to use and available to everyone to avoid the unequal access to deniability that arises when forging is accessible only to tech-savvy users. Additional research is thus necessary to assess the usability of KeyForge and KeyForge$^+$.

5 Legal case studies

This section considers deniability in the legal context by presenting our analysis of court cases that mention messaging applications in Switzerland. To the best of our knowledge, this is the first such analysis capturing both the common and civil law paradigms (the studies that we cite in Section 2.2 considered participants or cases in the United States) and therefore fills a gap that Yadav et al. highlighted [60].

[4]One might think that DKIM protection increases email deniability, since anyone with access to the email server—such as an employee—can forge signatures. In reality, DKIM signing keys are kept private by email providers, as losing these keys would enable internal or external adversaries to mount email-based attacks such as phishing.

Yadav et al. conducted an analysis of 228 court cases in which WhatsApp conversations were proposed as evidence. They found that WhatsApp chats where *never* rejected as evidence because of their cryptographic deniability properties, even when the defendants claim that conversations can be forged. One illuminating example from their analysis is the United States v. Ojimbda case:

> "Defendant's objection that the text messages, in this case, are unreliable is made *without any persuasive evidence* and is thus overruled... The court explained that Mr. Ojimba could attack the reliability of the messages at trial, but that reliability was ultimately a matter for the jury."

We conducted a similar analysis in Switzerland. We start by introducing our methodology and focus later on the results.

5.1 Methodology

Our legal analysis aims at answering three research questions. We settled for WhatsApp (which is more widely used) in our research as none of the decisions we surveyed mentioned Signal. Even if WhatsApp is less deniable than Signal because it is less private, i.e., additional metadata may be available to an adversary, we believe that a court's opinion would be similar or the same for Signal or other messaging applications. This is especially true if the court does not subpoena the relevant service provider for metadata, which is true for the Swiss cases we considered. The three research questions are as follows:

(1) Do judges in Swiss courts use WhatsApp messages as evidence?
(2) When WhatsApp messages are used as evidence, is their usage contested by any of the parties involved?
(3) What are the specific reasons cited for disputing the legal validity of WhatsApp messages, and how do judges respond to these disputes?

Our hypothesis was twofold: Firstly, judges in Swiss courts use WhatsApp messages as evidence in penal cases. Secondly, the validity of these messages is generally not disputed by the parties involved, despite the cryptographic deniability properties inherent in WhatsApp's algorithm, i.e., the same as Signal.

To design our study, we consulted with four Swiss lawyers. We used the website https://entscheidsuche.ch, a private initiative that publishes decisions from all instances of Swiss courts. We queried the website with an Elastic search query (Fig. 4) to select the initial cases, i.e., penal cases in French or Italian that contain the word "WhatsApp". We opted for cases in French and Italian because we speak these languages and did not use any translation tool. We focused on penal cases to analyze those with a significant coercive impact on the defendants. As of February 22, 2024, our query returned an initial set of 419 decisions from all instances. From this initial set we removed all the decisions from federal courts by removing the corresponding URLs from the query results (e.g., `CH_BGer` for supreme court). We opted for this additional pruning for two reasons:

(1) The federal criminal court treats cases originating from the office of the attorney general of Switzerland, involving major investigations where conversations play a minimal role, if any.
(2) The federal supreme court, the highest court in Switzerland, has jurisdiction over different violations, including penal cases. However, this court does not accept the addition or removal of evidence: the set of available evidence is fixed at the first two levels of the Swiss legal system.

We thus focused on the first two level of Swiss courts with jurisdiction over penal cases, leaving us with a set of 341 decisions. We performed an additional verification of decisions from the supreme court, as it is the court most likely to rule against accepting messaging conversations as evidence if such a decision is appealed. We found no such ruling, and the lawyers confirmed that it does not exist.

We manually analyzed the 341 cases and divided them into four categories. When in doubt, we consulted with the four lawyers who helped us design the analysis. The four categories are as follows.

- N/A: The decision mentions the word "WhatsApp" for purposes other than evidence, such as to summarize police investigations or to mention that two parties can or cannot contact each other via WhatsApp. We also put outliers in this category (e.g., administrative cases that the website wrongly returns as penal ones or federal courts cases that we missed during the initial pruning).
- Evidence: WhatsApp chat is used as an evidence in the final decision.
- Contested: One of the parties contests the validity of WhatsApp conversation as legal evidence.
- Rejected: The judges decide to reject WhatsApp chat as evidence after a party contested its validity.

In Section B.2 we provide the URLs for the 341 cases (from the website https://entscheidsuche.ch).

5.2 Results

Table 1: Summary of legal case analysis results.

Total Cases	N/A	Evidence	Contested	Rejected
341	201 (59%)	140 (41%)	2	0

Table 1 summarizes the results of our analysis. We find that 59% of the cases that mention the word "WhatsApp" do not report the use of conversations as evidence. Conversely, for 41% of the decisions that we analyze, WhatsApp conversations are used as evidence. In all except one case that we discuss below, the authenticity of messages, or their legal validity, is never questioned by any of the parties involved. In particular, deniability is never invoked (successfully or otherwise) in all the 140 cases in which electronic messages are used as evidence. We find *two* cases in which a party in an appeal denies being the author of messages appearing on a screenshot. In the first decision [17] (entry (12) in the list of Section B.2) we can read (text translated by the authors):

> On the other hand, she disputes having written or sent the WhatsApp messages contained in the Justice of the Peace's file, even though they were produced by

> C._____ on November 21, 2017. She asserts that the documents could have been altered or that there might have been manipulation on the WhatsApp application regarding the name of the sender of the messages.[5]

Despite the dispute, the judges decided not to reject the evidence or pursue the matter further (for example by ordering a forensic analysis of the appellant's phone). The judges deem the appellant's argumentation not credible and they posit that their invalidation would not change the original ruling:

> Her argument is far from convincing. Under these circumstances, there is no reason to order an analysis of the parties' mobile phones. Moreover, even if such an analysis were ordered and it demonstrated that the messages did not originate from the appellant, this would still not constitute sufficient suspicion that the respondent has committed one or more criminal offenses [...].[6]

In the second decision [39] (entry (81) in the list of Section B.2), an appellant claims that "he was not the author of the messages that B_____ had received [...]"[7]. This claim builds on the idea that someone sent messages to the potential victim posing as the plaintiff, using a new phone number. As in the first case, the judges considered the arguments not credible. In particular, the content of the conversation and the "Machiavellian sense" to mount such an attack convinced the judges that the author of the messages can only be the appellant.

5.2.1 Additional results. Our analysis enabled us to find other interesting results besides the primary focus that our research questions set; here we present some of them.

WhatsApp as an investigative tool. From one of the cases that we analyze (entry (147) in the list of Section B.2), we evince that an undercover policeman used WhatsApp to communicate with a potential pedophile.

