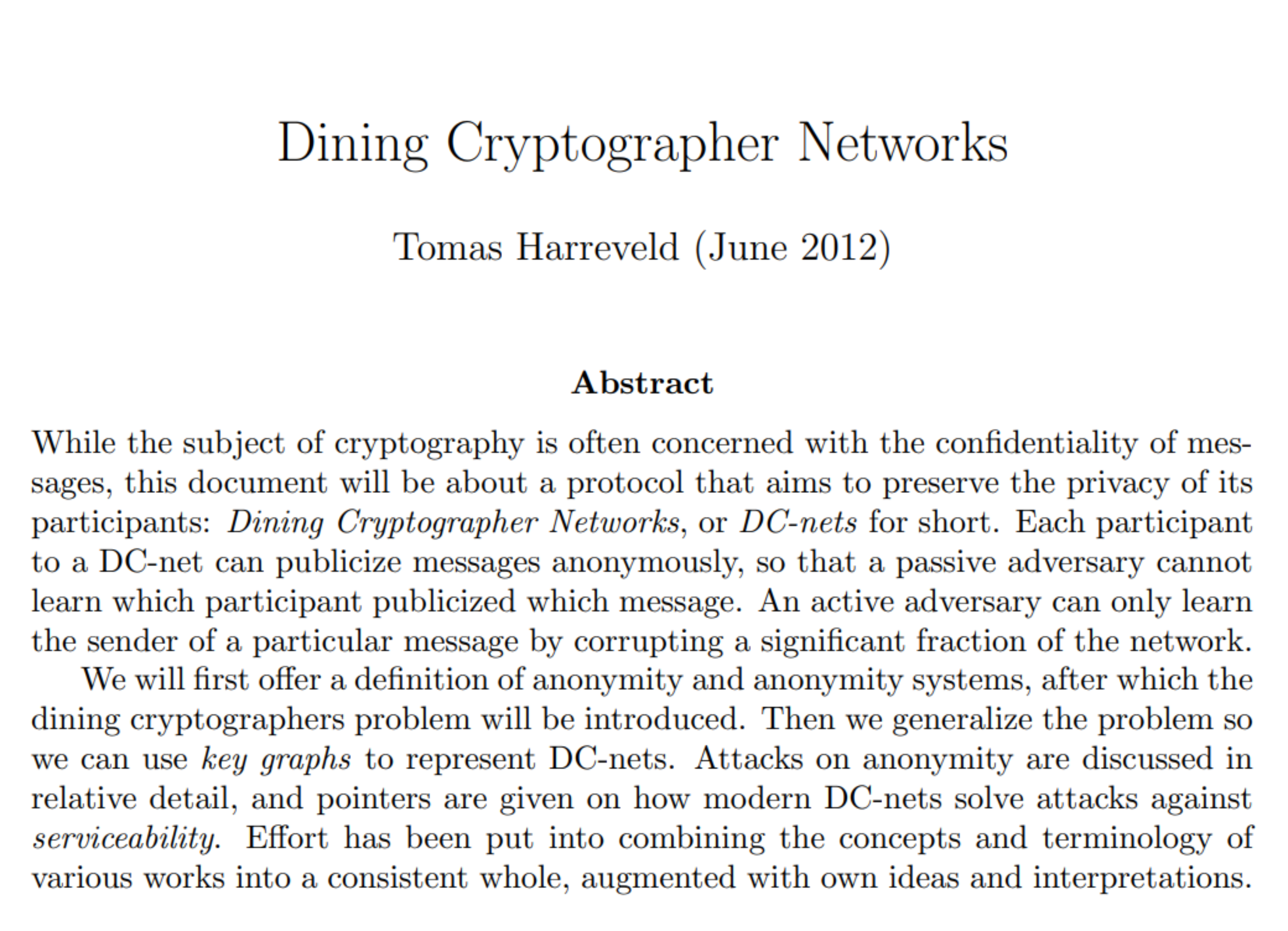
Dining Cryptographer Networks

UNCONDITIONAL SENDER AND RECIPIENT UNTRACEABILITY

Harreveld (2012)



Dining Cryptographer Networks

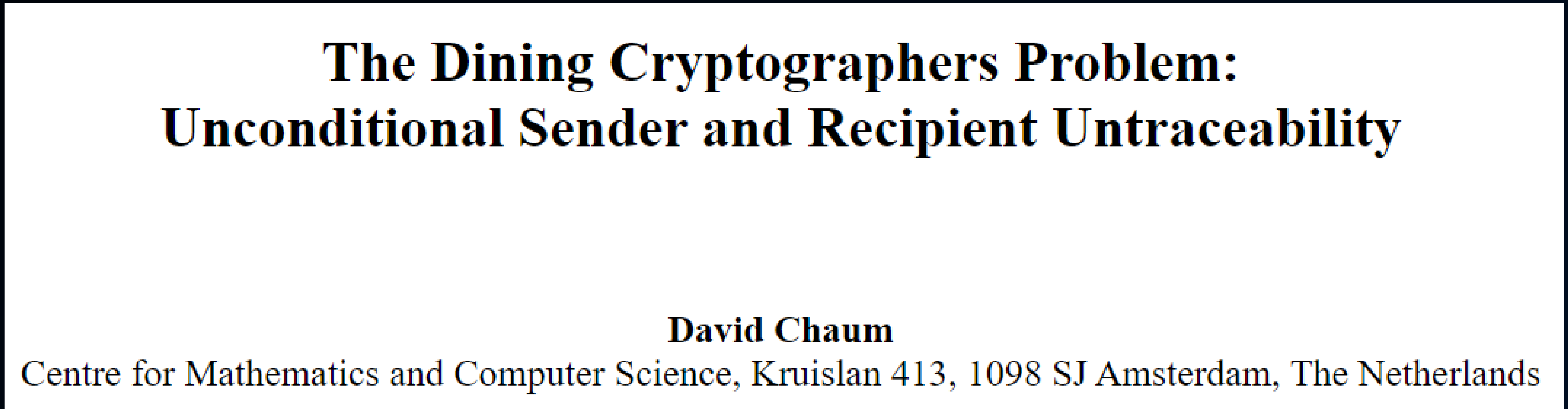
Tomas Harreveld (June 2012)

Abstract

While the subject of cryptography is often concerned with the confidentiality of messages, this document will be about a protocol that aims to preserve the privacy of its participants: *Dining Cryptographer Networks*, or *DC-nets* for short. Each participant to a DC-net can publicize messages anonymously, so that a passive adversary cannot learn which participant publicized which message. An active adversary can only learn the sender of a particular message by corrupting a significant fraction of the network.

