Anonymous Whistleblowing over Authenticated Channels

Thomas Agrikola[2,⋆], Geoffroy Couteau[1,⋆⋆], and Sven Maier[2,⋆]

[1] CNRS, IRIF, Université de Paris, France geoffroy.couteau@irif.fr
[2] Karlsruhe Institute of Technology, Karlsruhe, Germany
{thomas.agrikola,sven.maier}@kit.edu

Abstract. The goal of *anonymous whistleblowing* is to publicly disclose a message while at the same time hiding the identity of the sender in a way that even if suspected of being the sender, this cannot be proven. While many solutions to this problem have been proposed over the years, they all require some form of interaction with trusted or non-colluding parties. In this work, we ask whether this is fundamentally inherent. We put forth the notion of *anonymous transfer* as a primitive allowing to solve this problem *without* relying on any participating trusted parties. We initiate the theoretical study of this question, and derive negative and positive results on the existence of such a protocol. We refute the feasibility of *asymptotically* secure anonymous transfer, where the message will be received with overwhelming probability while at the same time the identity of the sender remains hidden with overwhelming probability. On the other hand, resorting to *fine-grained* cryptography, we provide a heuristic instantiation (assuming ideal obfuscation) which guarantees that the message will be correctly received with overwhelming probability and the identity of the sender leaks with vanishing probability. Our results provide strong foundations for the study of the possibility of anonymous communications through authenticated channels, an intriguing goal which we believe to be of fundamental interest.

1 Introduction

The term whistleblowing denotes "the disclosure by a person, usually an employee in a government agency or private enterprise, to the public or to those in authority, of mismanagement, corruption, illegality, or some other wrongdoing" [Whi]. Consider the following scenario. You are happily employed by some government agency. However, one day, you learn that your employer violates human rights. You strongly disagree with this breach of trust and law but you are bound by law to keep internal information secret. Consequently, you are faced with a dilemma: either you ignore the human rights violation, or you face dishonorable discharge or even jail. In fact, whistleblowers often take an immense personal

⋆ Supported by funding from the topic Engineering Secure Systems of the Helmholtz Association (HGF) and by KASTEL Security Research Labs.
⋆⋆ Supported by ANR SCENE.

risk, and face sentences ranging from exile [BEA14] to incarceration [Phi18] or worse. Whistleblowing is crucial for democracy to educate the public of misdeeds and to call those in power to account. Therefore, it is desirable to cryptographically protect the identity of the whistleblower to allow a low-risk disclosure of wrongdoing.

The importance of this question is well recognized in cryptography and security. It has been the subject of several influential works (e.g. DC-nets [Cha88], Riposte [CBM15] or Blinder [APY20]). Concrete solutions include the use of secure messaging apps [CGCD+20; Ber16], mix-nets [Cha03], onion routing systems such as the Tor network [DMS04], or solutions built on top of DC-nets and secure computation techniques [CBM15; APY20] (see also [ECZ+21; NSSD21]).

Yet, all current approaches to anonymous whistleblowing rely on trusted parties (or non-colluding partially trusted servers), which either receive privately the communication, or implement a distributed protocol to emulate an anonymous network. Therefore, however ingenious and scalable some of these solutions are, whistleblowers must ultimately trust that they will interact with parties or servers which will (at least for some of them) remain honest and refuse to collude throughout the transmission.

In this work, we ask whether this is fundamentally inherent, or whether anonymous whistleblowing is possible in theory without having to privately communicate with trusted parties. In its most basic form, the question we ask is the following:

Is it possible for a whistleblower (who is communicating solely through authenticated point-to-point or broadcast channels) to publicly reveal some message m while remaining anonymous without assuming trusted participating parties?

We do allow a Common Reference String (CRS) for technical reasons, and stress that while it is technically also a trust assumption, it is much weaker; instead of trusting a set of parties *every time* to follow the exact protocol and to not cheat in any way, we *only* require a CRS to be set up *once*: A CRS that was successfully sampled just once can be used for all future interactions.

The above is, of course, trivially impossible if the whistleblower is the only communicating party. However, it becomes meaningful in a multiparty setting, where a number of parties (unaware of the intent of the whistleblower) exchange innocent-looking messages (think of a group of people having a conversation, or using some public messaging service like Twitter or Facebook to broadcast information). In this context, the question translates as follows: could the whistleblower somehow disguise its communication as an innocent-looking conversation with the other parties, such that the message m can be publicly extracted (by anyone) from the *entire conversation*, yet the identity of which party was indeed the whistleblower remains hidden? To our knowledge, this intriguing question has never been studied in the past. Our main contributions are threefold:

1. **A definitional framework.** We put forth a formal definition for a cryptographic primitive that realizes the above goal, which we call an *Anonymous Transfer*. We study the relation between variants of the notion.

2. **Impossibility results.** We prove a strong impossibility result: we show that Anonymous Transfer with overwhelming correctness and anonymity cannot be realized in any polynomial number of rounds, by exhibiting a general attack against any such protocol. This non-trivial result demonstrates that anonymously communicating over authenticated channels is impossible with standard cryptographic security levels, even assuming strong cryptographic primitives such as ideal obfuscation.
3. **Feasibility result.** We complement our impossibility result by an intriguing *feasibility* result: we show that *fine-grained* Anonymous Transfer is possible assuming ideal obfuscation. The term fine-grained refers to cryptographic constructions which are only guaranteed secure against adversaries whose computational power is a fixed polynomial in the computing power of the honest parties (in our case, the gap is quadratic). Our instantiation is a plausible heuristic candidate when instantiating the ideal obfuscation by candidate indistinguishability obfuscation schemes.

Both our negative and positive results are highly non-trivial and require a very careful analysis. We view our work as addressing a fundamental question regarding the a priori possibility of secure whistleblowing without interacting with trusted parties, through the lens of anonymous communications over authenticated channels. Nevertheless, our study is of a purely theoretical nature, and does not have immediate practical relevance. In particular, we do not compare our results to the practical real-world methods which whistleblowers can employ.

Anonymous Transfer and plausible deniability. The fundamental goal of an Anonymous Transfer protocol is to achieve plausible deniability: the whistleblower should be able to hide its identity among a group of parties, such that even if it is strongly *suspected* that he is the whistleblower, this cannot be *proven* – any party could equally be the whistleblower. Importantly, the involved parties are never required to be aware that a message is being transmitted: their consent or collaboration is not needed for the Anonymous Transfer to take place, and they themselves have no advantage in finding out who the whistleblower was.

1.1 Undetectable Secure Computation

Secure Multiparty Computation (MPC) allows a set of parties to jointly evaluate a function on their inputs without revealing these inputs. In certain scenarios, however, the standard guarantees of MPC become insufficient: the mere fact that a party is participating to a certain protocol already reveals information about that party. Consider for example the following scenario: your company was hacked, but you do not have enough forensic data to trace the attackers. If several companies fell victim to the same hacker, a joint effort may yield enough information to successfully trace the hacker. However, the very fact that you are *initiating* such a protocol reveals that your company has been hacked.

The notion of Covert Multiparty Computation (CMPC) [vHL05; CGO+07] was introduced to cope with situations in which even revealing one's participation

to the protocol is undesirable. CMPC allows a set of parties to securely compute a protocol among n parties with the following two guarantees: (1) If all parties are actually willing to participate in the protocol (and are not simply having innocent conversations), and if the output of the protocol was *acceptable* (which is specified by some function g of the joint input), then everyone learns the result of the protocol. (2) Otherwise (if at least *one* party was not participating, or the output was not acceptable), no one learns anything about who were the participating parties (or even whether there was any).

CMPC is a powerful strengthening of secure computation. However, it still has two important downsides: a single non-participating party is sufficient to make the entire protocol fail (no one gets any output), and when all parties participate, they all learn that they participated (hence, no one can deny anymore having participated in the protocol). One of the primary motivations behind the study of Anonymous Transfer, which we put forth in this work, is to open the avenue to the study of a significantly more powerful form of secure computation that provides the strongest deniability guarantees one can hope for: a secure computation protocol where, even after the successful protocol execution, no one learns *who the participants were*. Specifically, we consider the following setting: N individuals are interacting. Among them, k players are willing to jointly compute a public function f on their private inputs $(x_1, \ldots, x_k)$, while the remaining $(N - k)$ are not interested in taking part to the protocol (nor are even aware of the fact that a secure computation might be taking place). At the end of the protocol, the k participants should all receive the output, but no party should be able to find out which of the parties were actually participating. We call this strengthening of secure computation *undetectable secure computation*.

Since undetectable secure computation is stronger than Anonymous Transfer (which it implies), our impossibility results for Anonymous Transfer also translate to impossibility results for undetectable secure computation[3]. Furthermore, building on our positive result, we show how to construct *anonymous oblivious transfer* (in the fine-grained security setting), a core building block for constructing undetectable secure computation for more general functionalities.

1.2 Defining Anonymous Transfer

An Anonymous Transfer (AT) protocol describes the interaction between a sender, a receiver and a non-participant. We assume all parties to interact in the synchronous model over a public broadcast channel, i.e., in each round each participant broadcasts a message which only depends on messages from previous rounds. The non-participant is not aware that a protocol takes place, and is only having an innocent conversation (we call them the "dummy player", or the "dummy friend"). We follow [vHL05; CGO+07] and model non-participating

[3] This follows directly from the fact that given undetectable secure computation for *any* function f, we can directly construct AT by computing a function that lets two potential senders insert either a bitstring for transfer or $\perp$ and outputs one of them (*i.e.* the one input that is not $\perp$) to the receiver.

parties as parties that only broadcast uniform randomness in each round, since any ordinary communication pattern can be viewed as an embedding of the uniform distribution due to standard techniques [vHL05; HLv02; vH04]. The sender aims to transmit a message to the receiver in a way that does not leak its identity (the notion easily generalizes to more non-participating parties). We say that an AT protocol is ε-correct if the probability that the receiver successfully receives the message is at least ε. Further, we say that an AT protocol is δ-anonymous if no adversary is able to determine the identity of the sender (given the transcript and the receiver's random tape) with advantage more than $(1-\delta)/2$ over guessing. These are the core properties which shape an AT protocol. If the protocol allows the receiver to remain silent throughout the protocol execution, sending a message corresponds to publicly revealing the message (*i.e.*, whistleblowing). Eventually, we call *fine-grained* AT an Anonymous Transfer, where anonymity is only required to hold against adversaries from a restricted complexity class (typically, adversaries whose runtime is bounded by a fixed polynomial in the runtime of the honest parties).

1.3 Impossibility Result

Our first main result shows that AT is impossible in a strong sense.

Theorem 1 (Impossibility of AT, informal). *There is no Anonymous Transfer protocol with overwhelming correctness and anonymity, with any polynomial number of rounds and any number $n \geq 1$ of non-participating parties, even for transmitting a single bit message.*

Our proof proceeds in several steps. First, we show that any Anonymous Transfer for transmitting a single bit with n non-participants, with overwhelming correctness and anonymity implies (in a black-box way) a *silent-receiver* Anonymous Transfer (where the receiver never speaks) for transmitting κ *bits* (where κ is some security parameter) with a single non-participating party. This reduction uses a relatively standard indistinguishability-based hybrid argument.