> [...] From March 10 to 29, 2021, he maintained a virtual exchange via e-mails and then WhatsApp messages with a young boy [...] who had indicated that he was 14 years old. On March 29, 2021, the appellant invited the child to a hotel [...] pretending to be his nephew, in order to have intimate relations [...]. In reality, he was conversing with a police officer.[8]

Disappearing messages as additional evidence. The "disappearing messages" feature, available in most modern messaging systems, enables users to send messages that automatically delete after a set period. We found some decisions in which the use of this feature is highlighted by the judges. In one case in particular (entry (180) in the list of Section B.2), disappearing messages represent additional incriminating evidence:

> Finally, the fact that I_____ and the appellant were initially represented by the same lawyer, on the one hand, and that their WhatsApp exchanges have disappeared from the appellant's phone, on the other hand, are additional incriminating indicators.[9]

Screenshots as evidence. In most of the cases that we analyze, the decision does not specify how parties present messages to the court. In some of them, the judges specify that parties present evidence through screenshots, without discussing or analyzing their truthfulness or chain of custody. Our analysis does not evidence any discussion of potential tampering or forging of these screenshots, for WhatsApp conversations or other evidence.

5.3 Analysis

The results we present in the previous section confirm our first hypothesis: judges use chats as evidence in penal cases. The validity of these messages is generally not disputed by the parties involved despite cryptographic deniability properties. Our study, along with with Yadav et al.'s findings [60] show that deniability seems irrelevant in court cases in Switzerland and the United States. Although the results cannot be generalised to other countries, they provide evidence that cryptographic deniability does not translate into real-world deniability and that a purely cryptographic approach does not impact the legal setting. When one party claims a forgery, judges consistently reject these claims. In what follows, we analyze possible reasons for this failure discuss potential countermeasures.

Lawyers, judges, and authorities are generally unaware of cryptographic deniability and its workings. The use of WhatsApp by the police as an investigative tool highlights this lack of knowledge. This results from society's lack of acceptance of deniability. Therefore, if deniability is a desired feature, it must be socially accepted, simple and practical for everyone. Simplicity could also limit the imbalance in access to justice when deniability could be a factor. Malicious parties could exploit deniability to forge messages that are considered real by courts, while wealthy defendants could hire experts to testify about deniability properties, an option not available to most people dealing with low-profile cases.

The two cases that we present in Section 5.2 highlight the importance of contextual information for deniability, in court or other settings. Communication transcripts are (1) one type of evidence among others with varying degree of importance (as in the first case discussed), and (2) can be authenticated given other contextual information (as in the second case). The first aspect is important: even with complete real-world deniability, judges would have the ability to convict perpetrators of particularly heinous crimes, several examples of which we encountered in our analysis.

[5] Original text: "D'autre part, elle conteste avoir écrit ou envoyé les messages WhatsApp contenus dans le dossier de la Justice de paix, quand bien même ceux-ci ont été produits par C._____ le 21 novembre 2017. Elle soutient qu'il pourrait s'agir soit de documents qui ont été modifiés, soit d'une manipulation effectuée sur l'application WhatsApp quant au nom de l'expéditeur des messages."

[6] Original text: "Son argumentation ne convainc ainsi pas, loin s'en faut. Dans ces circonstances, il n'y pas lieu d'ordonner une analyse des téléphones portables des parties. Par ailleurs, même dans l'hypothèse où une telle analyse était ordonnée et qu'elle démontrait que les messages n'émanaient pas de la recourante, cela ne constituerait pas encore un soupçon suffisant que l'intimé s'est rendu coupable d'une ou plusieurs infractions pénales [...]".

[7] Original text: "Il n'était pas l'auteur des messages que B_____ avait reçus [...]".

[8] Original text: "[...] Du 10 au 29 mars 2021, il a entretenu un échange virtuel via courriers électroniques puis par messages WhatsApp avec un jeune garçon [...] qui lui avait indiqué avoir 14 ans. Le recourant a invité l'enfant à se rendre le 29 mars 2021 dans un hôtel [...] en se faisant passer pour son neveu pour y entretenir des relations intimes [...]. En réalité, il conversait avec un policier."

[9] Original text: "Enfin, le fait que I_____ et l'appelant étaient initialement représentés par la même avocate, d'une part, et que leurs échanges Whatsapp ont disparu du téléphone de l'appelant, d'autre part, sont des indices de plus à charge."

In our model

We represent contextual information with auxiliary data in our model. Auxiliary data enables the judge to input different types of evidence with varying degree of importance. We assume the judge inputs all evidence through auxiliary data and then decides their weight in the specific case. The value v, which models the *in dubio pro reo* principle, captures the discretion that judges are afforded in evaluating each case.

In several cases, screenshots are accepted as evidence. This finding has several implications. Despite the prevalence of graphics editors, synthetic media such as deepfakes, and misinformation [9], people tend to believe that a simple screenshot can be accepted as evidence—as confirmed by the two survey studies detailed in Section 2. This implies once again a risk of unbalanced access to the deniability property: a malicious individual could use deniability to produce fake evidence against a victim and judges would likely consider this evidence as acceptable given our findings.

Finally, during our analysis we observed that judges believe that the law protects against forgeries, as false testimony is a criminal offense. This means any transcript, even a forgeable one like a screenshot, can be considered in court. This highlights the need for a discussion about deniability in various settings, as partially addressed by Yadav et al. [60]. We expand on this in the discussion below.

6 Discussion

In the previous sections we show how (cryptographic) deniability fails in practice from both a technical perspective—through the lens of our model defined in Section 3—as well as a legal perspective. This section elaborates on these findings on practical deniability and their consequences.

6.1 Either practical deniability or no deniability

Cryptographic deniability does not translate in the real world and appears to not be used in practice, which we have justified in the previous sections. The first question is therefore whether deniability is a necessary or desirable feature. Obtaining real deniability in practice implies non-trivial challenges that can impact users (e.g., spamming), performance (e.g., the cost of adding deniability in the protocol, attacks on sealed-sender-based authentication [51]), the service (e.g., respectability) or security (e.g., authenticity is somewhat lost if Bob can easily forge messages from Alice). Indeed, Apple has decided not to target deniability in PQ3, their new messaging protocol for iMessage [48].

Besides the technical challenges, it is important to consider whether deniability is a *beneficial* property. Practical deniability could facilitate the spread of "fake news" by enabling everyone to forge messages. Also, in the jurisprudence we reviewed, messages on victims' phones are used to convict criminals: practical deniability could invalidate this kind of evidence. Deniability would for example prevent a victim that receives abusive messages from proving the culpability of their author. However, as our legal analysis shows, conversations are usually one piece of evidence among others and do not represent the decisions' cornerstone. We also recall that the study of Yadav et al. [60] found that 60.2% of users surveyed desire only non-repudiation (i.e., they do not desire deniability), while only 12.7% desire deniability.