We will first offer a definition of anonymity and anonymity systems, after which the dining cryptographers problem will be introduced. Then we generalize the problem so we can use *key graphs* to represent DC-nets. Attacks on anonymity are discussed in relative detail, and pointers are given on how modern DC-nets solve attacks against *serviceability*. Effort has been put into combining the concepts and terminology of various works into a consistent whole, augmented with own ideas and interpretations.

Chaum (1988)



**The Dining Cryptographers Problem:
Unconditional Sender and Recipient Untraceability**

David Chaum
Centre for Mathematics and Computer Science, Kruislan 413, 1098 SJ Amsterdam, The Netherlands

http://homepages.herts.ac.uk/~comqjs1/Dining.pdf

http://www.cs.cornell.edu/people/egs/herbivore/dcnets.html

Last Week – CoinShuffle

- One issue with CoinJoin implementations is the reliance on a central coordinator. Removing the coordinator would require a secure method of participants declaring their anonymous addresses
- We can replace the coordinator with a Coin*Shuffle,* where each participant **onion-encrypts their address** with the public keys of the latter participants. They then **decrypt and shuffle** all encrypted addresses they have received with their own address, and proceed to hand off the encrypted addresses to the next participant.
- Scales poorly with many participants, ElectronCash(5)

Wasabi Research Club

- January 6th, 2020 – Knapsack CoinJoin
- January 13th, 2020 – SNICKER
- January 20th, 2020 – CoinShuffle
- January 27th, 2020 – Dining Cryptographer Networks
- February 3rd, 2020 - TBD

https://github.com/zkSNACKs/WasabiResearchClub

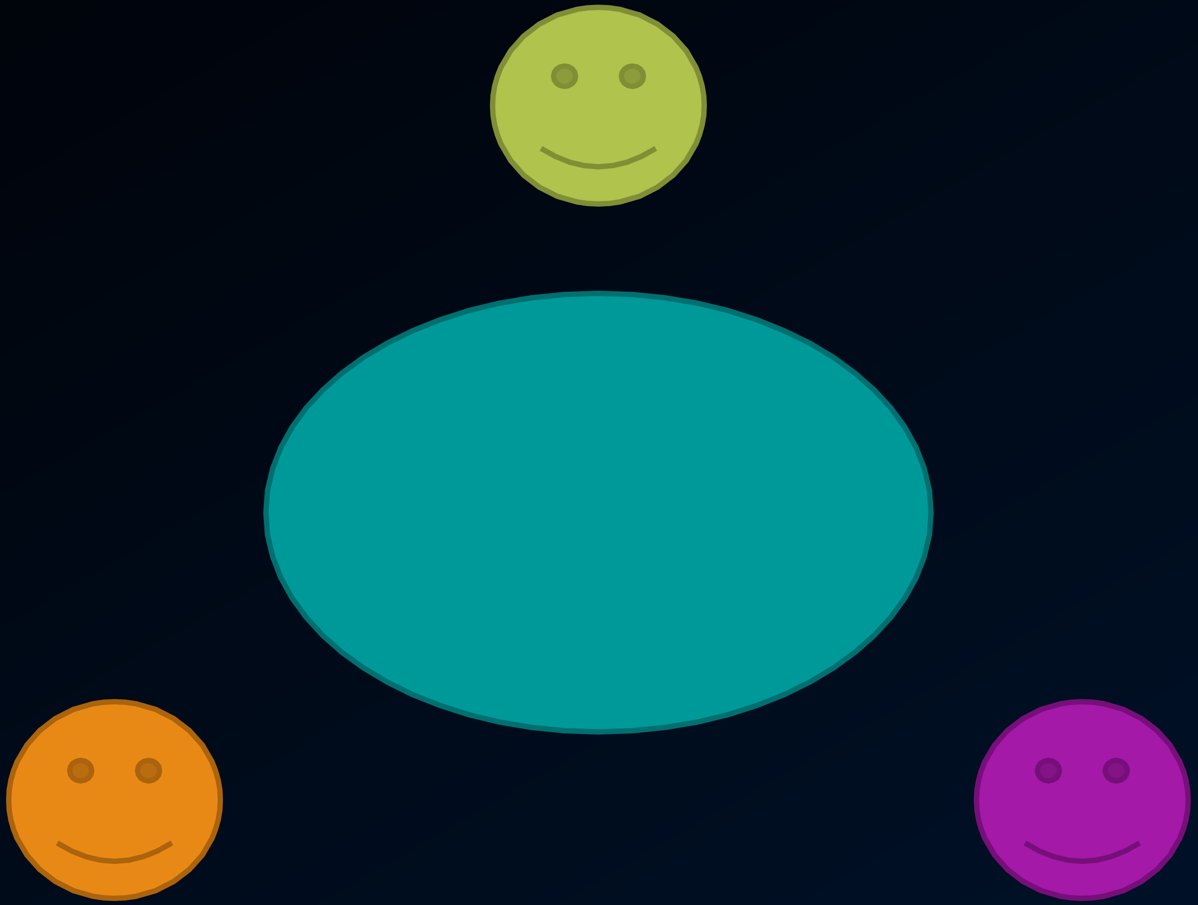
The Premise – The Cryptographers at Dinner