Then, the core of the proof rules out the existence of κ-bit silent-receiver 1-non-participant Anonymous Transfer with overwhelming correctness and anonymity. The key intuition is the following: let $\mathsf{P}_0, \mathsf{P}_1$ be the two parties interacting with the receiver, where P_b is the sender, and P_{1-b} is the non-participant. Let Π^{κ}_{AT} be the protocol which these two parties execute, and assume that it satisfies ε-correctness and δ-anonymity. Suppose that during their interaction, the parties produce a transcript π. We consider an adversary $\mathcal{A}$ which replaces the last message of P_0 by a random value, before running the receiver algorithm to reconstruct the transmitted message. Then if $b = 1$, the adversary just replaced the last (random) message of the non-participating party by another random message, and the transcript is still a perfectly valid transcript for Π^{κ}_{AT}, hence the reconstruction algorithm must still output the right string Σ with overall probability ε. On the other hand, if $b = 0$, then the transcript is a valid transcript for a "round-reduced" version of Π^{κ}_{AT}, where the last round is replaced by two

random messages. By the δ-anonymity, $\mathcal{A}$ should not distinguish between the two situations with advantage better than $(1-\delta)/2$. This implies that the correctness of the round-reduced protocol cannot be much lower than ε, hence that we constructed a δ-anonymous $(c-1)$-round protocol with non-trivial correctness guarantees. Then, $\mathcal{A}$ keeps repeating this procedure until we reach a 0-round protocol, which cannot possibly have any non-trivial correctness guarantee.

While the above provides an intuition of the approach, the real strategy is much more involved. In particular, using $\mathcal{A}$ to distinguish between a random transcript of Π^{κ}_{AT} and a random round-reduced transcript does not suffice to rule out arbitrary polynomial-round protocols (more precisely, it would only rule out logarithmic-round protocols, since the correctness guarantees would decrease roughly by a factor two at each step of round reduction). Instead, $\mathcal{A}$ will replace independently the last message of each party by a random value, getting two distinct transcripts (π_0, π_1). Then, $\mathcal{A}$ attempts to distinguish whether π_0 is a transcript of Π^{κ}_{AT} and π_1 is a round-reduced transcript, or the other way around. While this is the proper way to attack the protocol, the analysis is more involved, since now π_0, π_1 are not independent random variables anymore, as they share a common prefix (the transcript of the first $c-1$ rounds). Nevertheless, a more careful analysis shows that this dependency cannot significantly lower the distinguishing probability of $\mathcal{A}$.

In the full version [ACM21] we further prove that no AT protocol for $N > 3$ parties with overwhelming correctness and anonymity can exist unless a $N = 3$-party protocol exists with overwhelming correctness and anonymity—which cannot exist. It suffices to prove that any N-party Silent Receiver AT for $N > 3$ implies a (N)-party "normal" (*i.e.* with an actively participating receiver) AT, without losing the overwhelming correctness and anonymity in the process.

Intuitively, the receiver does not broadcast any messages in the N-party protocol; all communication comes from the $(N-1)$ potential senders. We construct an $(N-1)$-party protocol by letting the receiver play one non-participant, with the one difference being that this party is known not to be the sender (since it is the receiver); the sender can only be one of the $(N-2)$ other parties. While the correctness remains unaffected, the anonymity decreases due to the fact that *guessing* with one party less yields better results; yet we show that the anonymity still remains overwhelming in the security parameter. We then transform any N-party AT to an N-party SR-AT as described above and that to a $(N-1)$-party AT, until we have a 3-party AT that, assuming that the N-party AT has overwhelming anonymity and correctness, maintains these properties.

On a high level, this process lets the actual participants *simulate* non-participants behavior in their head; one-by-one their random tape is moved to the CRS until only three parties are left: a sender, a receiver, and a non-participant.

Our negative result applies to a weak model. In particular, non-participants are modeled to be semi-honest. Hence, our negative result does not leave much room for positive results.

1.4 A Candidate Fine-Grained Anonymous Transfer

To circumvent the above impossibility result, we need to give up asymptotic security and resort to the fine-grained setting: We only require anonymity against adversaries which require polynomially—quadratically, in our case—more resources than an honest protocol execution.

That is, our second main result shows that (perhaps surprisingly) non-trivial AT is indeed possible in a weaker setting:

Theorem 2 (Feasibility of AT, informal). *Let $N = 3$ be the number of individuals. Assuming ideal obfuscation, for any anonymity δ, there is a c-round Anonymous Transfer protocol Π^1_{AT} (for ℓ bit messages) that has overwhelming correctness, where anonymity δ holds against any adversary $\mathcal{A}$ with runtime $\ll c^2$.*

That is, for our second main contribution we propose a protocol which—assuming ideal obfuscation—allows to reduce the problem of de-anonymizing the sender to a distribution testing problem. More precisely, we show that determining the real sender in a c-round protocol given only a transcript of the AT protocol is as hard as differentiating between two *Bernoulli* oracles, where one returns 1 with probability p and the other returns 1 with $p + 1/(2c)$. For this distribution testing problem, strong lower bounds on the number of required samples and thus the adversarial runtime are known.

The protocol proceeds in rounds, where each honest message from the sender gradually increases the probability that the transmitted bit is correctly received. The sender first encrypts a verification key that is to-be-used be the obfuscated circuit, and in each successive round the sender encrypts the bit and a signature on both messages from the previous round to limit the ability of the adversary to manipulate the transcript when attacking anonymity. The non-participant only broadcasts random bits in each round. The Common Reference String contains an obfuscated program with hard-coded keys for the pseudorandom encryption scheme. The circuit checks the validity of the signatures of each round. Each consecutive valid round increases the confidence in the transmitted bit. Finally, the circuit outputs random bit according to the confidence gained. If all rounds are valid, the correct bit will be output with probability 1, if no round is valid, the correct bit will be output with probability 0.5.

While the high level intuition of the protocol is relatively clear, its exact instantiation is particularly delicate – any small variant in the design seems to open the avenue to devastating attacks. Furthermore, its analysis relies on long and complex hybrid arguments that progressively reduce the advantage of the adversary to contradictions with respect to known distribution testing bounds with a limited number of samples. The majority of our proof can be found in our full version [ACM21].

Our proof can be split in two parts. The first part exploits properties of the encryption schemes, the signature scheme, and ideal obfuscation to prove indistinguishability (against even PPT adversaries) between the actual protocol and a hybrid, where all reported messages are truly random and independent

from the sender and the transferred bit, and the obfuscated circuit only *counts* how many input messages are identical to those from the challenge transcript.

This game still contains information on the sending party as it treats those messages differently. To remove this dependency, we resort to *distribution testing* and view the obfuscated circuit as a *Bernoulli oracle* which follows one of two (known) distributions, and where the goal is to determine which one.

1.5 Discussions and Implications

In this section, we further discuss some implications and relations of our results to the literature.

'Philosophical implications:' between obfustopia and impossibilitopia. There is a small remaining gap between our negative and positive results: the possibility of building anonymous transfer secure against arbitrary polytime adversaries, but with non-negligible (e.g. inverse polynomial) anonymity error remains open. Closing this gap would have an intriguing philosophical consequence: stretching the terminology of Impagliazzo on the "worlds" of cryptography, it would establish the existence of a cryptographic primitive that plausibly exists in obfustopia (the world where indistinguishability obfuscation is possible) in the fine-grained setting, yet does not exist ("reside in impossibilitopia") with standard hardness gaps. Interestingly, there are several known examples where fine-grained constructions of a "higher world" primitive reside in a lower world; for example, (exponentially secure) one-way functions (a Minicrypt assumption) imply fine-grained public-key encryption (a Cryptomania assumption). Our work seems to provide a new example of this behavior, at the highest possible level of the hierarchy, showing that impossible primitives might end up existing if we weaken their security to the fine-grained setting.

Relation to the anonymous whistleblowing literature. We clarify how our (positive and negative) results relate to the literature on anonymous broadcast and secure whistleblowing. In general, a whistleblower willing to reveal something anonymously has two alternative choices: (1) the whistleblower has access to an anonymous communication channel, for example by putting their message (say, encrypted with the receiver public key) on some public website that somehow cannot be traced to them. However, access to an anonymous channel is typically a 'physical' assumption, and one which is *very* hard to guarantee. This issue is developed in great detail in the literature: see for example the discussion in Spectrum [NSSD21] about how metadata have been used by federal judges to trace and prosecute people who leaked data through secure messaging apps, or the discussion in Riposte [CBM15] and Express [ECZ+21] on how traffic analysis can be used to trace whistleblowers on the Tor network or the SecureDrop service. Hence, most of the literature focuses on scenario (2): the individuals interact over a communication network, and we do not assume that this network guarantees anonymity in itself. In this case, what we want is to *emulate* this anonymity, by

developing a strategy to help the whistleblower transmit a message anonymously to the receiver.

The literature on this subject is incredibly vast, but this emulated anonymity is *always* achieved using the same template in all solutions we are aware of (including Spectrum, Blinder, Riposte, Express, Talek, P3, Pung, Riffle, Atom, XRD, Vuvuzela, Alpenhorn, Stadium (or any other Mixnet-based solution), Karaoke, Dissent, Verdict, and many more): when the whistleblower wants to anonymously transmit a message, either to everyone (anonymous broadcast) or to a target receiver, other users generate 'honest' traffic in which communications can be hidden. To do so, the users interact with a set of *non-colluding* servers (sometimes two servers, sometimes more, some with honest majority, some without). This is never even discussed or remarked: it is taken as an obvious fact that this is *the* structure of an anonymous broadcast (or messaging) protocol. And indeed, the need to generate honest traffic feels clear – if the whistleblower is the sole sender, observing traffic directly leaks their identity. That the use of non-colluding servers was never challenged or even discussed probably means that it also *feels* clear – but this assumption is precisely what we challenge in our work: we do assume that some users generate honest traffic, but we ask whether the assumption of non-colluding participating servers is avoidable. Of course, any scientific treatment of a broad question ('are non-colluding helpful participants required for anonymous broadcast?') is bound to move from the broad question to a formal model, in which (feasibility or impossibility) results can be achieved. Nevertheless, we believe that our impossibility result demonstrates that the use of non-colluding servers in all previous works was indeed unavoidable, at least insofar as their aim was to achieve anonymity against arbitrary polynomial-time adversaries.

Non-participating parties versus malicious parties. Our choice of formalism, with the notion of anonymous transfer, allows to study whether the assumption of honest, non-colluding, participating servers can be replaced by a considerably weaker trust assumption: that of non-participating parties, not trying to take part to the protocol in any way (and not even required to be *aware* of the execution of the protocol) beyond generating traffic. As we show, this weaker assumption does not suffice against arbitrary polynomial-time adversaries, but possibly suffices against bounded polynomial-time adversaries (where the bound is sub-quadratic). As a natural next step, one could push the question even further and ask: what if some of the non-participating parties were in fact planted by a malicious adversary, and now play *maliciously* during the protocol? It seems plausible, that our general strategy can be extended to deal with malicious non-participants. However, we expect the analysis to require different techniques than the ones we used. We leave a formal proof of this to future work.

1.6 Further Results and Open Questions

In the full version [ACM21] we extend our fine-grained AT such that it transfers ℓ-bit messages directly, which achieves the same level of security as the single-bit

AT but requires twice as many rounds. We instantiate asymptotically secure AT in the designated-sender setting with non-trivial (but not useful) parameters for ε and δ. We define an extension of AT called *Strong AT* which we require for Undetectable Computation. We define undetectable versions of both OT (called *Undetectable Oblivious Transfer (UOT)*) and MPC (called Undetectable Multiparty Computation (UMPC)), where k parties hide the respective execution in a group of N individuals. We provide an instantiation of UOT based on strong AT and use that to instantiate UMPC for $k = 3$.

Our work leaves open two exciting questions:

(1) *Can our impossibility result for asymptotically secure AT with overwhelming correctness and anonymity be extended to rule out asymptotically secure AT with anonymity* $1 - 1/\mathsf{poly}(\kappa)$*?*
(2) *Is it possible to instantiate AT in the fine-grained setting from "Obfustopia" standard assumptions achieving similar parameter as our instantiation?*

Given that both our open questions can be answered affirmatively, this would separate the realm of asymptotic security from the realm of fine-grained security.

1.7 Acknowledgements

We thank Rafael Pass for insightful comments and contributions to early stages of this work.

2 Preliminaries

2.1 Notations

For any party P we denote by T_{P} the random tape of P.