The very nature of deniability makes it hard to settle for either real-world deniability or no deniability at all. Differently from other security properties (e.g., encryption), users cannot control the deniability of their communication: deniability is effective only if *receivers* can plausibly forge messages. This however should not serve as an excuse to not consider vulnerable populations and to generalize survey results to different contexts [60, Section 8]. Unfortunately our field lacks research on practical defensive technologies that vulnerable groups use. One important exception is the work by Daffalla et al. [13] on the technological defense strategies that political activists used during the Sudanese revolution, which indicates that deniability is useful in this setting. Some participants used strategies to increase plausible deniability in case of arrest, such as adding decoy messages on their messaging application. Having complete deniability, such as ability to edit the user interface of Signal as shown in Fig. 3, would therefore help in cases where there is a need for a way to communicate freely without being held accountable, as the interviews of Daffalla et al. confirm.

If we opt for deniability, this should be *practical and accessible to everyone*. As we highlight in previous sections, impractical and unusable deniability is dangerous as only a few people can benefit from it. This was also pointed out by Jeff Burdges[10]: if deniability is not practical and widespread, the risk is that only rich and powerful people would have the resources to argue deniability in a court of law (or against the public opinion).

This discussion is reminiscent of the debate on end-to-end encryption, but the impact of deniability is less clear. Encryption solves, among other things, the problem of mass surveillance: an attacker has no access to messages in transit. Encryption is a mathematically-based property that has an immediate and global effect. By contrast, deniability is a more subtle, subjective property, which involves human and societal factors that can vary from case to case, e.g., which auxiliary information the judge has or which contextual information can help to frame the victim. Additionally, deniability holds if the *receiver* of a message satisfies some requirements: the author of a message cannot completely control which kind of deniability a message gets, whereas the sender, i.e., the author, has complete control over the message's encryption. Since deniability is not used in practice, it is difficult to argue about its impact and a community-wide discussion is necessary.

Finally, one might wonder if the strong deniability property that we seek is actually anonymity. By making it difficult for the judge to identify, profile and obtain data about the sender, or to link an account to them, we might avoid the deniability question in many cases. Achieving real-world plausible deniability in public messaging applications like WhatsApp and Signal, which link accounts to identities (e.g., phone numbers) seems very challenging. While allowing users to edit or add messages to a conversation would be a step towards this goal, it is unclear whether society would accept this as true deniability.

[10] https://mailarchive.ietf.org/arch/msg/mls/Kk6qai3kEza8B-L-_5R-qsEjtK0/

6.2 Deniability in front of whom

If real-world deniability is a goal for a given system, one must devise an appropriate threat model. The design must precisely define the judge in Definition 1. Who are we defending against? Which capacity and auxiliary information does the judge have? What about the incriminating adversary (Bob in our model)?

The leak of Hillary Clinton's email database [59] is often cited as an example where deniability would have had an impact. Wikileaks published *DKIM-protected* emails and some of the authors claimed that they were tampered with. However, the DKIM signatures mathematically proved that the leaks were legitimate [58]: the emails were *not* deniable. Moreover, the legitimacy of the emails is verifiable by the general public: it is sufficient to have the verification keys of the domain servers, publicly available by definition, to prove that no one tampered with the emails' content. Here the *general public* acts as the judge, rather than an official judge in a courtroom as usually envisioned. Our model captures this setting by reflecting real-life scenarios where Bob's data, such as keys and transcripts, are stolen published online, and Alice wants to deny sending the published messages to Bob. We model this by providing the judge with Bob's state st'_B after the interaction with Alice took place and the simulated state $\widetilde{\mathrm{st}}_B$ in the ideal world. This example highlights the importance of (1) properly defining the role of the judge and (2) applying our model to ensure that the judge's distinguishing probability is bounded by an acceptable probability.

As already discussed, if the threat model assumes a global adversary that *actively* colludes with Bob to frame a user, then deniability is impossible to achieve also because a lot of messaging applications register the users' identity (e.g., phone number). In this case, other properties and services such as anonymous communication over Tor should be used: the best way to avoid having to consider deniability in the first place is to guarantee anonymity.

6.3 How to make a deniable system

Privacy-focused messaging services like Signal (and others using the Signal protocol) use protocols that achieve a form of cryptographic deniability, like X3DH [36] and the recent PQXDH [20, 33]. These services market deniability as a feature, giving users the impression it is a practical aspect of the system. As we show in Section 4 the situation is not so simple: Signal in general does not achieve practical deniability, either technically or legally.

To achieve practical deniability, applications must be designed with deniability incorporated by design, similar to privacy. As we hinted before, the system must provide *practical* deniability against the kind of adversaries that the threat model considers. In particular, deniability must be plausible for all and accessible to everyone, regardless of the (technical) knowledge of the receiver, i.e., the potential accuser Bob in our model. The simulator in our model must also be practical: it should be feasible to "plausibly" run it to modify Bob's state (e.g., the phone).

One way to achieve this is to enable cracking as a controlled feature, allowing users to inject false messages authored by someone else into their conversation. This can be done by offering a feature that enables users to edit their conversation. This modification must be easily accessible in the application's user interface and must also alter messages that the application stores in the local database. The application must not record when and where the modifications take place. This solution mimics the real world: someone may report the content of a conversation, but the veracity of what is reported depends solely on the person reporting it, and other participant can challenge what is claimed. In practice, the modifications to Signal's user interface can mimic the tool used by Reitinger et al. [42] in their study (Fig. 3, Section A): incoming and outgoing messages can be edited and new messages can be inserted. If the application is offline, editing must not contradict the last access timestamp that the Signal server stores; in case of inconsistency, the application should warn the user. Reitinger et al.'s findings (Section 2.2) support this simple solution. In particular, the study suggests than an edit/create message feature significantly impacts people's opinion on someone's guilt, i.e., on plausibility of deniability claims.

Auxiliary data poses a significant threat to deniability. To enhance practical deniability, the system must minimize data retention, keeping only essential information and deleting surplus data once its purpose is fulfilled. This aligns privacy by design principles, which generally increase the plausibility of deniability.

7 Conclusion

This work discusses real-world deniability in messaging, highlighting how cryptographic deniability does not translate in the real world, both technically and legally. We analyze the technical side of deniability by applying our model on Signal and email with DKIM protection. In the legal setting, we examine court cases in Switzerland and find no successful claims of deniability in 140 cases where conversations were used as evidence.

We then propose a general discussion about deniability, considering whether it is a property messaging solutions should aim for. If the answer is affirmative, we discuss different settings for real world deniability—in particular who we want to defend against—and how we should design real-world deniable systems. We conclude with some remarks on the technical costs of real world deniability.

Messaging applications, and communication solutions in general, should either provide practical real-world deniability or stop claiming to do so. For messaging applications, particularly Signal, we propose a simple yet powerful solution: users must be able to modify sent and received messages stored on the device through the user interface. This mirrors the real world, where conversation can be truthfully or falsely reported, and others can challenge those reports. We believe Signal should adopt this approach to offer truly deniable communication, especially given its strong privacy focus.

With this work we aim to further the much-needed discussion on the practicality of deniability within the community and beyond. In this regard, some possible venues for future work are:

- Conduct more user studies on practical deniability. The studies discussed in Section 2.2 do not use real messaging applications, and it would be interesting to observe users' reactions when conversations can be modified in the interface.
- Investigate whether deniability is desirable in different contexts, especially for vulnerable groups such as political dissidents, whistleblowers, or harassment victims.
- Extend our model and analysis to *group messaging*, exploring, for example, cases where a judge corroborates with, or a party tries to frame, multiple group members.

