# How to Slash Misbehavior in Threshold Decryption

Dan Boneh

Follow

--

Listen

Share

Dan Boneh, Aditi Partap, and Lior Rotem

Current threshold decryption systems are insecure when participants are rational actors. As we explain below, decryption parties can be bribed into selling their secret keys with no accountability

. As such, decryption parties, who are meant to safeguard a threshold decryption system, can collect a risk-free reward by selling their decryption keys to the highest bidder, or to all bidders. This defeats the purpose of encryption in the first place.

In this post we propose a way to fix this problem. The full details can be found in oupaper on this topic. But let's start at the beginning. What is threshold decryption?

In a threshold decryption system there is a public key that anyone can use to encrypt data. The corresponding secret decryption key is secret shared among n

trusted parties, so that any quorum of t

trusted parties can decrypt a given ciphertext. More precisely, decryption works as follows:

1. the requestor sends the ciphertext to all n

trusted parties,

1. the parties that agree to decrypt, apply their secret key to obtain a decryption share, which they send back to the requestor, and

2. the requestor can combine any t

valid decryption shares to obtain the plaintext.

The point is that an attacker who compromises fewer than t

decryption parties, learns nothing about the decryption of a target ciphertext. To learn more about threshold decryption, please see the graduate applied cryptography book (chapter 22).

Threshold decryption is used in a wide array of applications. Some blockchains use it to implement anencrypted mempool, where transactions are encrypted until a block is finalized. This is intended to reduce front running and other MEV issues. Threshold decryption has also been used in private voting systems and in sealed-bid auctions. In all these applications the reason the data is encrypted in the first place is to ensure that it is not available in the clear before a certain time. For example, in a sealed-bid auction, encryption is used to ensure that bids remain hidden until the bidding window is closed. Sharing the secret key among n

trusted parties ensures that even if t

-1 parties are dishonest, and decrypt the data early, the data remains hidden until the designated time. At the designated time, any t

honest parties can decrypt the data.

T

he problem

is that a

searcher who wants to gain an advantage by decrypting ciphertexts early could offer a bribe to some t

trusted parties to sell their decryption keys. Once the searcher obtains t

key shares, it can decrypt any ciphertext of its choice in the comfort of its home. From the point of view of the decryption parties, this action simply provides additional revenue.

The standard way to discourage this type of misbehavior is to introduce a penalty (slashing) mechanism. For example, if a whistleblower posts the decryption key of one trusted party on chain, that party would get slashed. The hope is that the risk of slashing would deter trusted parties from selling their key shares.

Unfortunately, this does not work

. The problem is that a clever quorum of decryption parties would not sell their keys to the searcher as is. Instead, they could evade slashing by using a property of existing threshold decryption systems. Namely, t

parties can work together to construct a master key that cannot be traced to any single party. Consequently, t

parties, who are willing to be bribed, can sell this master key to the searcher without any fear of slashing. The information they sell to the searcher reveals nothing about their identity. This provides a risk-free profit for the parties.

W

hat to do?

What we need is a threshold decryption system that no matter how a quorum of parties combine their secret keys, there is always a way to trace the resulting data to at least one of them, or perhaps to all of them. We model this by treating the data that the parties sell to the searcher as a decryption algorithm

which we denote by D

(·

). When this algorithm is given a well-formed ciphertext c

as input, it outputs the decryption of c

. This algorithm D

(·

) is often called a decoder. The strategy described above, where the parties sell a master key to the searcher, is a special case of this general paradigm; the decoder they sell operates by simply applying the master secret to the given ciphertext c

. Such a decoder cannot be traced to any party that created it. We stress that a quorum of sophisticated parties can choose to use a more clever strategy to build the decoder D

. In fact, they might try to obfuscate the implementation of D

to make it as hard as possible to understand how D

works.

In our work, we design new threshold decryption systems that are equipped with a cryptographic tracing

algorithm with the following property. Suppose t

or more parties come together and produce a working stateless decoder D

. Then our tracing algorithm interacts with D

as a black box, and outputs at least one member of the group that created D

. The tracing algorithm works by feeding malformed ciphertexts to the decoder D

. The decoder will successfully decrypt some ciphertexts and will fail to decrypt others. The pattern of successes and failures enables our tracing algorithm to provably identify at least one party in the group that created D

.

Our starting point is a cryptographic technique called traitor tracing. Here n

parties each hold a secret decryption key, and each party can decrypt a well-formed ciphertext on its own

. Think of DVD players, where every DVD player should be able to decrypt an encrypted DVD disk on its own. However, if a subset of parties J ⊆

[n

] collude to create a pirate decoder D

(·

) that can decrypt well-formed ciphertexts, then it is possible to trace D

to at least one member of J

using only blackbox access to the decoder D

. Traitor tracing received much attention over the years and multiple schemes have been developed.

In our work we develop the concept of traitor tracing for threshold decryption

, where now a subset of t

or more parties must work together to create a pirate decoder D

(·

). Our motivation is to secure real-world deployments of threshold decryption where there is a strong economic incentive to bribe the trusted parties into selling a working decoder D

(·

). We present a number of constructions for traitor tracing for threshold decryption where both ciphertexts and public keys are short, so that posting this data on chain is inexpensive. In particular, ciphertexts in our system are just four group elements, independent of n

and t

. We leave the details as well as the many directions for future work to the full paper.

We hope that this approach will help deter misbehavior in threshold decryption systems, and will help secure them in real world applications.