For events (A, B), $\bar{A}$ denotes the complementary even of A, $\Pr[A \mid B]$ denotes the probability of A happening conditioned on B happening. For values (a, b), the notation $[\![a = b]\!]$ denote the bit value of the corresponding predicate. We let κ be a security parameter; we write $\mathsf{negl}(\kappa)$ to denote any function negligible in κ and $\mathsf{owhl}(\kappa)$ to denote a function overwhelming in κ (that is, $1 - \mathsf{owhl}(\kappa) = \mathsf{negl}(\kappa)$). For any probability distribution D, we denote by $\mathsf{Supp}(\mathsf{D})$ the support of D, and by $x \xleftarrow{\$} \mathsf{D}$ we denote that x is uniformly sampled from D.

For probability distributions p and q we write $p^{\otimes t}$ as the distribution arising from taking t sample from p, and $p \circ q$ as the distribution obtained by sampling one time from p and one time from q. We write $\|p\|_1$ to denote the L_1 norm of p.

For two bitstrings $A, B \in \{0,1\}^m$, $A \oplus B$ denotes the bitwise XOR of A and B. We write by $[n]$ for $n \in \mathbb{N}$ the set of numbers $\{1, \ldots, n\}$.

2.2 Distribution Testing

In this section, we introduce preliminaries for probability testing. We start by describing the *Total Variational Distance* between two distributions.

Definition 1 (Total Variational Distance). *Let p and q be two probability distributions over the countable set of possible outcomes Ω. The* total variational distance *between p and q is defined as:*

$$\mathsf{d}_{\mathsf{TV}}(p,q) \coloneqq \frac{1}{2}\sum_{\omega\in\Omega}|p(\omega)-q(\omega)| = \frac{1}{2}\|p-q\|_1 \tag{1}$$

An important property of the total variational distance is that it acts *sublinear* when taking many samples. When taking t samples from a Bernoulli distribution the corresponding distribution can be described by taking a single sample from a t-bit *Binomial* distribution. The sub-additivity then bounds the total variational distance of the corresponding binomial distribution:

Lemma 1 (Total variational distance of a t-fold probability distribution, folklore). *Let p and q be two Bernoulli distributions with total variational distance $\mathsf{d}_{\mathsf{TV}}(p,q)$. Then it holds for the binomial distributions $p^{\otimes t}$ and $q^{\otimes t}$ that result from sampling t times from the respective distributions:*

$$\mathsf{d}_{\mathsf{TV}}(p^{\otimes t}, q^{\otimes t}) \leq t \cdot \mathsf{d}_{\mathsf{TV}}(p,q) \tag{2}$$

Thus we can bound the distinguishing advantage of *any* distinguisher who has taken t samples form the same oracle using the total variational distance of the respective distributions directly.

A similar rule also holds for two *different* distributions, where the distinguisher has to distinguish whether two samples originate from $p \otimes r$ or from $q \otimes s$ for known values of p, q, r and s. In this case the rule states that:

Lemma 2 (Sub-Additivity of the Total Variational Distance for Product Distributions, folklore). *Let p and q be a probability distribution over $\{0,1\}^m$ with total variational distance $\mathsf{d}_{\mathsf{TV}}(p,q)$. Let r and s be two Bernoulli distributions with total variational distance $\mathsf{d}_{\mathsf{TV}}(r,s)$. Then it holds for the distribution derived from sampling from each distribution once and concatenating the outputs (which yields a sample from $\{0,1\}^{m+1}$ originating either from $p \circ r$ or $q \circ s$) that*

$$\mathsf{d}_{\mathsf{TV}}(p \circ r, q \circ s) \leq \mathsf{d}_{\mathsf{TV}}(p,q) + \mathsf{d}_{\mathsf{TV}}(r,s)$$

The following lemma limits the distinguishing advantage of any distinguisher that tries to distinguish two distributions p and q based on a single sample.

Lemma 3 (Distinguishing distributions based on the Total Variational Distance). *Let p and q be two distributions with total variational distance $\mathsf{d}_{\mathsf{TV}}(p,q)$. If $\mathsf{d}_{\mathsf{TV}}(p,q) < \frac{1}{3}$, then no algorithm can exist that distinguishes p and q with probability $\geq \frac{2}{3}$ based on a single sample.*

Using Lemmas 1 and 3 we can provide lower bounds on the sampling complexity of distinguishing two distributions p and q with advantage $\alpha/2$.

Corollary 1 (Distinguishing two Bernoulli-Distributions with t samples). *Any distinguisher $\mathcal{D}$ that distinguishes between p and q with probability $\geq \frac{1}{2} + \frac{\alpha}{2}$ requires $t \in \Omega\left(\frac{\alpha}{\mathsf{d}_{\mathsf{TV}}(p,q)}\right)$ samples.*

We refer the reader to [ACM21] for proofs of Lemma 3 and Corollary 1.

3 Anonymous Transfer

We consider the following situation: some secret agent P_b is willing to transfer a message Σ to a receiver R, while hiding his identity b among two individuals. We call Anonymous Transfer (AT) an interactive protocol that achieves this goal.

3.1 Network Model and Non-Participating Parties

The goal of an anonymous transfer protocol is to hide the transferred message among innocent conversations by individuals, which are not taking part in the protocol. By a well-established folklore result in steganography, this task can be reduced to the simpler task of hiding the transferred message among *uniformly random beacons*, broadcast by the other individuals: the uniform channel, where all protocol messages look uniformly random, can be compiled into any other ordinary communication pattern [vHL05; HLv02; vH04]. Therefore, as in previous works (see von Ahn, Hopper, and Langford [vHL05] and Chandran, Goyal, Ostrovsky, and Sahai [CGO+07]), we consider a set of k parties who interact with each other via broadcast channels and focus, without loss of generality, on protocols for the uniform channel. Consequently, we will model the non-participating parties as "dummy parties" that only broadcast uniformly random messages of a fixed length at each round.

3.2 The Model

Let $b \in \{1, \cdots, N-1\}$ denote the index of the sender and let $\Sigma \in \{0,1\}^\ell$ be the message that P_b wants to transfer to the receiver. We consider an interactive protocol in the Common Reference String (CRS) model between N players $(\mathsf{P}_1, \cdots, \mathsf{P}_{N-1}, \mathsf{R})$, where R and P_b participate in the protocol, and P_i for $i \neq b$ are non-participating but present players that only broadcast random strings. The receiver R gets the CRS as input and the sender P_b gets the CRS and the message Σ as input. For any player P, let T_{P} denote the random tape from which P draws his random coins. The players interact through authenticated broadcast channels in the synchronous model: the protocol proceeds in rounds, and each player broadcasts a message at each round. We denote by $\langle \mathsf{R}, \mathsf{P}_1, \cdots, \mathsf{P}_{N-1} \rangle (crs, b, \Sigma)$ the distribution of the possible transcripts of the protocol in this setting (*i.e.*, the sequence of all messages broadcasted by the players during an execution of the protocol), where the probabilities are taken over the random coins T_{P} of the players $\mathsf{P} \in \{\mathsf{R}, \mathsf{P}_1, \cdots, \mathsf{P}_{N-1}\}$ and the random choice of the CRS *crs*.

Definition 2 ($(\varepsilon, \delta, c, \ell)$-Anonymous Transfer). *An N-party $(\varepsilon, \delta, c, \ell)$-Anonymous Transfer (AT) for $\varepsilon, \delta \in \mathbb{R}_{[0,1]}$ and $N, c, \ell \in \mathbb{N}$ (all possibly functions in κ) is a tuple containing three PPT algorithms $(\mathsf{Setup}, \mathsf{Transfer}, \mathsf{Reconstruct})$. The number of rounds in the $\mathsf{Transfer}$ protocol is given as c and the bitlength ℓ defines the length of the transferred message Σ. The algorithms are defined as follows:*

$\mathsf{Setup}(1^\kappa)$ *takes as input the security parameter 1^κ in unary encoding and outputs a Common Reference String crs.*

$\mathsf{Transfer}(crs, b, \Sigma)$ *defines a c-round protocol*[4] *that takes as input the Common Reference String crs, an index* $b \leq N-1$ *specifying the sender, and the message* $\Sigma \in \{0,1\}^\ell$ *from the sender and outputs a transcript* π*. The non-sender sends independent uniformly distributed noise in each round. All protocol messages sent by the receiver, the sender and the non-participating parties at each round are bitstrings of length* $m = m(\kappa)$*, where* m *is implicitly specified by the* Transfer *protocol.*

$\mathsf{Reconstruct}(crs, \pi, T_\mathsf{R})$ *is a local algorithm executed by the receiver that takes as input the CRS crs, the protocol transcript* π *and the receiver's random tape* T_R *and outputs a message* Σ'*.*

The algorithms additionally satisfy the ε*-correctness and the* δ*-anonymity properties defined in Definitions 3 and 4.*

Definition 3 (ε-Correctness). *For any sufficiently large security parameter* κ*, for any number of individuals* $N \in \mathsf{poly}(\kappa)$*, for any participant* $b \in [N-1]$*, for any message length* $\ell \in \mathsf{poly}(\kappa)$*, for any message* $\Sigma \in \{0,1\}^\ell$*, and for any CRS* $crs \leftarrow \mathsf{Setup}(1^\kappa)$*, an Anonymous Transfer protocol* Π^ℓ_{AT} *between players* $(\mathsf{P}_1, \ldots, \mathsf{P}_{N-1}, \mathsf{R})$ *is* ε-correct *if the following holds:*

$$\Pr\left[\begin{array}{c} \pi \xleftarrow{\$} \mathsf{Transfer}_{\langle \mathsf{R}, \mathsf{P}_1, \ldots, \mathsf{P}_{N-1}\rangle}(crs, b, \Sigma) \\ \Sigma' \leftarrow \mathsf{Reconstruct}(crs, \pi, T_\mathsf{R}) \end{array} : \Sigma = \Sigma'\right] \geq \varepsilon \tag{3}$$

Note that ε *can take on any value between* 0 *and* 1*. The naive algorithm that lets the receiver sample a uniformly random* ℓ*-bit string has* $\varepsilon = 1/2^\ell$*.*

Definition 4 (δ-Anonymity). *For any PPT algorithm* $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$*, for all sufficiently large security parameters* κ*, for any number of individuals* $N \in \mathsf{poly}(\kappa)$*, and for any message length* $\ell \in \mathsf{poly}(\kappa)$*, an Anonymous Transfer protocol* Π^ℓ_{AT} *between players* $(\mathsf{P}_1, \ldots, \mathsf{P}_{N-1}, \mathsf{R})$ *is* δ-anonymous *if it holds that*

$$\left| \Pr_{b \xleftarrow{\$} [N-1]}\left[\mathrm{Exp}^{\mathit{anon}}_{\Pi^\ell_{AT}, \mathsf{A}, b}(\kappa) = b\right] - \frac{1}{N-1} \right| \leq (1-\delta) \cdot \frac{N-2}{N-1} \tag{4}$$

where $\mathrm{Exp}^{\mathit{anon}}_{\Pi^\ell_{AT}, \mathsf{A}, b}(\kappa)$ *is defined in Fig. 1.*

The value δ can take any value between 0 and 1. The higher δ the stronger the provided anonymity guarantees. If a protocol is $\delta = 1$-anonymous, the advantage over guessing at random equals 0, and if a protocol is $\delta = 0$-anonymous, the advantage over guessing at random equals 1. The right-hand-side of Definition 4 contains a scaling factor of $(N-1)/(N-2)$. This is due to the fact that even under perfect anonymity ($\delta = 1$), the receiver can still *guess* the sender. Knowing that one of the N parties—namely itself—is *not* the sender, there are $(N-1)$

[4] A c-round protocol corresponds to a synchronous model, where each message is broadcasted and the messages in each round only depend on messages from previous rounds, see [ACM21] for a formal definition.