Acknowledgments

We thank Elisabetta Colombo, Martino Colombo, Samy Mahjoubi and Olivier Peter for their assistance with the legal analysis. We appreciate the feedback from the PETS reviewers, which greatly improved this work. We thank Sylvain Chatel, Vero Estrada-Galiñanes, Bryan Ford, Abdullah Talayhan and the RWC 2023 reviewers for their comments on earlier versions of this work. This work was supported in part by the ETH4D Humanitarian Action Challenges project PAIDIT, the IC3-Ethereum Foundation and the Swiss National Science Foundation (project no. 192364). The authors used ChatGPT-4 to revise the English translation of court case decision extracts that appear in Section 5.

References

[1] Marcel Armour and Elizabeth Quaglia. 2022. Subverting Deniability. In *ProvSec*.
[2] Connor Bell and Saba Eskandarian. 2024. Anonymous Complaint Aggregation for Secure Messaging. *PoPETs* (2024).
[3] Mihir Bellare and Adriana Palacio. 2004. The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols. In *CRYPTO*.
[4] Nikita Borisov, Ian Goldberg, and Eric A. Brewer. 2004. Off-the-record communication, or, why not to use PGP. In *WPES*.
[5] Jacqueline Brendel, Rune Fiedler, Felix Günther, Christian Janson, and Douglas Stebila. 2022. Post-quantum Asynchronous Deniable Key Exchange and the Signal Handshake. In *PKC*.
[6] Ryan Castellucci. 2020. DKIM: Show Your Privates . https://rya.nc/dkim-privates.html. Last visited on 26-10-2023.
[7] Sofía Celi and Iraklis Symeonidis. 2020. The current state of denial. In *HotPETs*.
[8] Suvradip Chakraborty, Dennis Hofheinz, Ueli Maurer, and Guilherme Rito. 2023. Deniable Authentication When Signing Keys Leak. In *EUROCRYPT*.
[9] Justin D. Cochran and Stuart Napshin. 2021. Deepfakes: Awareness, Concerns, and Platform Accountability. *Cyberpsychology Behav. Soc. Netw.* (2021).
[10] Aloni Cohen and Kobbi Nissim. 2020. Towards formalizing the GDPR's notion of singling out. *Proc. Natl. Acad. Sci. USA* (2020).
[11] Cas Cremers and Michèle Feltz. 2011. One-round Strongly Secure Key Exchange with Perfect Forward Secrecy and Deniability. *IACR Cryptol. ePrint Arch.* (2011).
[12] D. Crocker, T. Hansen, and M. Kucherawy. 2011. *DomainKeys Identified Mail (DKIM) Signatures*. STD 76. RFC Editor. http://www.rfc-editor.org/rfc/rfc6376.txt.
[13] Alaa Daffalla, Lucy Simko, Tadayoshi Kohno, and Alexandru G. Bardas. 2021. Defensive Technology Use by Political Activists During the Sudanese Revolution. In *S&P*.
[14] Özgür Dagdelen, Marc Fischlin, Tommaso Gagliardoni, Giorgia Azzurra Marson, Arno Mittelbach, and Cristina Onete. 2013. A Cryptographic Analysis of OPACITY - (Extended Abstract). In *ESORICS*.
[15] Yevgeniy Dodis, Jonathan Katz, Adam D. Smith, and Shabsi Walfish. 2009. Composability and On-Line Deniability of Authentication. In *TCC*.
[16] Dropwizard. 2024. Dropwizard. https://github.com/dropwizard/dropwizard.
[17] Tribunal Cantonal du Canton de Fribourg. 2022. Jurisprudence du Tribunal Cantonal. https://publicationtc.fr.ch/. Last visited on 07-11-2022 (in French).
[18] Cynthia Dwork, Moni Naor, and Amit Sahai. 1998. Concurrent Zero-Knowledge. In *STOC*.
[19] Peter Elkind, Jack Gillum, and Craig Silverman. 2021. How Facebook Undermines Privacy Protections for Its 2 Billion WhatsApp Users. https://www.propublica.org/article/how-facebook-undermines-privacy-protections-for-its-2-billion-whatsapp-users. Last visited on 26-10-2023.
[20] Rune Fiedler and Christian Janson. 2024. A Deniability Analysis of Signal's Initial Handshake PQXDH. *PoPETs* (2024).
[21] Marc Fischlin and Sogol Mazaheri. 2015. Notions of deniable message authentication. In *WPES*.
[22] Jonathan Frankle, Sunoo Park, Daniel Shaar, Shafi Goldwasser, and Daniel J. Weitzner. 2018. Practical Accountability of Secret Processes. In *USENIX Security*.
[23] Sanjam Garg, Shafi Goldwasser, and Prashant Nalini Vasudevan. 2020. Formalizing Data Deletion in the Context of the Right to Be Forgotten. In *EUROCRYPT*.
[24] Matthew Green. 2014. Noodling about IM protocols. https://blog.cryptographyengineering.com/2014/07/26/noodling-about-im-protocols/. Last visited on 02-11-2022.
[25] Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. 2017. Message Franking via Committing Authenticated Encryption. In *CRYPTO*.
[26] Lachlan J. Gunn, Ricardo Vieitez Parra, and N. Asokan. 2019. Circumventing Cryptographic Deniability with Remote Attestation. *PoPETs* (2019).
[27] Lein Harn, Chia-Yin Lee, Changlu Lin, and Chin-Chen Chang. 2011. Fully Deniable Message Authentication Protocols Preserving Confidentiality. *Comput. J.* (2011).
[28] Keitaro Hashimoto, Shuichi Katsumata, Kris Kwiatkowski, and Thomas Prest. 2022. An efficient and generic construction for signal's handshake (X3DH): post-quantum, state leakage secure, and deniable. *Journal of Cryptology* (2022).
[29] Mike Hearn. 2014. Value of deniability (mailing list discussion). https://moderncrypto.org/mail-archive/messaging/2014/001173.html. Last visited on 09-11-2022.
[30] Andreas Hülsing and Florian Weber. 2021. Epochal Signatures for Deniable Group Chats. In *S&P*.
[31] Rawane Issa, Nicolas Alhaddad, and Mayank Varia. 2022. Hecate: Abuse Reporting in Secure Messengers with Sealed Sender. In *USENIX Security*.
[32] Ehren Kret. 2023. Quantum Resistance and the Signal Protocol. https://signal.org/blog/pqxdh/. Last visited on 26-10-2023.
[33] Ehren Kret and Rolfe Schmidt. 2023. The PQXDH Key Agreement Protocol. https://signal.org/docs/specifications/pqxdh/. Last visited on 23-11-2023.
[34] Joshua Lund. 2018. Technology preview: Sealed sender for Signal. https://signal.org/blog/sealed-sender/. Last visited on 03-11-2022.
[35] Moxie Marlinspike. 2013. Simplifying OTR deniability. https://signal.org/blog/simplifying-otr-deniability/. Last visited on 25-10-2022.
[36] Moxie Marlinspike and Trevor Perrin. 2016. The X3DH Key Agreement Protocol. https://www.signal.org/docs/specifications/x3dh/. Last visited on 25-10-2022.
[37] Ian Martiny, Gabriel Kaptchuk, Adam J Aviv, Daniel S Roche, and Eric Wustrow. 2021. Improving Signal's Sealed Sender.. In *NDSS*.
[38] Trevor Perrin and Moxie Marlinspike. 2016. The Double Ratchet Algorithm. https://signal.org/docs/specifications/doubleratchet/. Last visited on 25-10-2022.
[39] Chambre pénale d'appel et de révision de la République et canton de Genève. 2018. Arrêt AARP/54/2018. https://justice.ge.ch/apps/decis/fr/parp/show/1593458. Last visited on 07-11-2022 (in French).
[40] Mario Di Raimondo and Rosario Gennaro. 2005. New approaches for deniable authentication. In *CCS*.
[41] Mario Di Raimondo, Rosario Gennaro, and Hugo Krawczyk. 2006. Deniable authentication and key exchange. In *CCS*.
[42] N. Reitinger, N. Malkin, O. Akgul, M. L. Mazurek, and I. Miers. 2023. Is Cryptographic Deniability Sufficient? Non-Expert Perceptions of Deniability in Secure Messaging. In *S&P*.
[43] Sarah Scheffler and Mayank Varia. 2021. Protecting Cryptography Against Compelled Self-Incrimination. In *USENIX Security*.
[44] Signal. 2021. Grand jury subpoena for Signal user data, Central District of California. https://signal.org/bigbrother/central-california-grand-jury/. Last visited on 25-10-2022.
[45] Signal. 2021. Grand jury subpoena for Signal user data, Central District of California (again!). https://signal.org/bigbrother/cd-california-grand-jury/. Last visited on 25-10-2022.
[46] Signal. 2022. Signal-Server. https://github.com/signalapp/Signal-Server.
[47] Michael A. Specter, Sunoo Park, and Matthew Green. 2021. KeyForge: Non-Attributable Email from Forward-Forgeable Signatures. In *USENIX Security*.
[48] Douglas Stebila. 2024. Security analysis of the iMessage PQ3 protocol. Cryptology ePrint Archive, Paper 2024/357. https://eprint.iacr.org/2024/357 https://eprint.iacr.org/2024/357.
[49] Alan M. Turing. 1950. Computing machinery and intelligence. *Mind* LIX, 236 (1950), 433–460.
[50] Nirvan Tyagi, Paul Grubbs, Julia Len, Ian Miers, and Thomas Ristenpart. 2019. Asymmetric Message Franking: Content Moderation for Metadata-Private End-to-End Encryption. In *CRYPTO*.
[51] Nirvan Tyagi, Julia Len, Ian Miers, and Thomas Ristenpart. 2022. Orca: Blocklisting in Sender-Anonymous Messaging. In *USENIX Security*.
[52] Nik Unger and Ian Goldberg. 2015. Deniable key exchanges for secure messaging. In *CCS*.
[53] Nik Unger and Ian Goldberg. 2018. Improved Strongly Deniable Authenticated Key Exchanges for Secure Messaging. In *PoPETs*.
[54] Nihal Vatandas, Rosario Gennaro, Bertrand Ithurburn, and Hugo Krawczyk. 2020. On the Cryptographic Deniability of the Signal Protocol. In *ACNS*.
[55] Faxing Wang, Shaanan Cohney, Riad Wahby, and Joseph Bonneau. 2023. NOTRY: deniable messaging with retroactive avowal. *IACR Cryptol. ePrint Arch.* (2023).
[56] WhatsApp. 2024. How to check read receipts. https://faq.whatsapp.com/665923838265756/?cms_platform=android. Last visited on 13-6-2024.
[57] WhatsApp. 2024. Information for Law Enforcement Authorities. https://faq.whatsapp.com/444002211197967. Last visited on 23-2-2024.
[58] WikiLeaks. 2016. DKIM Verification. https://wikileaks.org/DKIM-Verification.html. Last visited on 07-11-2022.
[59] WikiLeaks. 2016. Hillary Clinton Email Archive. https://wikileaks.org//clinton-emails/. Last visited on 07-11-2022.
[60] Tarun Kumar Yadav, Devashish Gosain, and Kent E. Seamons. 2023. Cryptographic Deniability: A Multi-perspective Study of User Perceptions and Expectations. In *USENIX Security*.
[61] Fan Zhang, Deepak Maram, Harjasleen Malvai, Steven Goldfeder, and Ari Juels. 2020. DECO: Liberating Web Data Using Decentralized Oracles for TLS. In *CCS*.