- *Three cryptographers are sitting down to dinner at their favorite three-star restaurant. Their waiter informs them that arrangements have been made with the maitre d'hotel for* ***the bill to be paid anonymously. One of the cryptographers might be paying for the dinner, or it might have been NSA*** *(U.S. National Security Agency). The three cryptographers respect each other's right to make an anonymous payment, but they wonder if NSA is paying. They resolve their uncertainty fairly by carrying out the following protocol-*

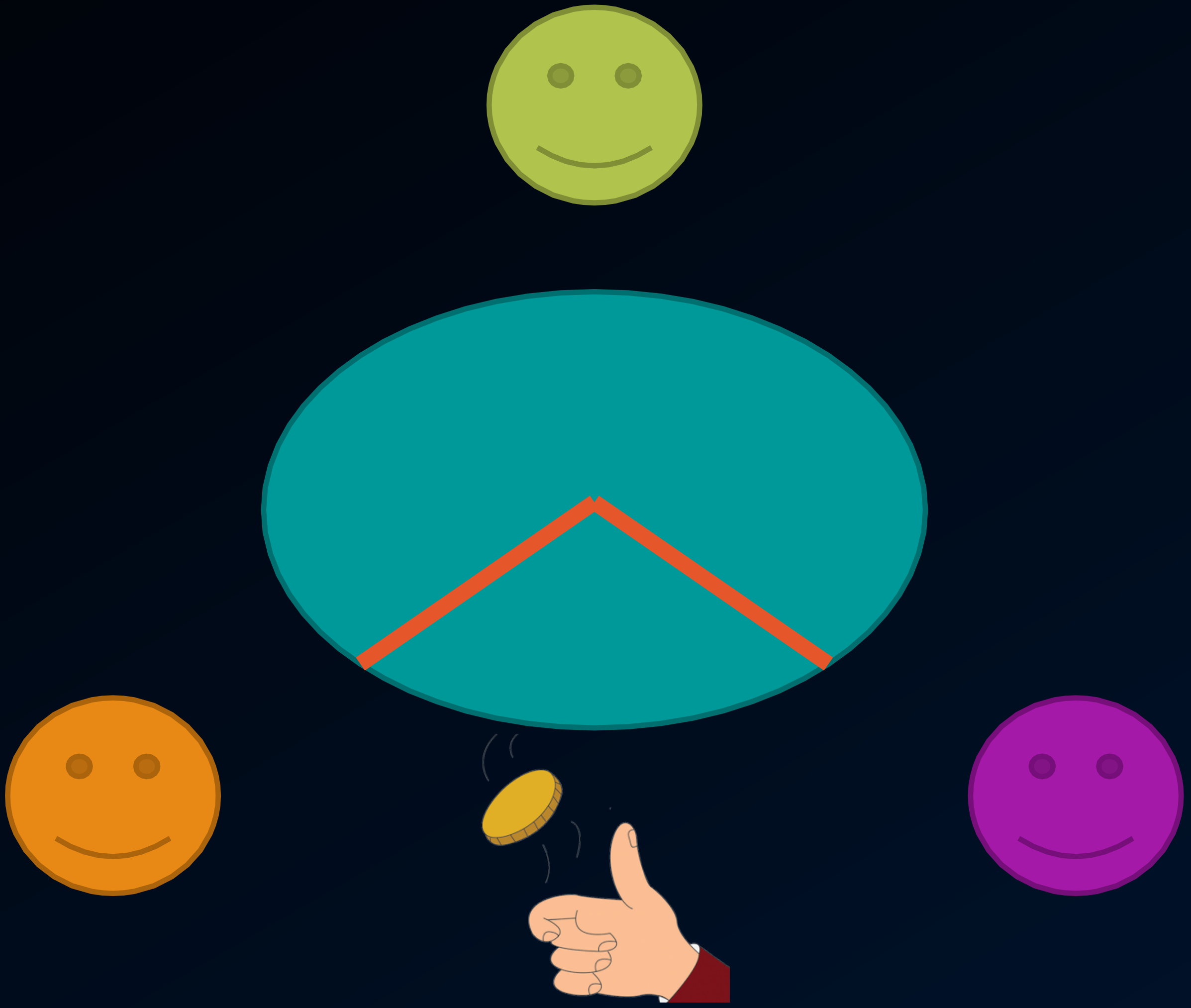
The Premise – The Cryptographers at Dinner

- Participants are trustworthy
- Participants want to be able to tell if someone in the group paid
- Participants don’t want to ‘out’ the payer, because he/she deserves the right to be able to pay anonymously