$\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{A},N,b}(\kappa)$

$crs \xleftarrow{\$} \mathsf{Setup}(1^{\kappa})$

$T_{\mathsf{R}} \xleftarrow{\$} \{0,1\}^{\mathsf{poly}(\kappa)}$

$(\Sigma, st) \leftarrow \mathsf{A}_0(crs, T_{\mathsf{R}})$

$\pi \xleftarrow{\$} \mathsf{Transfer}_{\langle \mathsf{R},\mathsf{P}_1,\ldots,\mathsf{P}_{N-1}\rangle}(crs, b, \Sigma; T_{\mathsf{R}}, \cdot, \cdot)$

return $\mathsf{A}_1(\pi, T_{\mathsf{R}}, st)$

Fig. 1: Definition of the game $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\ell}_{AT},\mathsf{A},b}(\kappa)$.

potential senders, of which $(N-2)$ are just dummy friends. Thus, the probability of *guessing wrong* is given by the aforementioned factor.

Note that we require anonymity to hold, in particular, against the receiver. Therefore, the adversary in the anonymity game may know the receiver's random tape T_{R} from the beginning.

The guessing algorithm is split between A_0 who is given the CRS and the random tape T_{R} the receiver is going to use during the protocol, and outputs the target message Σ that should be transferred and a state st. In the second phase, the algorithm A_1 which is given the inputs π and the state.

Unless stated otherwise, we consider the case $N = 3$, i.e., one non-participant.

3.3 Fine-grained Anonymous Transfer

Fine-grained cryptographic primitives are only secure against adversaries with an a-priori bounded runtime which is greater than the runtime of the honest algorithms, [Mer78; DVV16]. We use the notion of [DVV16]. In the following, $\mathfrak{C}_1$ and $\mathfrak{C}_2$ are function classes.

Definition 5 ($\mathfrak{C}_1$-fine-grained $(\varepsilon, \delta, c, \ell)$-Anonymous Transfer against $\mathfrak{C}_2$). *The tuple* $(\mathsf{Setup}, \mathsf{Transfer}, \mathsf{Reconstruct})$ *(as defined in Definition 2) is a* $\mathfrak{C}_1$-fine-grained $(\varepsilon, \delta, c, \ell)$-Anonymous Transfer *for* $\varepsilon, \delta \in \mathbb{R}_{[0,1]}$ *and* $c, \ell \in \mathbb{N}$ *against* $\mathfrak{C}_2$ *if the following two conditions hold:*

Efficiency. *The algorithms* $(\mathsf{Setup}, \mathsf{Transfer}, \mathsf{Reconstruct})$ *are in* $\mathfrak{C}_1$.
Security. *Anonymity (Definition 4) is only required to hold against adversaries in* $\mathfrak{C}_2$.

The definition of correctness remains as in Definition 3.

Example 1 (Merkle-Puzzles). Merkle-Puzzles [Mer78] are a fine-grained protocol to exchange a shared key from symmetric encryptions where successful encryptions can be efficiently distinguished from false ones. The sender S creates n_{mer} many ciphertexts, each under a different (relatively short) key, containing a unique identifier and a symmetric key. The receiver R then randomly picks one of the ciphertexts and runs a brute-force attack (which we assume to cost m_{mer} many steps) to recover the key and to send the identifier back to the sender.

Here $\mathfrak{C}_1 \coloneqq \mathcal{O}(n_{\text{mer}} + m_{\text{mer}})$ as the sender has to create n_{mer} puzzles and the receiver must use m_{mer} steps to break one of them, and $\mathfrak{C}_2 \coloneqq \mathcal{O}(n_{\text{mer}} \cdot m_{\text{mer}})$ as an adversary has to break at worst all the n_{mer} ciphertexts to recover the key.

3.4 Trivial Anonymous Transfers

For simplicity, we focus on 3-party anonymous transfer in the following discussions, with two players $\mathsf{P}_0, \mathsf{P}_1$ and a receiver R.

Remark 1 (Perfect correctness.). A perfectly correct (*i.e.* $\varepsilon = 1$) protocol is impossible. Given a player P_b with input Σ, there is always a probability that the non-participating player P_{1-b} behaves exactly as a participating player with input $\Sigma' \neq \Sigma$, in which case R cannot obtain the correct output for sure.

Therefore, the best one can hope for is a correctness statistically close to 1. In the following, we demonstrate ATs with trivial parameters.

Example 2 (Trivial single-bit AT). Consider the following trivial single-round AT to transfer a single bit σ: P_b broadcasts his input σ (and P_{1-b} broadcasts a random bit). Upon receiving (σ_0, σ_1) from P_0 and P_1, if $\sigma_0 = \sigma_1$, R outputs σ_0; otherwise, R outputs a uniformly random bit. As P_{1-b} broadcasts a random bit, it holds that $\sigma_0 = \sigma_1$ with probability $1/2$, in which case R obtains the correct output $\sigma = \sigma_b$; else, R obtains the correct output with probability $1/2$. Overall, R obtains the correct output with probability $3/4$. The protocol is $1/2$-anonymous since the adversary knows the message to be transmitted and can hence determine the sender whenever the transmitted bits are distinct and guess with probability $1/2$ otherwise. Hence, the above protocol is a $(3/4, 1/2, 1, 1)$-AT.

Example 3 (Trivial ℓ-bit AT). One can also construct a trivial ℓ-bit AT. To transmit a message $\Sigma \in \{0,1\}^\ell$: P_b simply sends Σ repeated κ times. Clearly, (not only) R finds out both Σ and b with overwhelming probability. Hence, the above protocol is a $(1 - \mathsf{negl}(\kappa), \mathsf{negl}(\kappa), \kappa \cdot \ell, \ell)$-AT.

In this work, we study whether ATs with non-trivial parameters can exist.

3.5 Reductions Among AT Protocols

In this section, we show that several simplified variants of anonymous transfer are equivalent to the original definition.

AT implies silent-receiver AT. We say that an anonymous transfer has *silent receiver* if the receiver never sends messages during the $\mathsf{Transfer}$ protocol, and $\mathsf{Reconstruct}$ is a deterministic function of the CRS and the transcript π. Any AT directly implies a silent-receiver AT with the same parameters for correctness and anonymity, but at the cost of secrecy: Any (non-)participant is able to reconstruct the message given only the transcript of broadcasted messages, not just the receiving party of the protocol, which might be undesirable for practical applications. Let Π^ℓ_{AT} be a $(\varepsilon, \delta, c, \ell)$-Anonymous Transfer. Define the silent-receiver AT Π^ℓ_{SR} as follows:

$\Pi^{\ell}_{SR}.\mathsf{Setup}(1^{\kappa})$ runs $crs \leftarrow \Pi^{\ell}_{AT}.\mathsf{Setup}(1^{\kappa})$ and samples a uniform random tape T_{R} for R. It outputs (crs, T_{R}).

$\Pi^{\ell}_{SR}.\mathsf{Transfer}(crs, b, \Sigma)$ proceeds exactly as $\Pi^{\ell}_{AT}.\mathsf{Transfer}(crs, b, \Sigma)$, except that the receiver R does not broadcast any message. At each round $\chi = 1$ to $\chi = c$, the sender P_b locally appends the χ-th receiver message x_χ in $\Pi^{\ell}_{AT}.\mathsf{Transfer}(crs, b, \Sigma; T_{\mathsf{R}}, \cdot, \cdot)$ to the current transcript $\pi[\chi]$ (note that x_χ can be computed deterministically from $\pi[\chi]$ and T_{R}), and compute its next message as in $\Pi^{\ell}_{AT}.\mathsf{Transfer}$ using the transcript $\pi[\chi] \| x_\chi$.

$\Pi^{\ell}_{SR}.\mathsf{Reconstruct}(crs, \pi, T_{\mathsf{R}})$ is defined exactly as $\Pi^{\ell}_{AT}.\mathsf{Reconstruct}(crs, \pi, T_{\mathsf{R}})$, except that it first expands the transcript π by recomputing (deterministically) the messages of R in $\Pi^{\ell}_{AT}.\mathsf{Transfer}(crs, b, \Sigma; T_{\mathsf{R}}, \cdot, \cdot)$ and appending them to π at each round.

The notion of silent receiver AT captures the notion of an anonymous transfer whose aim is to *publicly reveal* a message (*i.e.*, whistleblowing) rather than sending it to a single receiver. An other way to look at it is to consider that the silent receiver transformation can be seen as *passive to active security transformation* for the receiver: If there is a secure AT protocol against a *passive* receiver, then there is a secure silent receiver AT against an *active* receiver, simply because the receiver has no option to cheat as no messages are sent.

Lemma 4. *Π^{ℓ}_{SR} is an $(\varepsilon, \delta, c, \ell)$-Anonymous Transfer.*

Proof (sketch). Correctness and number of rounds follow directly from the description of Π^{ℓ}_{SR}, which simply mimics Π^{ℓ}_{AT}, except that the random tape of the receiver is made public, and its messages are computed on the fly locally by the sender and during the reconstruction. Anonymity follows also immediately by observing that T_{R} is given to the adversary in the anonymity game, hence making it public cannot harm anonymity. □

Since the converse direction is straightforward, AT and silent receiver AT are therefore equivalent.

Single-bit AT implies many-bit AT. In this section, we analyze how a single-bit AT can be generically transformed into an AT which allows to transmit bitstrings. We construct an ℓ-bit AT by executing the single-bit AT ℓ times (sequentially) to transmit the message bit-by-bit. Let Π^{1}_{AT} be a $\mathfrak{C}_1$-fine-grained-$(\varepsilon, \delta, c, 1)$-Anonymous Transfer against $\mathfrak{C}_2$. Further, let Π^{ℓ}_{AT} be the protocol which uses ℓ instances of Π^{1}_{AT} to transmit ℓ-bit messages bit-by-bit.

We analyze Π^{ℓ}_{AT} using the fine-grained definition. The results directly apply using asymptotic security.

Lemma 5. *Let Π^{1}_{AT} be a $\mathfrak{C}_1$-fine-grained $(\varepsilon, \delta, c, 1)$-Anonymous Transfer against $\mathfrak{C}_2$. Then, the protocol Π^{ℓ}_{AT} is a $\mathfrak{C}'_1 := \mathfrak{C}_1 \cdot \ell$-fine-grained $\big(\varepsilon', \delta', c \cdot \ell, \ell\big)$-AT against $\mathfrak{C}'_2 := \mathfrak{C}_2 - \ell \cdot \mathfrak{C}_1$, where $\varepsilon' = \varepsilon^{\ell}$ and $\delta' = (\delta\ell - \ell - \delta + 2)$.*[5]

[5] We slightly abuse notation but we believe the meaning to be clear.

Proof. For $\Sigma \in \{0,1\}^\ell$, we have $\varepsilon' = \Pr_{crs,\pi,\Sigma'}[\Sigma = \Sigma'] = \varepsilon^\ell$.

For the purpose of avoiding notational overhead, we prove anonymity for $N = 3$ parties, i.e., for one non-participant. The general case follows by generalizing notation. Let A be an adversary against the anonymity of Π^ℓ_{AT}. We define a sequence of hybrid games $\mathrm{H}_1, \ldots, \mathrm{H}_\ell$ between $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\ell_{AT},\mathsf{A},0}(\kappa)$ and $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\ell_{AT},\mathsf{A},1}(\kappa)$ in Fig. 2. H_1 is identical to $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\ell_{AT},\mathsf{A},1}(\kappa)$ and H_ℓ is identical to $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\ell_{AT},\mathsf{A},0}(\kappa)$.

We construct an adversary B against the anonymity of Π^1_{AT} in Fig. 2. If B plays $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^1_{AT},\mathsf{B},0}(\kappa)$, then B simulates H_{i+1} for A. Otherwise, if B plays $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^1_{AT},\mathsf{B},1}(\kappa)$, then B simulates H_i for A.