A Additional information on meta-deniability

Fig. 3 shows the interactive forging tool that Reitinger et al. use for their study [42].

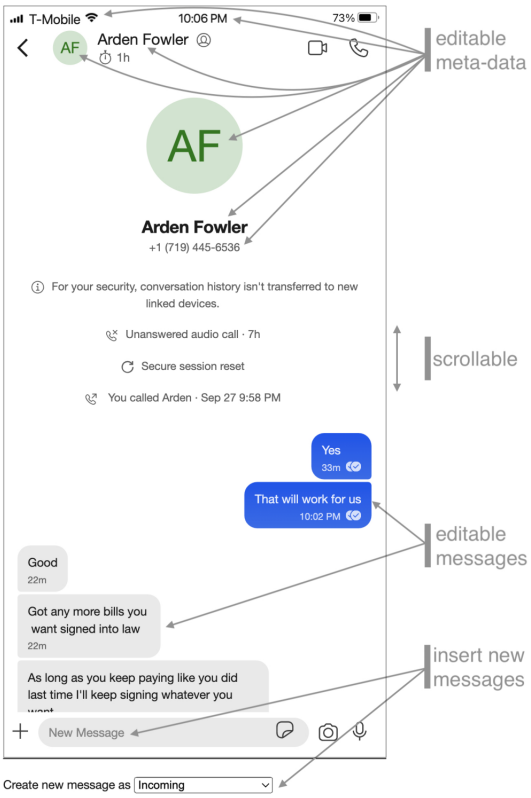

Figure 3: Interactive forging tool that Reitinger et al. use for their study of deniability in secure messaging [42].

B Additional information on legal analysis

In this section we provide additional information on the legal analysis that we discuss in Section 5.

B.1 Elasticsearch query for legal analysis

Fig. 4 shows the Elasticsearch query that we use to select the initial set of cases from the website entscheidsuche.ch.

```
"query": { "bool": {
  "must": [
    { "match": {
        "attachment.content": "WhatsApp"
      }
    }
  ],
  "should": [
    { "match_phrase": {
        "attachment.content": "droit pénal"
      }
    },
    { "match_phrase": {
        "attachment.content": "chambre pénale"
      }
    },
    { "match_phrase": {
        "attachment.content": "diritto penale"
      }
    }
  ],
  "minimum_should_match": 1
}}
```

Figure 4: Elasticsearch query for the legal analysis.