The Premise – The Cryptographers at Dinner



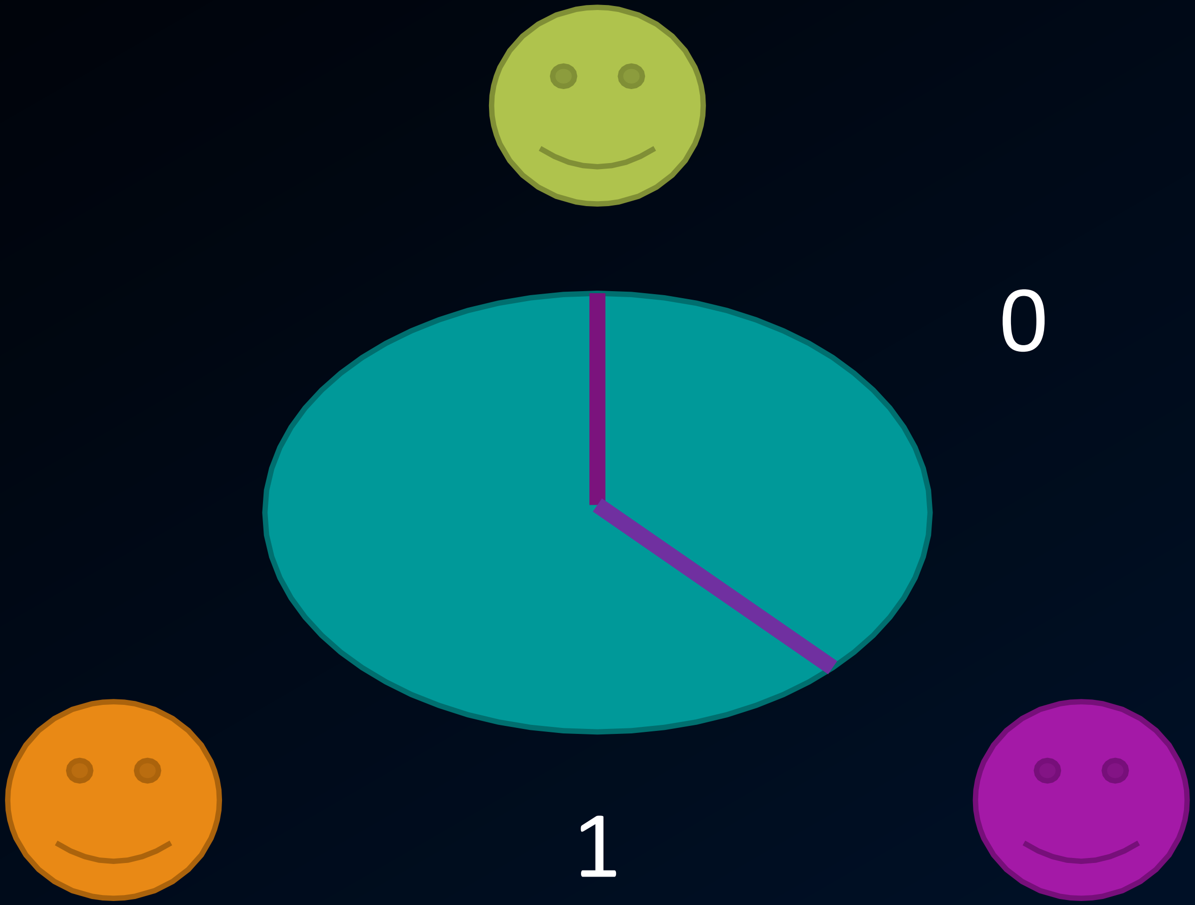
The Premise – The Cryptographers at Dinner



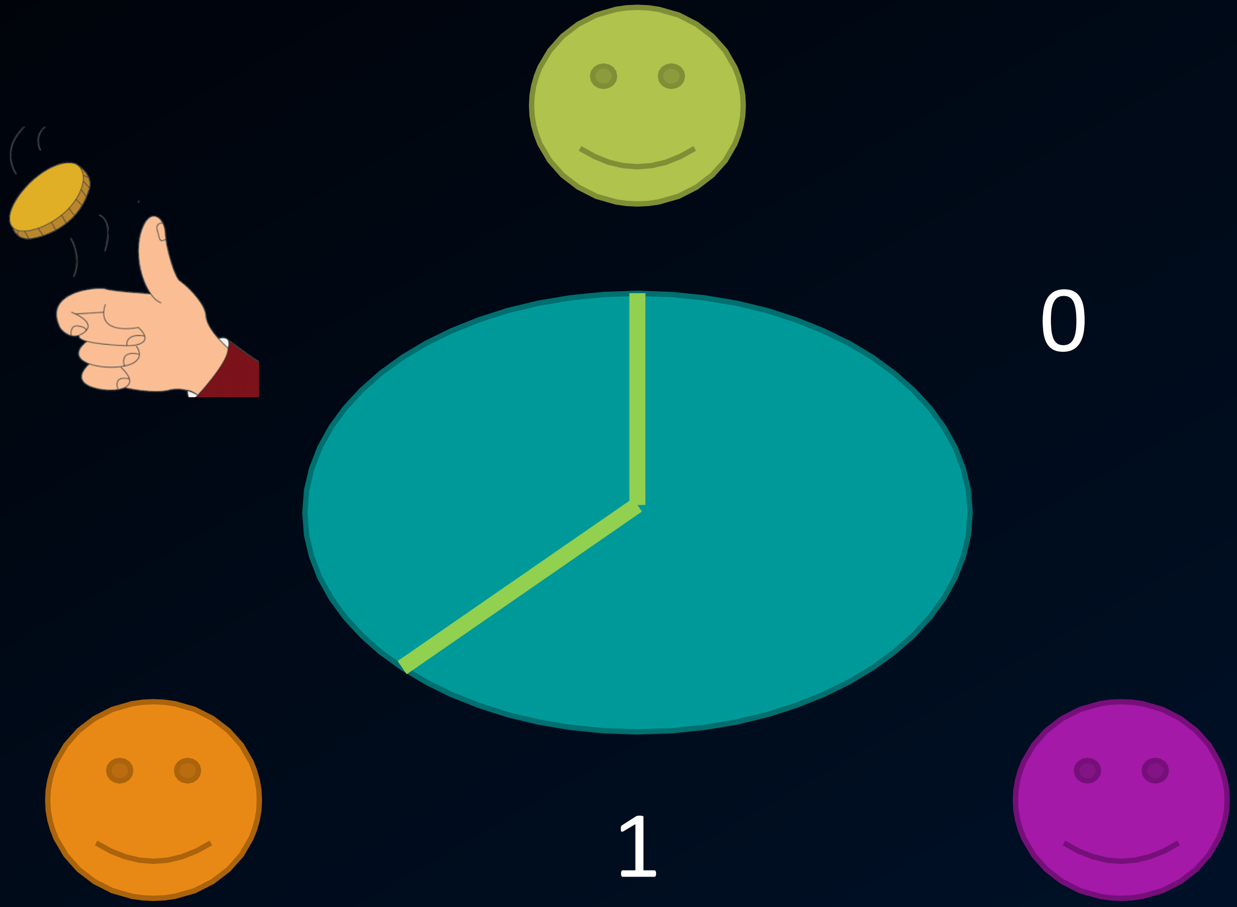
The Premise – The Cryptographers at Dinner