H_i

for $j \in [\ell]$ **do**
  $crs_j \leftarrow \mathsf{Setup}(1^\kappa)$
$crs' := (crs_1, \ldots, crs_\ell)$
$T'_{\mathsf{R}} := (T_{\mathsf{R},1}, \ldots, T_{\mathsf{R},\ell}) \leftarrow (\{0,1\}^{\mathsf{poly}(\kappa)})^\ell$
$(\Sigma, st_{\mathsf{A}}) \leftarrow \mathsf{A}_0(crs', T'_{\mathsf{R}})$
for $j \in \{1, \ldots, i-1\}$ **do**
  $\pi_j \leftarrow \mathsf{Transfer}(crs, 0, \Sigma[j]; T_{\mathsf{R},i}, \cdot, \cdot)$
for $j \in \{i, \ldots, \ell\}$ **do**
  $\pi_j \leftarrow \mathsf{Transfer}(crs, 1, \Sigma[j]; T_{\mathsf{R},i}, \cdot, \cdot)$
return $\mathsf{A}_1((\pi_1, \ldots, \pi_\ell), st_{\mathsf{A}})$

$\mathsf{B}_0(crs, T_{\mathsf{R}})$

$i \leftarrow \{1, \ldots, \ell - 1\}$
for $j \in [\ell] \setminus \{i\}$ **do**
  $crs_j \leftarrow \mathsf{Setup}(1^\kappa)$
  $T_{\mathsf{R},j} \leftarrow \{0,1\}^{\mathsf{poly}(\kappa)}$
$crs_i := crs$, $T_{\mathsf{R},i} := T_{\mathsf{R}}$
$crs' := (crs_1, \ldots, crs_\ell)$
$T'_{\mathsf{R}} := (T_{\mathsf{R},1}, \ldots, T_{\mathsf{R},\ell})$
$(\Sigma, st_{\mathsf{A}}) \leftarrow \mathsf{A}_0(crs', T'_{\mathsf{R}})$
$st := (\Sigma, i, st_{\mathsf{A}})$
return $(\Sigma[i], st)$

$\mathsf{B}_1(\pi, st)$

parse $st =: (\Sigma, i, st_{\mathsf{A}})$
for $j \in \{1, \ldots, i-1\}$ **do**
  $\pi_j \leftarrow \mathsf{Transfer}(crs, 0, \Sigma[j]; T_{\mathsf{R},j}, \cdot, \cdot)$
for $j \in \{i+1, \ldots, \ell\}$ **do**
  $\pi_j \leftarrow \mathsf{Transfer}(crs, 1, \Sigma[j]; T_{\mathsf{R},j}, \cdot, \cdot)$
$\pi_i := \pi$
return $\mathsf{A}_1((\pi_1, \ldots, \pi_\ell), st_{\mathsf{A}})$

Fig. 2: Hybrid games for the expansion of single-bit AT to multi-bit AT (left) and the adversary (middle and right).

Provided that B is in $\mathfrak{C}_2$, we have

$$\begin{aligned} 1 - \delta &\geq |\Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\ell_{AT},\mathsf{B},0}(\kappa)] - \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\ell_{AT},\mathsf{B},1}(\kappa)]| \\ &= \frac{1}{\ell - 1}(\Pr[\mathrm{H}_\ell] - \Pr[\mathrm{H}_1]) = \frac{1}{\ell - 1}\Big(\Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\ell_{AT},\mathsf{A},0}(\kappa)] - \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^\ell_{AT},\mathsf{A},1}(\kappa)]\Big) \end{aligned}$$

We have that $\mathfrak{T}(\mathsf{B}) = \mathfrak{T}(\mathsf{A}) + (\ell - 1) \cdot \mathfrak{C}_1 = \mathfrak{T}(\mathsf{A}) + \ell \cdot \mathfrak{C}_1$. Hence, given that $\mathfrak{T}(\mathsf{A}) = \mathfrak{T}(\mathsf{B}) - \mathfrak{C}_1 \in \mathfrak{C}_2 - \ell \cdot \mathfrak{C}_1$, the anonymity advantage of A is $(1 - \delta)(\ell - 1)/2$, yielding anonymity of $\delta' = \delta\ell - \ell - \delta + 2$. □

4 Impossibility of Anonymous Transfer

In this section, we prove that no anonymous transfer protocol, with an arbitrary polynomial number of rounds, can simultaneously enjoy overwhelming correctness ($\varepsilon = 1 - \mathsf{negl}(\kappa)$) and overwhelming anonymity ($\delta = 1 - \mathsf{negl}(\kappa)$), even for transmitting single bit messages.

Theorem 3 (Impossibility of AT). *Let $\mu : \mathbb{N} \mapsto \mathbb{R}$ be any negligible function and p be any polynomial. There is no $(1 - \mu(\kappa), 1 - \mu(\kappa), p(\kappa), 1)$-Anonymous Transfer, for any number of parties.*

Theorem 3 will follow as a corollary from a more general result bounding the relation between ε and δ in any c-round protocol. Throughout this section we will focus on $N = 3$, that is, the case with one dummy player. This is without loss of generality as we will show in the full version [ACM21] that any N-party anonymous transfer with $N > 3$ implies in particular a 3-party anonymous transfer, for which we will show here that it can not exist.

4.1 The Attacker

From now on, we focus on building a generic attack against 3-party *silent-receiver* anonymous transfer for κ-bit messages. The theorem will follow from the reductions from 1-bit anonymous transfer to multibit silent-receiver anonymous transfer described in Section 3.5.

Let Π^{κ}_{AT} be a silent-receiver $(\varepsilon, \delta, c, \kappa)$-Anonymous Transfer. Let $m = m(\kappa)$ be the bitlength of the message from the non-participating party. Let Rand denote the following procedure: on input a transcript π of Π^{κ}_{AT}, $\mathsf{Rand}(\pi)$ truncates π to $c-1$ rounds of the AT protocol, and replaces the messages of the last round by two uniformly random length-m bitstrings[6]. It outputs the new rerandomized transcript π'. For every $\Sigma \in \{0,1\}^{\kappa}$ and $b \in \{0,1\}$, we let $\mathcal{D}_{b,\Sigma}, \mathcal{D}'_{b,\Sigma}, \mathcal{DR}$ denote the following distribution:

$$\mathcal{D}_{b,\Sigma} = \{\Sigma' \ : \ crs \leftarrow \mathsf{Setup}(1^{\kappa}), \pi \leftarrow \mathsf{Transfer}(b, \Sigma), \Sigma' \leftarrow \mathsf{Reconstruct}(crs, \pi)\}$$

$$\mathcal{D}'_{b,\Sigma} = \left\{ \Sigma' \ : \ \begin{array}{l} crs \leftarrow \mathsf{Setup}(1^{\kappa}), \\ \pi' \leftarrow \mathsf{Rand}(\mathsf{Transfer}(b, \Sigma)), \\ \Sigma' \leftarrow \mathsf{Reconstruct}(crs, \pi') \end{array} \right\}, \quad \mathcal{DR} = \left\{ \Sigma' \ : \ \begin{array}{l} crs \leftarrow \mathsf{Setup}(1^{\kappa}), \\ \pi' \xleftarrow{\$} (\{0,1\}^m \times \{0,1\}^m)^c, \\ \Sigma' \leftarrow \mathsf{Reconstruct}(crs, \pi') \end{array} \right\}$$

Fix an arbitrary polynomial t. We define an attacker $\mathcal{A}^t = (\mathsf{A}^t_0, \mathsf{A}^t_1)$ against the anonymity of Π^{κ}_{AT}, parameterized by the polynomial t, on Figure 3. In the following, we will not use $\mathcal{A}^t$ directly to attack the full c-round protocol: rather, we will use $\mathcal{A}^t$ as a distinguisher between the c-round protocol Π^{κ}_{AT}, and the $(c-1)$-round protocol obtained by running Π^{κ}_{AT} for $(c-1)$ rounds, and replacing the messages of the last round by uniformly random m-bit strings. From there, the proof of impossibility will proceed by induction; we refer the reader to the introduction for a high-level intuition of our proof.

Base case: advantage of $\mathcal{A}^t$ when $c = 1$. We start the induction by bounding the advantage of $\mathcal{A}^t$ in the anonymity game when Π^{κ}_{AT} is non-interactive (i.e., $\mathsf{Transfer}$ consists of a single message from each of $\mathsf{P}_0, \mathsf{P}_1$ to the receiver). Before proceeding, we make two key observations:

[6] Since the protocol is silent-receiver, there is no message from the receiver; furthermore, assuming that the sender message is m-bit is without loss of generality, since otherwise the protocol is trivially not anonymous.

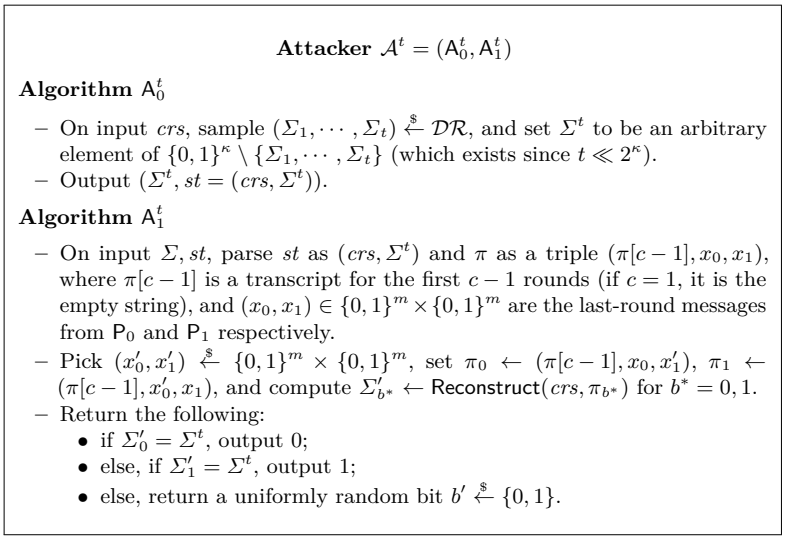

Fig. 3: Attacker $\mathcal{A}^t$ against the δ-anonymity of the silent-receiver κ-bit AT protocol Π_{AT}^κ, parameterized by a polynomial $t = t(\kappa)$.

(1) When $c = 1$, $\mathcal{D}'_{b,\Sigma} = \mathcal{DR}$ for any (b, Σ). In particular, this means that $\mathcal{D}'_{b,\Sigma}$ is independent of (b, Σ).

(2) When $c = 1$ and $b = 0$, the distribution of the values (Σ'_0, Σ'_1) constructed by A_1^t given as input a random transcript $\pi \leftarrow \mathsf{Transfer}(0, \Sigma^t)$ is exactly the distribution $\mathcal{D}_{0,\Sigma^t} \times \mathcal{DR}$. This is because x_0 is a random message from the sender with input $b = 0$ and value Σ^t, and (x_1, x'_0, x'_1) are three uniformly random elements of $\{0,1\}^m$, hence (x_0, x'_1) is exactly a random transcript of Π_{AT}^κ with (b, Σ^t), while (x'_0, x_1) is just a pair of random messages. Similarly, if $b = 1$, the distribution of the values (Σ'_0, Σ'_1) constructed by A_1^t given as input a random transcript $\pi \leftarrow \mathsf{Transfer}(1, \Sigma^t)$ is exactly the distribution $\mathcal{DR} \times \mathcal{D}_{1,\Sigma^t}$.

Both observations follow directly from the definitions of $\mathcal{D}_{b,\Sigma}, \mathcal{D}'_{b,\Sigma}, \mathcal{DR}$ and of A_1^t. Building on the above observations, we show that for an appropriate choice of t, the advantage of $\mathcal{A}^t$ in the anonymity game can be made arbitrarily close to $(\varepsilon - 1)/2$:

Claim. For any polynomial n, there is a polynomial t such that

$$\left| \Pr_{b \xleftarrow{\$} \{0,1\}} \left[\mathrm{Exp}^{\mathsf{anon}}_{\Pi_{AT}^\kappa, \mathcal{A}^t, b}(\kappa) = b \right] - 1/2 \right| \geq \frac{\varepsilon}{2} - \frac{1}{n}, \tag{5}$$

which implies that any silent-receiver $(\varepsilon, \delta, 1, \kappa)$-Anonymous Transfer must satisfy $\delta \leq 1 - \varepsilon + 2/n$ for any polynomial n; equivalently, $\delta \leq 1 - \varepsilon + \mathsf{negl}(\kappa)$. In particular,

this means that if the AT has overwhelming correctness ($\varepsilon = 1 - \mathsf{negl}(\kappa)$), then δ must be negligible.