B.2 List of cases

Below we give the list of cases that we consider for our analysis, after pruning the cases pertaining to federal courts.

(1) https://entscheidsuche.ch/docs/TI_Gerichte/TI_CARP_001_17-2017-175_2017-10-26.html
(2) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-1014-2020_2022-01-19.html
(3) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2023-102_2023-09-18.pdf
(4) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2022-110_2022-11-23.pdf
(5) https://entscheidsuche.ch/docs/GR_Gerichte/GR_KG_005_SK2-2023-41_2023-08-24.pdf
(6) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-15331-2022_2023-08-14.html
(7) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2020-23_2020-06-02.pdf
(8) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2020-37_2020-05-06.pdf
(9) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2022-24_2022-09-22.pdf
(10) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-25515-2023_2023-12-20.html
(11) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2019-225_2019-08-14.pdf
(12) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2020-76_2020-08-11.pdf
(13) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2021-237_2021-11-16.pdf
(14) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2023---441_nodate.html
(15) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-15520-2022_2022-11-30.html
(16) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-16796-2017_2018-05-08.html
(17) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2022-84_2022-06-10.pdf
(18) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-18317-2020_2022-09-09.html
(19) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2020-18_2020-03-04.pdf

(20) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2019-31_2019-02-26.pdf
(21) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-19781-2016_2017-12-08.html
(22) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2020-107_2020-09-16.pdf
(23) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2019-288_2019-11-18.pdf
(24) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2022-63_2022-03-28.pdf
(25) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-21424-2022_2022-12-13.html
(26) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-2975-2022_2022-06-24.html
(27) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-19131-2020_2023-05-30.html
(28) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-7180-2020_2022-10-06.html
(29) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-20041-2019_2021-11-24.html
(30) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2023-84_2023-06-13.pdf
(31) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2019-124_2019-05-21.pdf
(32) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2019-304_2019-11-19.pdf
(33) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-8836-2018_2021-11-30.html
(34) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-23526-2018_2022-09-01.html
(35) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-2912-2020_2021-07-13.html
(36) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-21299-2022_2023-05-04.html
(37) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-2401-2023_2023-11-13.html
(38) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-15754-2015_2017-05-30.html
(39) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_013_D-cision---2019---22_nodate.html
(40) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-11428-2019_2022-07-01.html
(41) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2017-180_2017-06-27.pdf
(42) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-1200-2020_2023-01-27.html
(43) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-1631-2023_2023-11-24.html
(44) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2022-205_2022-12-09.pdf
(45) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-8307-2023_2023-10-04.html
(46) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-15766-2018_2019-11-12.html
(47) https://entscheidsuche.ch/docs/TI_Gerichte/TI_TRAP_002_60-2022-41_2022-10-18.html
(48) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-18308-2021_2021-11-26.html
(49) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-2719-2020_2021-02-02.html
(50) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-25515-2023_2023-12-22.html
(51) https://entscheidsuche.ch/docs/JU_Gerichte/JU_TC_004_CPR-2022-140_2023-01-16.pdf
(52) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2019-182_2019-06-28.pdf
(53) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2019-231_2019-08-29.pdf
(54) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2019-14_2019-02-19.pdf
(55) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-3463-2018_2019-05-21.html
(56) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-990-2022_2023-05-02.html
(57) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2021-159_2021-08-18.pdf
(58) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-19228-2021_2022-11-23.html
(59) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-1438-2021_2022-02-25.html
(60) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-8223-2021_2022-09-21.html
(61) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-6315-2022_2023-12-05.html
(62) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-3544-2022_2022-11-21.html
(63) https://entscheidsuche.ch/docs/NE_Omni/NE_TC_009_CPEN-2017-98_2018-07-31.html
(64) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-4257-2019_2020-08-27.html
(65) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-25665-2022_2023-04-26.html
(66) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-20958-2021_2023-03-21.html
(67) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-2603-2022_2023-04-26.html
(68) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2022-212_2023-05-17.pdf
(69) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-1796-2022_2023-05-22.html
(70) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-17161-2016_2017-11-07.html
(71) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_013_D-cision---2020---97_nodate.html
(72) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-257-2020_2022-11-07.html
(73) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-5629-2014_2017-02-07.html
(74) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-10477-2020_2022-09-23.html
(75) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2016---387_nodate.html

(76) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-1023-2020_2020-09-08.html
(77) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-18649-2022_2023-04-24.html
(78) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-550-2020_2020-04-30.html
(79) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2019-282_2020-02-21.pdf
(80) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-15649-2017_2018-05-29.html
(81) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-2779-2015_2018-02-22.html
(82) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TAPI_001_A-2961-2022_2023-03-16.html
(83) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2019-75_2019-06-24.pdf
(84) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-4085-2019_2021-05-25.html
(85) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-23938-2016_2018-03-19.html
(86) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-26859-2022_2023-09-25.html
(87) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2017-156_2017-06-12.pdf
(88) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-888-2021_2021-06-01.html
(89) https://entscheidsuche.ch/docs/CH_BGE/CH_BGE_006_BGE-149-IV-57_2022-10-31.html
(90) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-4535-2023_2023-10-23.html
(91) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-13713-2023_2023-12-12.html
(92) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-23766-2023_2024-01-17.html
(93) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2017---120_nodate.html
(94) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-7681-2017_2022-09-19.html
(95) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2016-302_2016-12-12.pdf
(96) https://entscheidsuche.ch/docs/CH_BVGer/CH_BVGE_001_A-5738-2022_2023-11-10.pdf
(97) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-6109-2023_2023-08-25.html
(98) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-1214-2017_2020-03-10.html
(99) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-457-2019_2023-02-28.html
(100) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-12446-2022_2023-06-09.html
(101) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-23884-2020_2021-10-19.html
(102) https://entscheidsuche.ch/docs/NE_Omni/NE_TC_008_ARMP-2021-33_2021-03-30.html
(103) https://entscheidsuche.ch/docs/BE_ZivilStraf/BE_OG_008_BK-2021-371_2021-09-28.pdf
(104) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-21428-2019_2021-11-18.html
(105) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-2940-2022_2023-04-26.html
(106) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-7411-2022_2023-03-27.html
(107) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-24107-2022_2023-12-01.html
(108) https://entscheidsuche.ch/docs/CH_BVGer/CH_BVGE_001_F-2572-2019_2020-08-17.pdf
(109) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-3388-2020_2022-06-20.html
(110) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-3130-2020_2021-09-23.html
(111) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-4741-2021_2023-08-29.html
(112) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2016-218_2016-10-31.pdf
(113) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2021-198_2021-09-30.pdf
(114) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---324_nodate.html
(115) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-1465-2021_2021-07-06.html
(116) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2019---76_nodate.html
(117) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2020---147_nodate.html
(118) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-22161-2019_2023-02-27.html
(119) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2023---365_nodate.html
(120) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-6838-2023_2023-12-05.html
(121) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-1097-2022_2023-07-06.html
(122) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-18054-2021_2022-02-15.html
(123) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-24493-2020_2021-12-10.html
(124) https://entscheidsuche.ch/docs/JU_Gerichte/JU_TC_004_CPR-2020-60_2020-11-23.pdf
(125) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2020-35_2020-04-14.pdf
(126) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-21128-2014_2016-05-13.html
(127) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-11449-2016_2017-01-24.html
(128) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-4768-2020_2021-11-08.html
(129) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-8000-2018_2021-06-25.html
(130) https://entscheidsuche.ch/docs/NE_Omni/NE_TC_009_CPEN-2017-50_2017-12-13.html
(131) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-2700-2021_2022-08-03.html