The Premise – The Cryptographers at Dinner

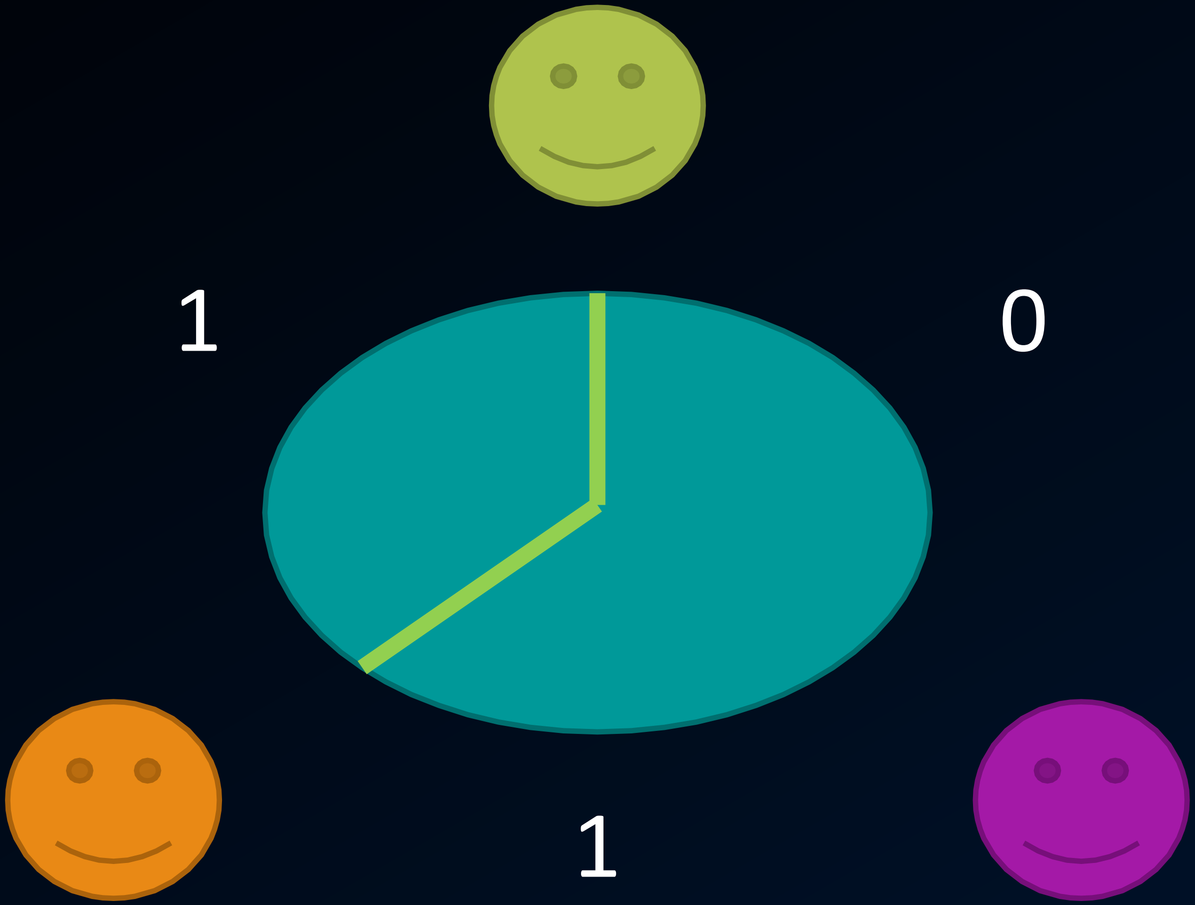
The Premise – The Cryptographers at Dinner