The proof for this claim can be found in the full version [ACM21].

4.2 Putting the Pieces Together

With the above analysis, we showed that for any silent-receiver $(\varepsilon, \delta, c, \kappa)$-Anonymous Transfer, it must necessarily hold that $(1-\delta)/2 \geq \varepsilon/2c - \mathsf{negl}(\kappa)$. Since any $(\varepsilon, \delta, c, \kappa)$-Anonymous Transfer implies a silent-receiver $(\varepsilon, \delta, c, \kappa)$-Anonymous Transfer (with the exact same parameters, see Section 3.5), we obtain:

Corollary 2. *Any $(\varepsilon, \delta, c, \kappa)$-Anonymous Transfer must satisfy*

$$\frac{1-\delta}{2} \geq \frac{\varepsilon}{2c} - \mathsf{negl}(\kappa).$$

In particular, this implies that there exists no κ-bit AT with overwhelming correctness and anonymity, for any polynomial number of rounds.

Furthermore, as shown in Section 3.5, any *single-bit* c-round AT with correctness $\varepsilon = 1 - \mathsf{negl}(\kappa)$ and anonymity $\delta = 1 - \mathsf{negl}(\kappa)$ implies a κ-bit AT with correctness $\varepsilon' = \varepsilon^\kappa = (1 - \mathsf{negl}(\kappa))^\kappa = 1 - \mathsf{negl}(\kappa)$, and anonymity $\delta' = (\delta - 1) \cdot \kappa - \delta + 2 = 1 - \mathsf{negl}(\kappa)$. Combining this reduction with Corollary 2 concludes the proof of Theorem 3.

4.3 Extensions and Limitations

The adversary in our impossibility result makes a black-box use of an arbitrary 3-party silent receiver multibit anonymous transfer; the reduction to N-party single-bit anonymous transfer is black-box as well. In particular, this means that our impossibility result relativizes: it remains true relative to any oracle, where access to the oracle is granted to all participants and all algorithms (including the adversary).

In the next section, we will provide a heuristic construction of *fine-grained* anonymous transfer. The aim of this construction is to complement our impossibility result, and to draw an interesting and surprising picture: anonymous transfer appears to be impossible to realize with the standard superpolynomial cryptographic hardness gaps, but becomes feasible if one settles for a small polynomial hardness gap. Our fine-grained construction is described and formally proven secure using an ideal obfuscation scheme; instantiating the scheme with candidate indistinguishability obfuscation schemes gives a plausible heuristic construction (the same way that instantiating the random oracle model with standard hash functions gives plausible heuristic constructions of various cryptographic primitives, when the construction is not pathological). Because our impossibility result relativizes, in contrast, standard anonymous transfer remains provably impossible relative to an ideal obfuscation oracle (while fine-grained anonymous transfer, as we will see, provably exist relative to such an oracle).

Impossibility of fine-grained multibit AT with overwhelming correctness and anonymity. In the multibit setting, where the sender wants to transmit $\omega(\log \kappa)$ bits to the receiver, our result further demonstrates that there exists no *fine-grained* anonymous transfer with overwhelming correctness and anonymity $1 - \mathsf{negl}(\kappa)$, even with an *arbitrary small polynomial gap* between the runtime of the honest parties and that of the adversary. Indeed, let $r = \mathcal{O}(c \cdot m)$ be a lower bound on the runtime of the honest parties (r is the total number of bits sent by the sender, hence it is a clear lower bound on its running time), and consider an adversary $\mathcal{A}^t$ with $t = \kappa \cdot c^g$, where $g > 0$ is an arbitrarily small constant. Then by construction, the runtime of $\mathcal{A}^t$ is $\mathcal{O}(\kappa \cdot r \cdot c^g) \leq \mathcal{O}(\kappa \cdot r^{1+g})$ (as it is dominated by the cost of sampling t random transcripts for A_0^t). Then this adversary satisfies

$$\frac{1-\delta}{2} \geq \left| \Pr[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^{\kappa}_{AT}, \mathcal{A}^t, b}(\kappa) = b] - \frac{1}{2} \right| \geq \frac{1}{c} \cdot \left(\frac{\varepsilon}{2} - \frac{1}{c^g} \right), \tag{6}$$

which implies that δ and ε cannot be simultaneously equal to $1 - \mathsf{negl}(\kappa)$ (since $1/(2c) - 1/c^{1+g}$ cannot be a negligible function for any polynomial c and any constant $g > 0$).

Limitations of the impossibility result. Even putting aside the heuristic security guarantee of our fine-grained construction (or its security in an idealized model), a gap remains between our impossibility result and our construction: our impossibility result does not rule out the possibility of having, say, a $(1 - \mathsf{negl}(\kappa), 1 - 1/c, c, \kappa)$-Anonymous Transfer – that is, an anonymous transfer with overwhelming correctness, and vanishing anonymity error $1/c$ in c rounds, with standard (superpolynomial) security. In contrast, our heuristic construction only achieves overwhelming correctness and anonymity arbitrarily close to $1/c$ against *fine-grained* adversaries. It is an interesting open question to close this gap. We conjecture that the true answer is negative:

Conjecture 1. There exists no $(1 - \mathsf{negl}(\kappa), 1 - 1/c, c, \kappa)$-Anonymous Transfer.

What follows assumes that the reader is familiar with standard philosophical considerations on the worlds of Impagliazzo. Proving the above conjecture would have a very interesting (theoretical) consequence: it would demonstrate the existence of a natural cryptographic primitive that plausibly exists within the realm of fine-grained cryptography, yet is impossible with standard hardness gap. It is known that fine-grained constructions sometimes allow building "high-end" cryptographic primitives in "low-end" cryptographic realms. For example, Merkle puzzles, which can be instantiated under exponentially strong one-way functions [BGI08], provide a fine-grained key exchange; borrowing Impagliazzo's terminology [Imp95], this places "fine-grained Cryptomania" inside (a strong form of) Minicrypt. Proving the conjecture would induce a comparable result, but at the highest level of the hierarchy: it would, in a sense, place fine-grained Impossibilitopia (a world of cryptographic primitives so powerful that they simply cannot exist) inside Obfustopia.

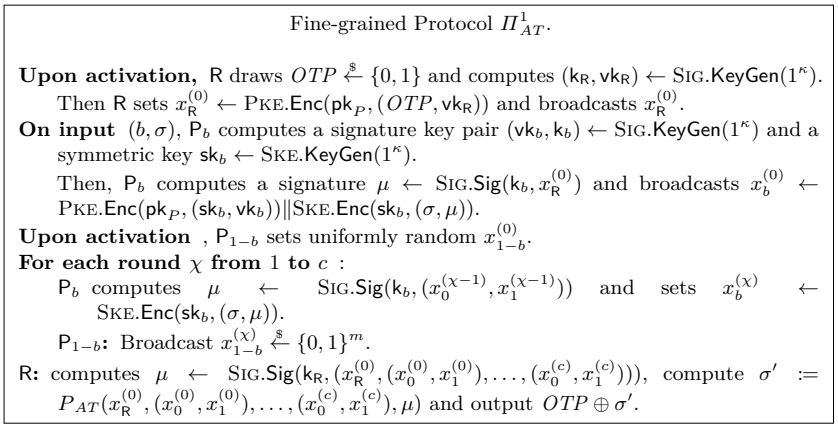

Fig. 4: The protocol Π^1_{AT} for fine-grained Anonymous Transfer. The circuit P_{AT} is defined in Fig. 5.

5 Fine-Grained AT from Ideal Obfuscation

In this section, we focus on realizing Anonymous Transfer with fine-grained security according to Definition 5. More precisely, we construct a c-round protocol which achieves anonymity δ, where the honest parties have runtime in $\mathfrak{C}_1 := \mathcal{O}(c)$ against adversaries in $\mathfrak{C}_2 := \mathrm{o}(c^2(1-\delta))$, where $c = c(\kappa)$ is a polynomial in κ. For the sake of simplicity we introduce the protocol with $N = 3$, implying a single dummy friend. However, expanding this protocol to an arbitrary $N \in \mathbb{N}$ is straightforward as the behavior of all dummy friends is the same by definition and instead of two messages, each round now contains $N - 1$ messages.

We exploit the limited runtime of the adversary and provide a protocol in Fig. 4 with c rounds. In each new round (or with each valid sender message) the probability that the correct bit is eventually returned increases, *i.e.*, each valid round increases the receiver's confidence in the message. Each round lets the sender compute a signature μ using a sEUF-CMA secure signature scheme[7] for the transcript of the previous round. The transferred bit σ and the signature μ are then sent. The verification key for the signature scheme is transmitted by the sender in the first round. In order to make the sent messages look random the message is not sent directly. Instead, the sender encrypts the message using an IND$-CCA secure encryption scheme[7], [Rog04]. Since not every length m bitstring is a valid ciphertext, we use a special function Dec^* instead of the normal function Dec, which is defined as follows: If Dec on input ct returns $\perp$ then Dec^* returns $\mathsf{F}(ct)$, otherwise Dec^* returns $\mathsf{Dec}(ct)$. Hence, every possible input allows an interpretation as a cleartext. We use those for both the asymmetric and symmetric schemes.

In order to make the output unusable for any other party, the receiver draws a One-Time-Pad as first message which eventually masks the final output, and

[7] See [ACM21] for definitions of sEUF-CMA, IND$-CCA and ideal obfuscation.

$P_{AT}[\mathsf{pk}_P, c]\left(x_{\mathsf{R}}^{(0)}, \left(x_0^{(0)}, x_1^{(0)}\right), \left(x_0^{(1)}, x_1^{(1)}\right) \ldots, \left(x_0^{(c)}, x_1^{(c)}\right)\right)$

$(OTP, \mathsf{vk}_{\mathsf{R}}) \coloneqq \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, x_{\mathsf{R}}^{(0)})$,

$(\mathsf{sk}_0, \mathsf{vk}_0) \coloneqq \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, x_0^{(0)}[1\colon m])$, $\quad(\sigma_0, \mu_0) \coloneqq \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_0, x_0^{(0)}[m+1\colon 2m])$,

$(\mathsf{sk}_1, \mathsf{vk}_1) \coloneqq \textsc{Pke}.\mathsf{Dec}^*(\mathsf{sk}_P, x_1^{(0)}[1\colon m])$, $\quad(\sigma_1, \mu_1) \coloneqq \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_1, x_1^{(0)}[m+1\colon 2m])$,

if $\neg\textsc{Sig}.\mathsf{Vfy}(\mathsf{vk}_{\mathsf{R}}, (x_{\mathsf{R}}^{(0)}, (x_0^{(0)}, x_1^{(0)}), \ldots, (x_0^{(c)}, x_1^{(c)})))$ **then** :

  return $\mathsf{CointossS}^{(\pi)}_{(0.5)}(0,1)$

$\chi_0 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_0, \mathsf{vk}_0, x_{\mathsf{R}}^{(0)})]\!] \cdot (c+1)$, $\quad\chi_1 \coloneqq [\![\textsc{Sig}.\mathsf{Vfy}(\mu_1, \mathsf{vk}_1, x_{\mathsf{R}}^{(1)})]\!] \cdot (c+1)$,

foreach $\chi \in \{1, \ldots, c\}$ **do**:

  foreach $b \in \{b' | b' \in \{0,1\}, \chi_b = (c+1)\}$ **do**: // Take on the role of each potential sender.