(132) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-7305-2021_2023-12-22.html
(133) https://entscheidsuche.ch/docs/JU_Gerichte/JU_TC_004_CPR-2022-139_2023-01-16.pdf
(134) https://entscheidsuche.ch/docs/NE_Omni/NE_TC_009_CPEN-2019-33_2020-09-24.html
(135) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-5652-2018_2021-12-16.html
(136) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-6639-2020_2022-04-29.html
(137) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-15481-2018_2022-07-11.html
(138) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2020---381_nodate.html
(139) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-17122-2023_2023-12-04.html
(140) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-23939-2016_2017-06-12.html
(141) https://entscheidsuche.ch/docs/JU_Gerichte/JU_TC_004_CPR-2021-62_2021-10-07.pdf
(142) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-18659-2020_2022-09-26.html
(143) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-4153-2021_2022-11-01.html
(144) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2016---341_nodate.html
(145) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_299----------_nodate.html
(146) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-16154-2013_2014-03-20.html
(147) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2022-8_2022-01-24.pdf
(148) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-5601-2023_2023-12-18.html
(149) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-13448-2022_2023-09-28.html
(150) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-13389-2017_2024-01-09.html
(151) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-3130-2020_2022-04-21.html
(152) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-3356-2022_2023-02-27.html
(153) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-17522-2020_2023-07-25.html
(154) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-11884-2018_2023-02-07.html
(155) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-14939-2017_2018-05-03.html
(156) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-20079-2021_2023-06-26.html
(157) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---167_nodate.html
(158) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-15668-2017_2018-10-04.html
(159) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-11473-2015_2016-03-17.html
(160) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-16588-2020_2022-07-23.html
(161) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-24857-2020_2021-09-02.html
(162) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-1958-2020_2023-01-12.html
(163) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-12483-2020_2022-08-16.html
(164) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-7371-2022_2022-09-14.html
(165) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_003_C-10439-2020_2022-03-21.html
(166) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-16192-2015_2017-02-10.html
(167) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_139----------_nodate.html
(168) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2020-118_2020-10-08.pdf
(169) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_417----------_nodate.html
(170) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---9_nodate.html
(171) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---406_nodate.html
(172) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2020---340_nodate.html
(173) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-13640-2020_2023-08-30.html
(174) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_013_D-cision---2023---39_nodate.html
(175) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-14994-2021_2023-10-19.html
(176) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_006_501-2017-222_2018-05-23.pdf
(177) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-15649-2017_2017-12-11.html
(178) https://entscheidsuche.ch/docs/CH_BVGer/CH_BVGE_001_F-1516-2018_2020-02-18.pdf
(179) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2016---121_nodate.html
(180) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-8413-2020_2022-10-06.html
(181) https://entscheidsuche.ch/docs/TI_Gerichte/TI_CARP_001_17-2020-361_2021-05-20.html
(182) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---42_nodate.html
(183) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_003_C-26550-2019_2022-04-04.html
(184) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-14819-2022_2023-08-18.html
(185) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-11563-2017_2018-04-27.html
(186) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_006_501-2021-199_2022-08-23.pdf
(187) https://entscheidsuche.ch/docs/JU_Gerichte/JU_TC_004_CPR-2022-63_2022-06-17.pdf

(188) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_013_D-cision---2022---66_nodate.html
(189) https://entscheidsuche.ch/docs/VS_Gerichte/VS_BZG_999_P1-22-20_2022-08-29.pdf
(190) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2018---354_nodate.html
(191) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CAPJ_001_CAPJ-4-2017_2018-12-28.html
(192) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-18569-2015_2017-09-15.html
(193) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2017---309_nodate.html
(194) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-19602-2020_2023-11-02.html
(195) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-11574-2020_2023-06-01.html
(196) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-15892-2017_2018-10-02.html
(197) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-23939-2016_2017-11-30.html
(198) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-15739-2017_2017-12-01.html
(199) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-3008-2023_2023-11-09.html
(200) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-12018-2019_2023-03-13.html
(201) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-12945-2020_2022-04-12.html
(202) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-23680-2016_2023-04-04.html
(203) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-17197-2021_2022-09-13.html
(204) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2023---420_nodate.html
(205) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-24952-2017_2022-12-21.html
(206) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-8833-2022_2022-07-11.html
(207) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2020---262_nodate.html
(208) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2018---227_nodate.html
(209) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2022---27_nodate.html
(210) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2023---245_nodate.html
(211) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_005_502-2020-25_2020-03-02.pdf
(212) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2022---157_nodate.html
(213) https://entscheidsuche.ch/docs/NE_Omni/NE_TC_008_ARMP-2023-111_2023-10-23.html
(214) https://entscheidsuche.ch/docs/VS_Gerichte/VS_BZG_999_P1-20-107_2023-03-01.pdf
(215) https://entscheidsuche.ch/docs/NE_Omni/NE_TC_009_CPEN-2021-14_2021-10-19.html
(216) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-414-2020_2022-03-22.html
(217) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-21761-2020_2023-06-19.html
(218) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2022---449_nodate.html
(219) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2020---359_nodate.html
(220) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2017---245_nodate.html
(221) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-1286-2020_2022-08-11.html
(222) https://entscheidsuche.ch/docs/NE_Omni/NE_TC_008_ARMP-2023-45_2023-04-24.html
(223) https://entscheidsuche.ch/docs/VS_Gerichte/VS_BZG_999_P1-19-38_2021-08-25.pdf
(224) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-4246-2018_2023-04-13.html
(225) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-13988-2020_2021-10-21.html
(226) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-13524-2017_2024-01-23.html
(227) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-1668-2022_2022-09-06.html
(228) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-7877-2020_2022-11-08.html
(229) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2019---456_nodate.html
(230) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2018---155_nodate.html
(231) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2019---263_nodate.html
(232) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-144-2016_2018-10-19.html
(233) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-8063-2020_2021-11-16.html
(234) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-1383-2021_2023-05-08.html
(235) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-10401-2016_2018-07-10.html
(236) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-2358-2018_2021-09-27.html
(237) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-19096-2019_2022-10-26.html
(238) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-8266-2012_2014-10-14.html
(239) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---204_nodate.html
(240) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2020---270_nodate.html
(241) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2020---481_nodate.html
(242) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2020---491_nodate.html
(243) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2020---153_nodate.html