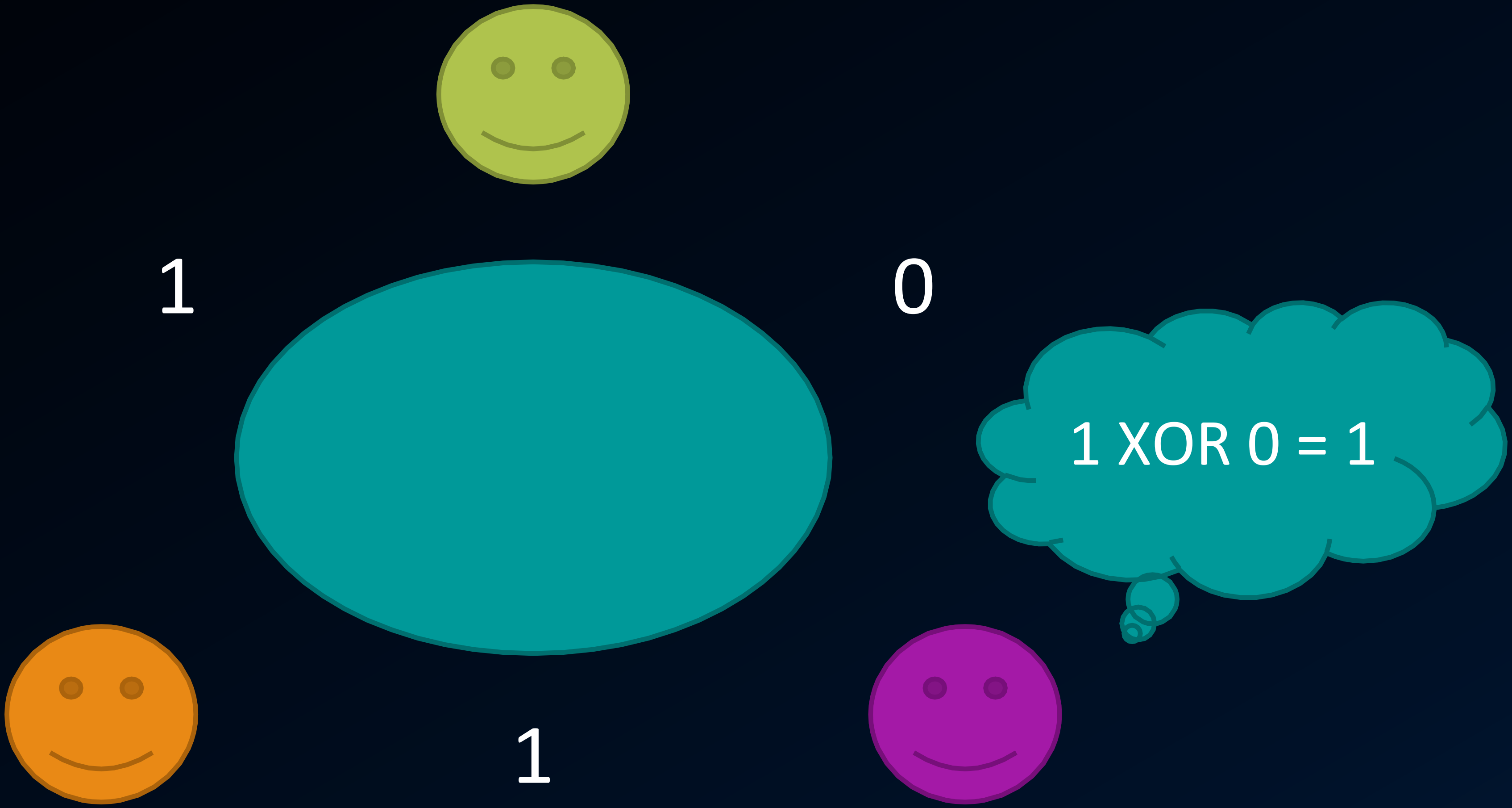
The Premise – The Cryptographers at Dinner



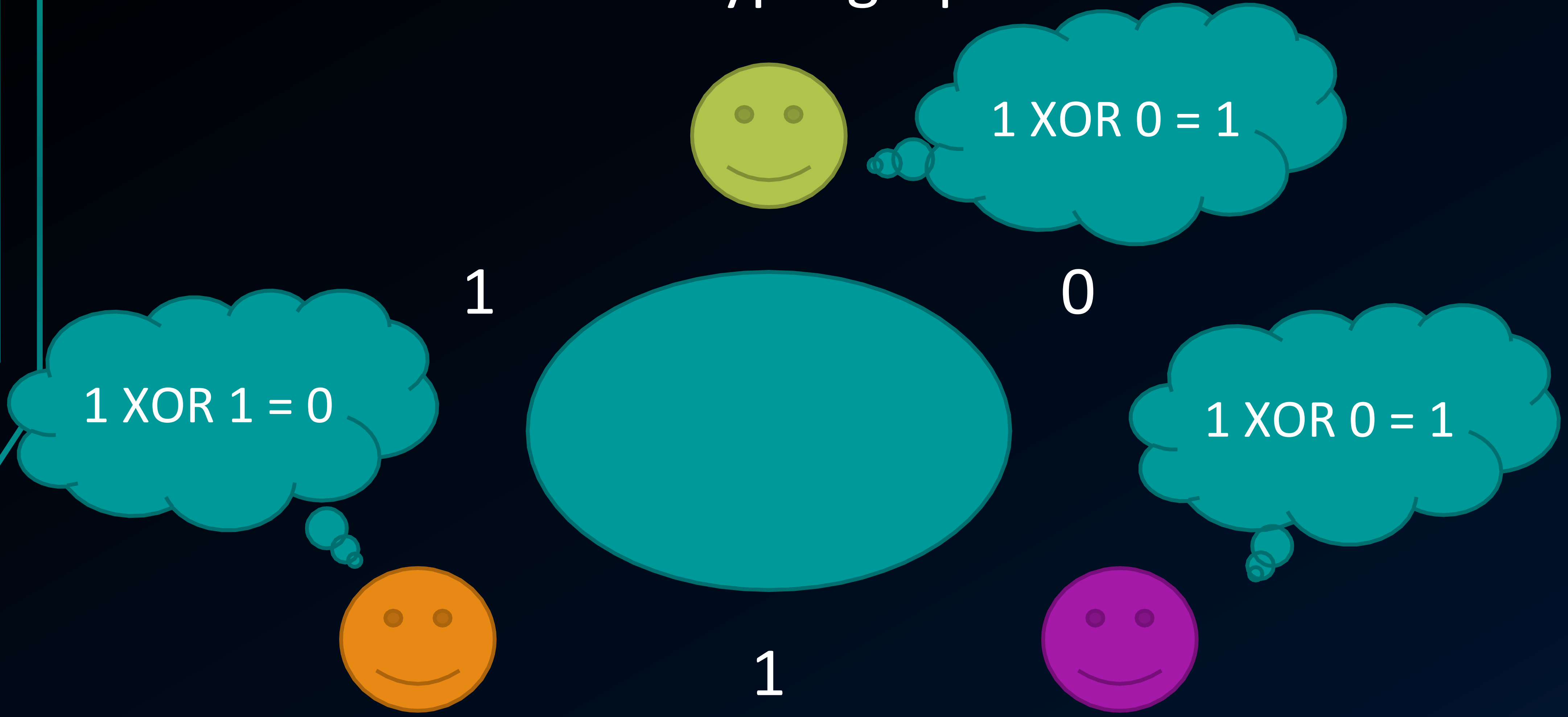
The Premise – The Cryptographers at Dinner