    $X_b \coloneqq \textsc{Ske}.\mathsf{Dec}^*(\mathsf{sk}_b, x_b^{(\chi)})$, $\quad\sigma'_b \coloneqq X_b[0]$, $\quad\mu_b \coloneqq X_b[1 : |X_b|]$

    if $\neg\textsc{Sig}.\mathsf{Vfy}(\mu_b, \mathsf{vk}_b, \pi[\chi - 1]) \vee \sigma_b \neq \sigma'_b$ **then** :

      $\chi_b \coloneqq \chi$ // Remember first bad round.

$b' \coloneqq \mathbf{argmax}_b(\chi_b)$

return $OTP \oplus \mathsf{CointossS}^{(\pi)}_{(1/2\cdot(1+\chi_{b'}/c))}(\sigma_{b'}, (1-\sigma_{b'}))$

Fig. 5: Obfuscated program P_{AT} for the fine-grained setting with c rounds.

a verification key of a signature scheme. The latter is used to ensure that the receiver *approves* with the transcript; after the two potential senders provided all messages, the receiver *signs* the entire transcript and only if this signature verifies the entire previous transcript, the circuit continues. The first message of the receiver is broadcast, while the signature is only used locally.

The receiver obtains its output by computing the signature as described above and feeding the final transcript alongside the signature into an obfuscated circuit which is supplied in a common reference string. The circuit is obfuscated using ideal obfuscation[7]. It hides a PRF key and a secret decryption key sk_P for the IND$-CCA secure PKE. The corresponding encryption key pk_P is also part of the CRS and, hence, known to all parties. This encryption scheme is used by the sender and the receiver to hide their respective *first* message. This uniquely determines the symmetric key used to decrypt the remaining messages of each potential sender. The message also contains a verification key used to sign the previous messages in future rounds, the bit that the sender wants to transfer, and the initial signature on the receivers message. The remaining rounds of the sender are encrypted using a *symmetric* scheme, namely the IND$-CCA secure SKE scheme, using the key transferred to the circuit in the first round.

The circuit is shown in Fig. 5. It starts by extracting the verification keys and symmetric encryption keys (one per potential sender) alongside the bits that the respective party wants to transfer and the initial signatures on the first receiver messages from the respective initial messages of both parties, and the receivers OTP and verification key from the receiver message. Then the circuit starts by verifying the signature of the receiver on the entire transcript, and if that does not

match, returns a uniformly random bit[8]. Otherwise, if the receiver's signature is valid, the circuit searches for the first *faulty* round of each potential sender. That is, the first round of each potential sender where the signature on the previous round fails to verify or where the encoded bit differs from the bit extracted from the initial message. The party who sent the most consecutive valid rounds is selected as the sending party. The circuit outputs the bit transmitted by that party with probability depending on the ratio between valid sender messages and the total number of rounds, which ranges between 1/2 (*i.e.*, a uniformly random bit) if no round was valid for any party and 1 (*i.e.*, deterministically returning the correct bit) if all rounds were correct. However, as stated before, the circuit does not output that bit directly, but instead *masks* it using the OTP extracted from the receiver's first message. This ensures secrecy[9], as other parties only get a masked output which information-theoretically hides the actual bit.

5.1 Security Analysis

Theorem 4 (Correctness). *If the protocol from Fig. 4 is instantiated with an Ideally Obfuscated version of the circuit from Fig. 5 the protocol is ε-correct with $\varepsilon = (1 - \mathsf{negl}(\kappa))$.*

At the end of an honest protocol execution, the maximum round in which a valid signature has been provided equals the number of rounds c. With overwhelming probability, the sending parties' input is the only one that contains c many valid rounds. Hence, the correctly masked bit is returned. Since the mask is input by the receiver and later applied to the output, the receiver obtains the correctly masked bit. We refer the reader to [ACM21] for a formal proof.

Theorem 5 (Anonymity). *Let P_{KE} be an IND$-CCA secure asymmetric encryption scheme, let S_{KE} be a tightly secure multi-challenge IND$-CCA secure symmetric encryption scheme, let S_{IG} be an sEUF-CMA secure signature scheme, let $\mathbb{O}$ be an ideal obfuscator, let F be a secure PRF, and let κ be the security parameter. Then the c-round protocol Π^1_{AT} for $N = 3$ satisfies δ-anonymity for all adversaries in $\mathfrak{C}_2 := \mathrm{o}(c^2(1-\delta))$.*

Proof (sketch). An outline of the entire proof is given in the full version [ACM21]. On a high level, the proof is structured into two parts. In the first part, we successively modify the anonymity game $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^1_{AT},\mathcal{A},b}(\kappa)$ and the obfuscated circuit oracle P_{AT} to remove as much computationally hidden information about

[8] This is denoted in the figure by the $\mathsf{CointossS}^{(\pi)}_{(p)}(\sigma,\overline{\sigma})$ function, which returns σ, *i.e.* the first argument, with probability p, and $\overline{\sigma}$, *i.e.* the second argument, with the complementary probability $(1-p)$, where the randomness for p is extracted from the argument provided by π.

[9] Secrecy is an additional property we require for *Strong AT*. Secrecy means that no third party can extract the transferred bit from the transcript (see the full version [ACM21] for the formal definition). This property will be relevant for applications that use AT as a building block.

Oracle O_i^β

if $\beta = 0$ **then**

$p_i \coloneqq \dfrac{i + c - 1}{2c}$

else

$p_i \coloneqq \dfrac{i + c}{2c}$

return $\mathsf{Ber}(p_i)$

$\mathcal{C}(c)$

$\beta \xleftarrow{\$} \{0, 1\}$

return $\mathcal{A}^{\mathsf{O}_1^\beta, \ldots, \mathsf{O}_c^\beta}(1^\kappa)$

$\mathcal{A}(1^\kappa)$

for $j = 1 \ldots t$ **do**

$i_j \leftarrow \mathit{Computations}$

$x_j \xleftarrow{\$} \mathsf{O}_{i_j}$

$\beta' \leftarrow \mathit{Computations}((i_j, x_j)_{j=1}^t)$

return β'

Fig. 6: Game to distinguish whether Bernoulli oracles follow a given distribution p or $q = p - 1/2c$.

b as possible. More precisely, we exploit the non-malleability of PKE and sEUF-CMA security of Sig to unnoticeably alter the oracle to determine the number of valid rounds by counting how many rounds of the input transcript are identical to the challenge transcript provided by $\mathrm{Exp}^{\mathsf{anon}}_{\Pi^1_{AT},\mathcal{A},b}(\kappa)$. The first round which is not entirely identical to the challenge transcript (i.e. either the sender message or the non-sender message differ) increases the valid rounds count only if the input sender message is identical to the challenge sender message or if the input sender message decrypts to the same content as the challenge sender message. The following round will be counted as invalid since the signature verification will fail. After this step, the decryption keys of SKE and PKE are not necessary for chosen-ciphertext simulation anymore. Then, we first replace the sender messages which are encrypted using SKE and then the first round sender message which is encrypted using PKE with uniform randomness exploiting the IND$-CCA security of both encryption schemes.

The only information about the bit b that is left in the present game is due to the oracle which counts valid sender messages by comparing the input sender message with the challenge sender message. Clearly, the final modification of the game must be the removal of this dependency on b. However, this removal will noticeably alter the output distribution of the oracle. Hence, an adversary with arbitrary polynomial runtime will be able to distinguish this hop with constant probability [CDV+14]. However, if we can limit the runtime of the adversary to be sub-quadratic in the runtime of the honest protocol execution, we are able to apply results from distribution testing to achieve a good bound for this distinguishing advantage. We will elaborate on this final game hop in more detail below and will refer to the second last game as $\mathrm{Game}_7^\sigma(\kappa)$ and to the last game (i.e. the game, where no information about b remains) as $\mathrm{Game}_8^\sigma(\kappa)$. For detailed descriptions of all game hops, we refer the reader to the full version [ACM21].

For the sake of reducing complexity of the problem of proving indistinguishability between $\mathrm{Game}_7^\sigma(\kappa)$ and $\mathrm{Game}_8^\sigma(\kappa)$ we describe an intermediate game in Fig. 6 that is provably as hard to solve as distinguishing the two games.

The key idea is the following: The challenger $\mathcal{C}$ creates c oracles where the probability to return 1 is equally distributed between $1/2$ and 1 in c steps.

On $\beta = 0$ the oracles are distributed equally between $[1/2, 1)$. On $\beta = 1$ the oracles are distributed equally between $(1/2, 1]$. That is, on $\beta = 0$ the oracle χ returns 1 with probability $(c + \chi - 1)/(2c)$ and on $\beta = 1$ it returns 1 with probability $(c + \chi)/(2c)$.

We now stress that this game is *as hard* as the problem of distinguishing the two games from $\mathrm{Game}_7^\sigma(\kappa)$ and $\mathrm{Game}_8^\sigma(\kappa)$:

Lemma 6. *Let $\mathcal{D}$ be a distinguisher distinguishing $\mathrm{Game}_7^\sigma(\kappa)$ and $\mathrm{Game}_8^\sigma(\kappa)$ with advantage α over guessing. Let t be the number of queries that $\mathcal{D}$ sends to the obfuscated circuit. There is a reduction adversary $\mathcal{A}$ that uses $\mathcal{D}$ which has advantage α over guessing in winning Fig. 6.*

Proof (sketch). To create the transcript, the adversary samples bits σ and b for the transferred bit and the sending party, respectively. It then creates $\pi_\mathcal{C}$ by sampling $2c$ random bitstrings of length m and assigns them to the two parties.

The oracle is simulated by letting $\mathcal{A}$ follow the behavor of the oracle: If the input is the challenge transcript, output the bit directly; if it is a completely new transcript, follow the honest protocol; otherwise, if the first-round messages of both parties are the same, $\mathcal{A}$ searches χ^* as the first round where the input differs from the challenge transcript.

If the message from the sending party in round $(\chi^* + 1)$ is from the challenge trascript, then $\mathcal{A}$ sends χ^* to the oracle O_{χ^*} provided by the challenger and obtains a bit σ^*, and returns $\sigma^* \oplus \bar{\sigma}$ (*i.e.* the return gets flipped if it should go towards 0). Otherwise, $\mathcal{A}$ returns σ with probability proportional to χ^*/c.

It follows (we elaborate on that in [ACM21]) that the result can be translated; if the distinguisher guesses $\mathrm{Game}_8^\sigma(\kappa)$ then $\mathcal{A}$ reports that the χ-th oracle returns 1 with probability $(\chi + c - 1)/2c$. Otherwise, if the distinguisher guesses $\mathrm{Game}_7^\sigma(\kappa)$, $\mathcal{A}$ reports that the probabilities were given as $(\chi + c)/2$.

The simulation is such that the challenge oracles are only queried if the input transcript contains the first χ^* messages of the challenge transcript from both parties and then in round $\chi^* + 1$ only the message of the sending party. In that case, the difference induced by the game hop states that in $\mathrm{Game}_7^\sigma(\kappa)$ the sending parties message still increases the probability by $1/(2c)$, whereas in $\mathrm{Game}_8^\sigma(\kappa)$ the message is ignored; which correspond exactly to the case we have to distinguish in our challenge. The full proof can be found in the full version [ACM21]. □

Proving indistinguishability has thus been reduced to showing that no fine-grained adversary can win the game from Fig. 6 with non-negligible advantage. The interface of an adversary in this game is given as a set of $2c$ *oracles*. Each oracle follows a *Bernoulli* distribution that returns the correct bit $\sigma_\mathcal{C}$ with probability p. For each round $\chi < c$ any distinguisher $\mathcal{D}$ is given access to two oracles. Each oracle can be queried by copying the first χ messages of *both* parties, but then using (exactly) one new message for round $(\chi + 1)$—which replaces either the sending parties message or that of the dummy friend. Any upper bound on winning the game from Fig. 6 translates to the underlying problem of distinguishing the final two games.