(244) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-8063-2020_2022-10-17.html
(245) https://entscheidsuche.ch/docs/NE_Omni/NE_TC_007_CMPEA-2019-65_2020-06-29.html
(246) https://entscheidsuche.ch/docs/NE_Omni/NE_TC_009_CPEN-2019-8_2019-12-19.html
(247) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-10479-2019_2022-11-04.html
(248) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2022---450_nodate.html
(249) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2020---386_nodate.html
(250) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2018---391_nodate.html
(251) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-17032-2018_2021-09-06.html
(252) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_229----------_nodate.html
(253) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-4603-2020_2022-05-18.html
(254) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-14428-2021_2023-02-27.html
(255) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-17359-2013_2017-07-06.html
(256) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-25787-2019_2023-01-14.html
(257) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---384_nodate.html
(258) https://entscheidsuche.ch/docs/BE_ZivilStraf/BE_OG_005_SK-2022-41_2022-10-05.pdf
(259) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2023---123_nodate.html
(260) https://entscheidsuche.ch/docs/JU_Gerichte/JU_TC_004_CPR-2021-60_2021-10-26.pdf
(261) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-16064-2015_2017-07-19.html
(262) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-16994-2020_2021-12-17.html
(263) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-23680-2016_2022-06-15.html
(264) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-8006-2019_2022-12-15.html
(265) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_001_C-8172-2020_2023-11-27.html
(266) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-24672-2020_2022-09-14.html
(267) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2020---474_nodate.html
(268) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_006_501-2019-99_2020-08-31.pdf
(269) https://entscheidsuche.ch/docs/NE_Omni/NE_TC_009_CPEN-2021-71_2022-08-16.html
(270) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_128----------_nodate.html
(271) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-15456-2019_2022-10-07.html
(272) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-14753-2020_2024-01-18.html
(273) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-23847-2016_2019-09-27.html
(274) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2023---108_nodate.html
(275) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2022---135_nodate.html
(276) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_006_501-2016-202_2018-01-30.pdf
(277) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-19131-2020_2022-09-12.html
(278) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-1209-2019_2020-03-03.html
(279) https://entscheidsuche.ch/docs/BE_ZivilStraf/BE_OG_005_SK-2023-32_2023-10-04.pdf
(280) https://entscheidsuche.ch/docs/BE_ZivilStraf/BE_OG_005_SK-2021-525_2022-06-09.pdf
(281) https://entscheidsuche.ch/docs/NE_Omni/NE_TC_009_CPEN-2021-75_2022-03-22.html
(282) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2019---399_nodate.html
(283) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2019---436_nodate.html
(284) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_014_A-1333-2022_2022-10-03.html
(285) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-9559-2019_2021-07-02.html
(286) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_006_C-3979-2020_2023-03-13.html
(287) https://entscheidsuche.ch/docs/NE_Omni/NE_TC_009_CPEN-2019-88_2020-05-06.html
(288) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2023---118_nodate.html
(289) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_254----------_nodate.html
(290) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---141_nodate.html
(291) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2022---195_nodate.html
(292) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_014_A-659-2019_2021-11-24.html
(293) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-14376-2019_2022-06-16.html
(294) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---131_nodate.html
(295) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2017---362_nodate.html
(296) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_006_501-2021-116_2022-05-18.pdf
(297) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-791-2017_2018-11-06.html
(298) https://entscheidsuche.ch/docs/JU_Gerichte/JU_TC_004_CPR-2023-7_2023-01-26.pdf
(299) https://entscheidsuche.ch/docs/JU_Gerichte/JU_TPI_006_TPI-2021-13_2021-06-29.pdf

(300) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2023---384_nodate.html
(301) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_165----------_nodate.html
(302) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---343_nodate.html
(303) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2018---386_nodate.html
(304) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-5702-2019_2022-05-30.html
(305) https://entscheidsuche.ch/docs/BE_ZivilStraf/BE_OG_005_SK-2021-112_2022-02-16.pdf
(306) https://entscheidsuche.ch/docs/JU_Gerichte/JU_TC_004_CPR-2021-40_2021-11-16.pdf
(307) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-12795-2019_2022-01-11.html
(308) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-12351-2017_2019-04-29.html
(309) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_430----------_nodate.html
(310) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---245_nodate.html
(311) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---302_nodate.html
(312) https://entscheidsuche.ch/docs/FR_Gerichte/FR_TC_006_501-2021-118_2022-05-17.pdf
(313) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_011_P-19938-2019_2021-12-10.html
(314) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-794-2021_2021-07-06.html
(315) https://entscheidsuche.ch/docs/BE_ZivilStraf/BE_OG_005_SK-2017-126_2017-12-06.pdf
(316) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-1852-2021_2022-05-24.html
(317) https://entscheidsuche.ch/docs/JU_Gerichte/JU_TC_001_TPI-2020-167_2021-06-22.pdf
(318) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-6853-2017_2022-04-08.html
(319) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2019---346_nodate.html
(320) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---309_nodate.html
(321) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2020---103_nodate.html
(322) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2017---389_nodate.html
(323) https://entscheidsuche.ch/docs/BE_ZivilStraf/BE_OG_005_SK-2022-390_2023-03-23.pdf
(324) https://entscheidsuche.ch/docs/VS_Gerichte/VS_TC_001_TCVS-P1-18-90_2021-06-30.pdf
(325) https://entscheidsuche.ch/docs/VS_Gerichte/VS_BZG_999_P1-18-90_2021-06-30.pdf
(326) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---390_nodate.html
(327) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-17047-2018_2023-01-23.html
(328) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-4214-2015_2017-06-22.html
(329) https://entscheidsuche.ch/docs/BE_ZivilStraf/BE_OG_005_SK-2022-640_2023-12-06.pdf
(330) https://entscheidsuche.ch/docs/TI_Gerichte/TI_CARP_001_17-2016-246_2017-04-21.html
(331) https://entscheidsuche.ch/docs/TI_Gerichte/TI_TCAS_001_38-2023-24_2023-06-19.html
(332) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_013_A-3054-2021_2022-09-13.html
(333) https://entscheidsuche.ch/docs/NE_Omni/NE_TC_009_CPEN-2020-31_2021-02-23.html
(334) https://entscheidsuche.ch/docs/GE_Gerichte/GE_TP_001_P-11681-2017_2021-11-05.html
(335) https://entscheidsuche.ch/docs/VD_FindInfo/VD_TC_003_Jug---2021---32_nodate.html
(336) https://entscheidsuche.ch/docs/TI_Gerichte/TI_TCAS_001_36-2016-140_2017-05-23.html
(337) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-11148-2020_2023-08-18.html
(338) https://entscheidsuche.ch/docs/GE_Gerichte/GE_CJ_009_P-6939-2019_2022-05-24.html
(339) https://entscheidsuche.ch/docs/NE_Omni/NE_TC_009_CPEN-2021-68_2022-04-27.html
(340) https://entscheidsuche.ch/docs/BE_ZivilStraf/BE_OG_005_SK-2019-175_2020-09-17.pdf
(341) https://entscheidsuche.ch/docs/BE_ZivilStraf/BE_OG_005_SK-2018-90_2019-04-04.pdf