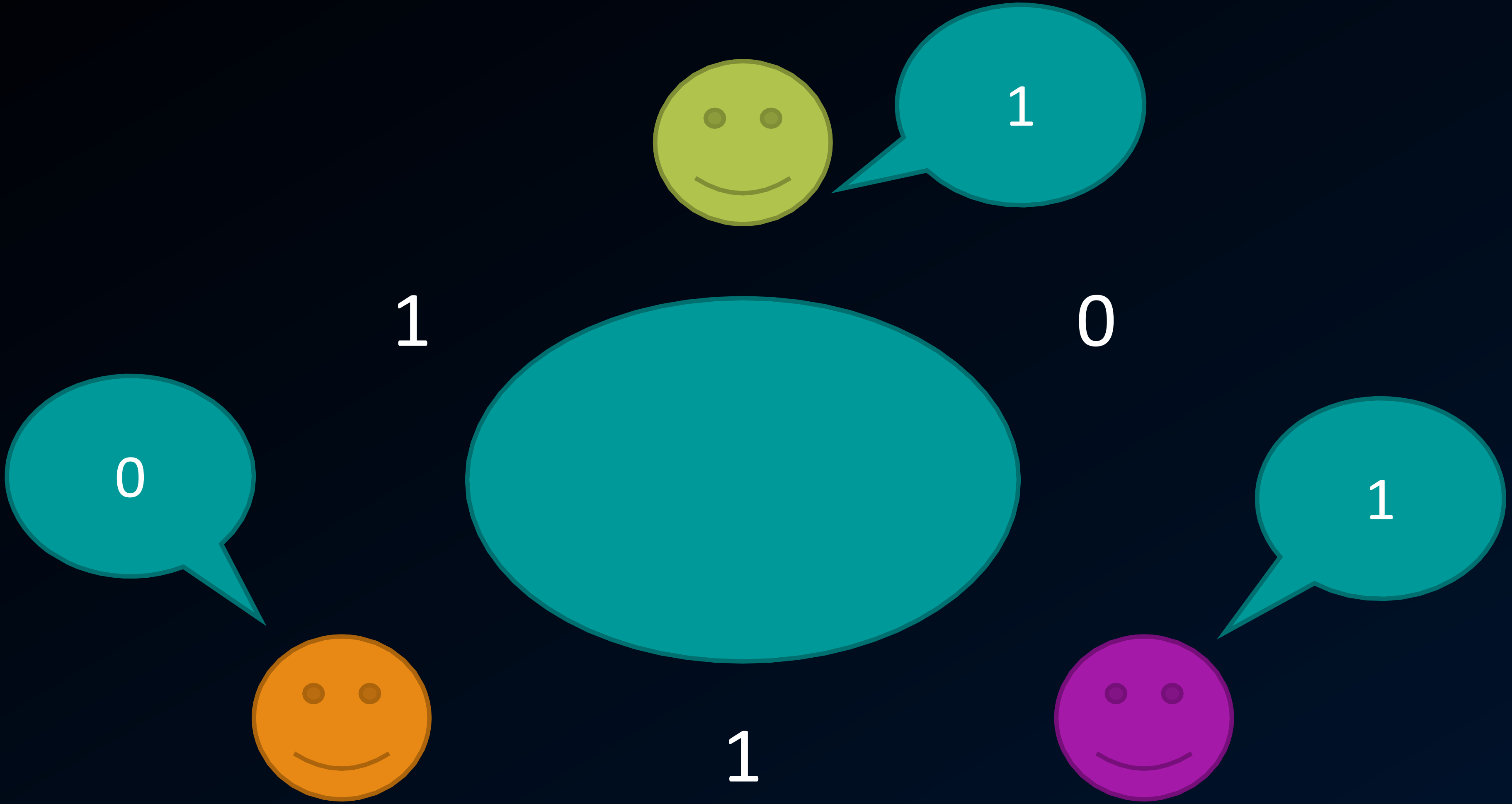
The Premise – The Cryptographers at Dinner