Analyzing the game from Fig. 6 comes down to probability theory. Recall from Corollary 1 that in order to distinguish two Bernoulli distributions p and q with advantage $\alpha/2$ we require $\Omega(\alpha/\mathsf{d}_{\mathsf{TV}}(p,q))$ many samples. Applying this corollary to Fig. 6 implies that we have c instances where the χ-th instance is to distinguish $p = \frac{\chi+c}{2c}$ from $q = \frac{\chi+c-1}{2c}$. This implies the following L_1-norm between p and q in round χ:

$$\begin{aligned}\mathsf{d}_{\mathsf{TV}}(p,q) &= \frac{1}{2}(|\Pr[p=1]-\Pr[q=1]| + |\Pr[p=0]-\Pr[q=0]|) \\ &= \frac{1}{2}\left(\left|\frac{c+\chi}{2c} - \frac{c+\chi-1}{2c}\right| + \left|\frac{c-\chi}{2c} - \frac{c-\chi+1}{2c}\right|\right) = \frac{1}{2c}\end{aligned} \tag{7}$$

Note here that the total variational distance in round χ is *independent* from the round χ and the same for all c oracles. Combining this information with Lemma 2 means that *any* distribution p and q resulting from sampling t times from *arbitrary* oracles results in a total variational distance $\leq t\frac{1}{2c}$. [10]

We now merge this insight with the result of Eq. (7) and the bound of Corollary 1. This leads a lower bound of:

$$t \in \Omega\left(\frac{\alpha}{\mathsf{d}_{\mathsf{TV}}(p,q)}\right) = \Omega(\alpha c) \tag{8}$$

We thus have:

Corollary 3. *Let $\mathcal{D}$ be a distinguisher in Fig. 6 that uses t samples and has runtime in $\mathfrak{C}_2 := \mathrm{o}(c^2/\alpha)$. Let the cost of acquiring a single sample be $\mathcal{O}(c)$. Then the distinguisher $\mathcal{D}$ is correct with probability at most $1/2 + \alpha/2$.*

Proof. The bound from Eq. (8) covers any adversary trying to win Fig. 6 regardless of how the t samples are distributed between the c oracles. This follows from the subadditional property of the total variational distance shown in Lemma 2 and the computation in Eq. (7) showing that the total variational distance is the same between all oracles; thus the bound from Lemma 1 still is valid and the total variational distance between any pair of t-fold distributions is at most $t \cdot \frac{1}{2c}$.

Thus Lemma 3 maintains its validity. Hence the lower bound of Eq. (8) matches our setting. The bound is linear in c with the linear cost of querying a single sample (as the adversary has to evaluate the entire circuit for each sample, which requires $\mathcal{O}(c)$ runtime) this limits the distinguisher in such a way that only strictly less samples can be drawn than required according to Eq. (8). □

Putting everything together, we have that for all PPT distinguishers D, $|\Pr[\mathsf{out}_{0,\mathsf{D}} = 1] - \Pr[\mathsf{out}_{8,\mathsf{D}} = 1]|$ is negligible in κ. In particular, $|\Pr[\mathsf{out}_{0,\mathsf{D}} = 1] - \Pr[\mathsf{out}_{8,\mathsf{D}} = 1]|$ is negligible for distinguishers D in $\mathfrak{C}_2$. Additionally, the

[10] This is in contrast to the Hellinger-distance H which yields tighter bounds but where the amount of information from a single query really depends on the oracle O_χ which is queried. This makes it harder to provide meaningful bounds for adversaries querying different oracles with their t samples.

employed reductions are in $\mathfrak{C}_1 = \mathcal{O}(c)$. Furthermore, for all adversaries $\mathcal{A}$, $|\Pr[\mathsf{out}_{8,\mathcal{A}} = 1|b = 0] - \Pr[\mathsf{out}_{8,\mathcal{A}} = 1|b = 1]| \leq \alpha$, where the runtime of the game also is in $\mathfrak{C}_1$. Hence, we may conclude that for all adversaries $\mathcal{A}$ in $\mathfrak{C}_2$, $|\Pr_{b \xleftarrow{\$} \{0,1\}}[\mathrm{Exp}^{\mathsf{anon}}_{\Pi^1_{AT},\mathcal{A},b}(\kappa) = b] - 1/2| \leq \alpha/2$. □

On the Need for Stronger Obfuscation. Due to [CLT+15], indistinguishability obfuscation (or more precisely, its probabilistic variant) can only guarantee indistinguishability if the distance between the output distributions of two circuits is statistically close to zero. This is not the case in our final game hop. Therefore, we crucially require a stronger form of obfuscation such as virtual black-box obfuscation or ideal obfuscation. Due to [JLL+22], employing ideal obfuscation yields a heuristic candidate proven secure in an idealized model. Hence, our result constitutes a first step towards instantiating anonymous transfer.

Stronger Anonymity Notions. Our positive result demonstrates that despite our strong negative result, *some* non-trivial anonymity is achievable. Note, however, that our positive result is still weak in many regards. Strengthening the achieved notion to, for instance, achieve anonymity against malicious non-participants, seems highly non-trivial. In particular, malicious non-participants may easily nullify any correctness guarantee by behaving exactly like a sender. Straightforward attempts to address this problem, e.g. letting the obfuscated circuit output all messages with equal confidence, open the gates for new attacks. For instance, in the above setup, replacing the last message of half of all possible senders causes the circuit to output either both the sender message and the injected message or only the injected message, depending on whether the real sender is part of the parties whose messages are replaced. This strategy allows to de-anonymize the sender in runtime $\mathcal{O}(c \log c)$.

5.2 Final Result

Let $c = c(\kappa)$ be a polynomial in κ. Let $\mathfrak{C}_1 := \mathcal{O}(c)$ and let $\mathfrak{C}_2 := \mathrm{o}(c^2(1-\delta))$ for some $\delta \in \mathbb{R}_{[0,1]}$. Putting Theorems 4 and 5 together, we have:

Corollary 4. *The protocol Π^1_{AT} is a strong $\mathfrak{C}_1$-fine-grained $(1 - \mathsf{negl}(\kappa), \delta, c, 1)$-AT against $\mathfrak{C}_2$.*

Applying Lemma 5 to transform our single-bit AT into an ℓ-bit AT yields:

Corollary 5. *The protocol Π^ℓ_{AT} is a strong $\mathfrak{C}'_1$-fine-grained $(1 - \mathsf{negl}(\kappa), (\delta\ell - \ell - \delta + 2), c \cdot \ell, \ell)$-AT against $\mathfrak{C}'_2$, where $\mathfrak{C}'_1 = \ell \cdot \mathfrak{C}_1$ and $\mathfrak{C}'_2 = \mathfrak{C}_2 - \ell \cdot \mathfrak{C}_1$.*

Using $\delta = 1 - \frac{1}{\sqrt{c}}$ and $c = \Omega(\ell^2)$ for the single-bit AT Π^1_{AT} we get that $\delta' := 1 - \frac{\ell-1}{c}$ and $\mathfrak{C}'_1 = \mathcal{O}(\ell \cdot c)$ and $\mathfrak{C}'_2 = \mathrm{o}(c^2(1-\delta) - \ell \cdot c) = \mathrm{o}(c^2(1-\delta)) = \mathrm{o}(c^{1.5})$.

A non-black-box change to the protocol Π^1_{AT} from Figs. 4 and 5 leads to better overall parameters. We introduce the necessary changes to the protocol alongside a security analysis in the full version [ACM21].

References

[ACM21] T. Agrikola, G. Couteau, and S. Maier. Anonymous whistleblowing over authenticated channels. Cryptology ePrint Archive, Report 2021/1341, 2021. https://eprint.iacr.org/2021/1341.

[APY20] I. Abraham, B. Pinkas, and A. Yanai. Blinder - scalable, robust anonymous committed broadcast. In J. Ligatti, X. Ou, J. Katz, and G. Vigna, editors, *ACM CCS 2020*, pages 1233–1252. ACM Press, November 2020.

[BEA14] B. Burrough, E Ellison, and S. Andrews. The snowden saga: a shadowland of secrets and light. *Vanity Fair*, 23, 2014.

[Ber16] C. Berret. Guide to securedrop, 2016.

[BGI08] E. Biham, Y. J. Goren, and Y. Ishai. Basing weak public-key cryptography on strong one-way functions. In R. Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 55–72. Springer, Heidelberg, March 2008.

[CBM15] H. Corrigan-Gibbs, D. Boneh, and D. Mazières. Riposte: an anonymous messaging system handling millions of users. In *2015 IEEE Symposium on Security and Privacy*, pages 321–338. IEEE Computer Society Press, May 2015.

[CDV+14] S. Chan, I. Diakonikolas, P. Valiant, and G. Valiant. Optimal algorithms for testing closeness of discrete distributions. In *25th SODA*, pages 1193–1203, 2014.

[CGCD+20] K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt, and D. Stebila. A formal security analysis of the signal messaging protocol. *Journal of Cryptology*, 33(4):1914–1983, 2020.

[CGO+07] N. Chandran, V. Goyal, R. Ostrovsky, and A. Sahai. Covert multiparty computation. In *48th FOCS*, pages 238–248. IEEE Computer Society Press, October 2007.

[Cha03] D. Chaum. Untraceable electronic mail, return addresses and digital pseudonyms. In D. Gritzalis, editor, *Secure Electronic Voting*. Volume 7, Advances in Information Security, pages 211–219. Springer, 2003.

[Cha88] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, January 1988.

[CLT+15] R. Canetti, H. Lin, S. Tessaro, and V. Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 468–497. Springer, Heidelberg, March 2015.

[DMS04] R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: the second-generation onion router. In M. Blaze, editor, *USENIX Security 2004*, pages 303–320. USENIX Association, August 2004.

[DVV16] A. Degwekar, V. Vaikuntanathan, and P. N. Vasudevan. Fine-grained cryptography. In M. Robshaw and J. Katz, editors, *CRYPTO 2016,*

Part III, volume 9816 of *LNCS*, pages 533–562. Springer, Heidelberg, August 2016.

[ECZ+21] S. Eskandarian, H. Corrigan-Gibbs, M. Zaharia, and D. Boneh. Express: lowering the cost of metadata-hiding communication with cryptographic privacy. In M. Bailey and R. Greenstadt, editors, *USENIX Security 2021*, pages 1775–1792. USENIX Association, August 2021.

[HLv02] N. J. Hopper, J. Langford, and L. von Ahn. Provably secure steganography. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 77–92. Springer, Heidelberg, August 2002.

[Imp95] R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*, pages 134–147. IEEE, 1995.

[JLL+22] A. Jain, H. Lin, J. Luo, and D. Wichs. The pseudorandom oracle model and ideal obfuscation. Cryptology ePrint Archive, Report 2022/1204, 2022. https://eprint.iacr.org/2022/1204.

[Mer78] R. C. Merkle. Secure communications over insecure channels. *Commun. ACM*, 21(4):294–299, 1978.

[NSSD21] Z. Newman, S. Servan-Schreiber, and S. Devadas. Spectrum: high-bandwidth anonymous broadcast with malicious security. Cryptology ePrint Archive, Report 2021/325, 2021. https://eprint.iacr.org/2021/325.

[Phi18] D. Philipps. Reality winner, former nsa translator, gets more than 5 years in leak of russian hacking report. *New York Times*, 23, 2018.

[Rog04] P. Rogaway. Nonce-based symmetric encryption. In B. K. Roy and W. Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 348–359. Springer, Heidelberg, February 2004.

[vH04] L. von Ahn and N. J. Hopper. Public-key steganography. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 323–341. Springer, Heidelberg, May 2004.

[vHL05] L. von Ahn, N. J. Hopper, and J. Langford. Covert two-party computation. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 513–522. ACM Press, May 2005.

[Whi] Whistleblowing. https://legal-dictionary.thefreedictionary.com/Whistleblowing. Accessed: 2021-09-29 from West's Encyclopedia of American Law, edition 2. (2008).