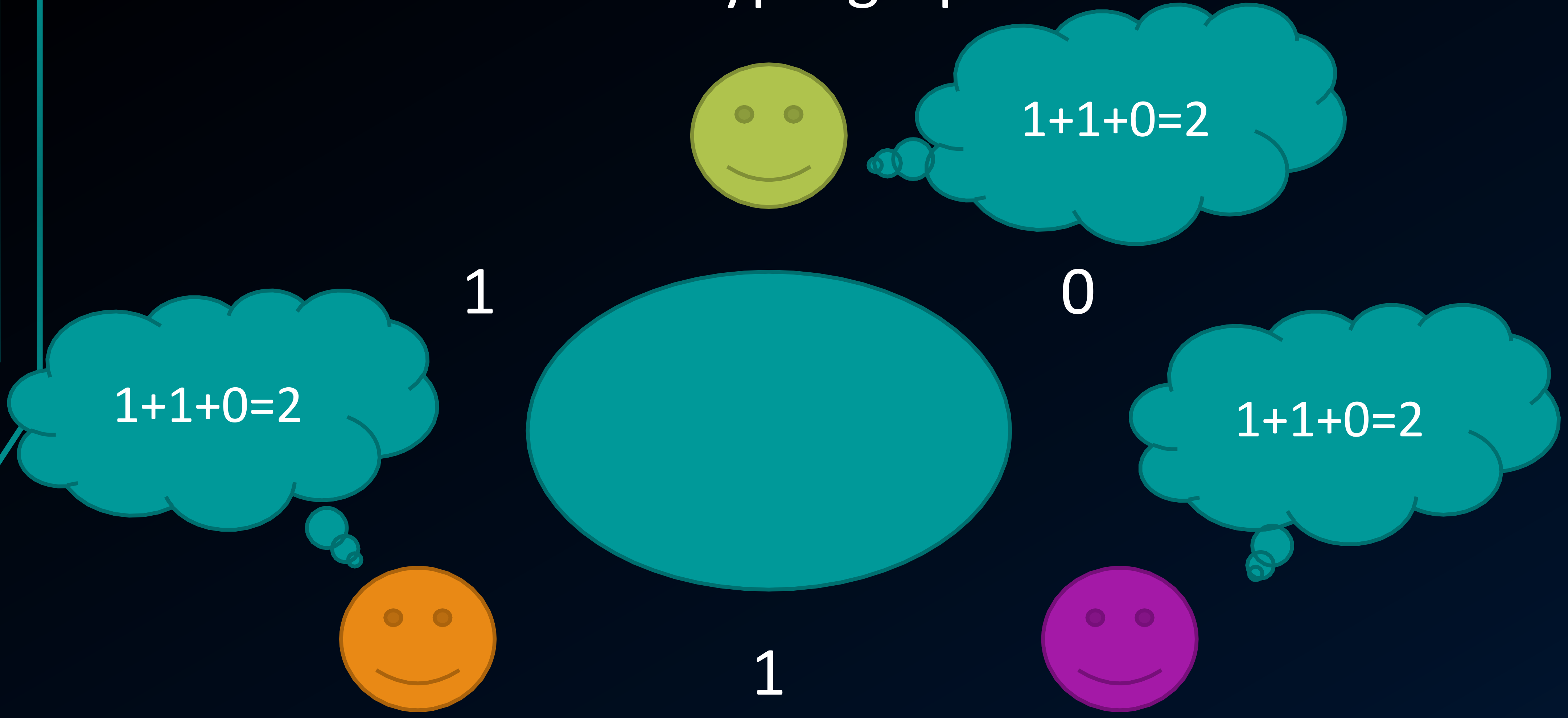
The Premise – The Cryptographers at Dinner



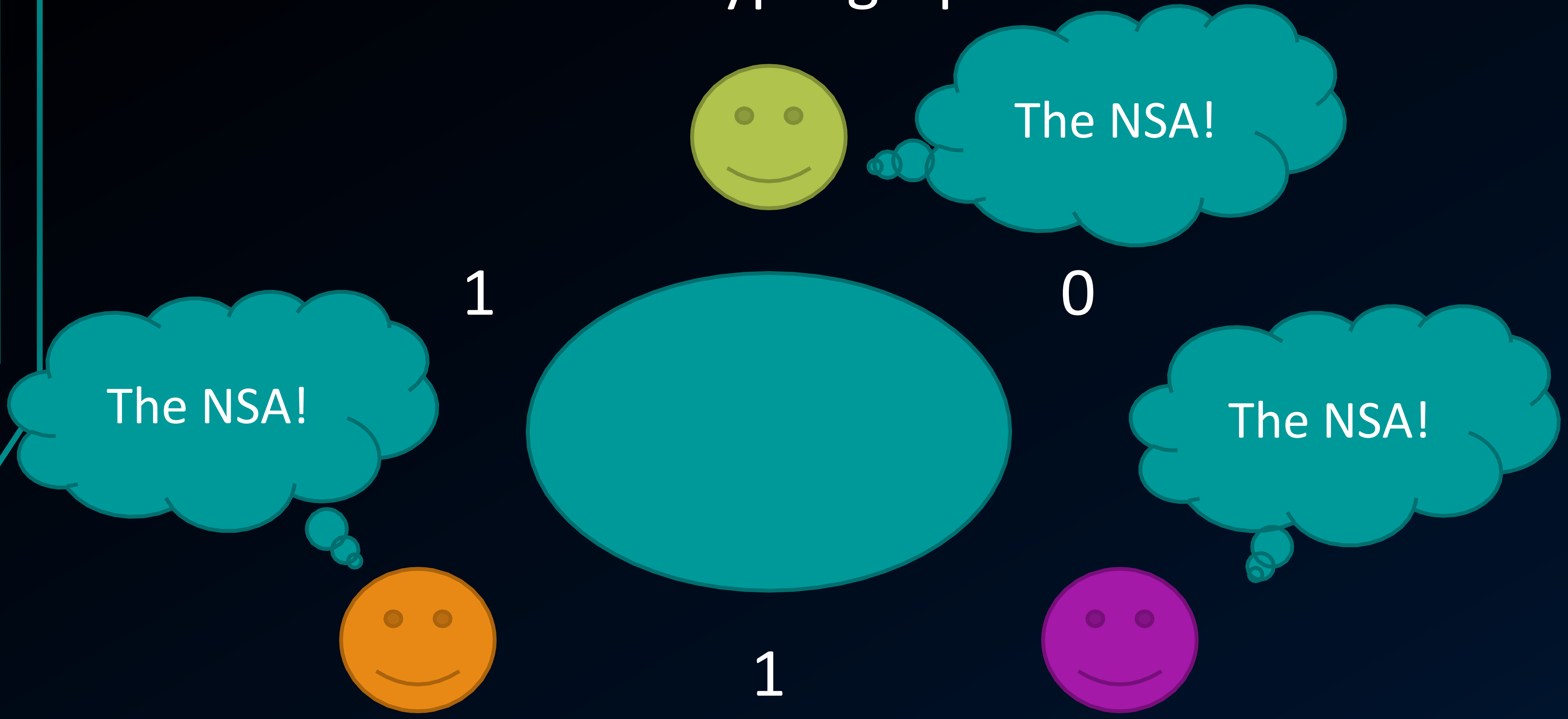
The Premise – The Cryptographers at Dinner



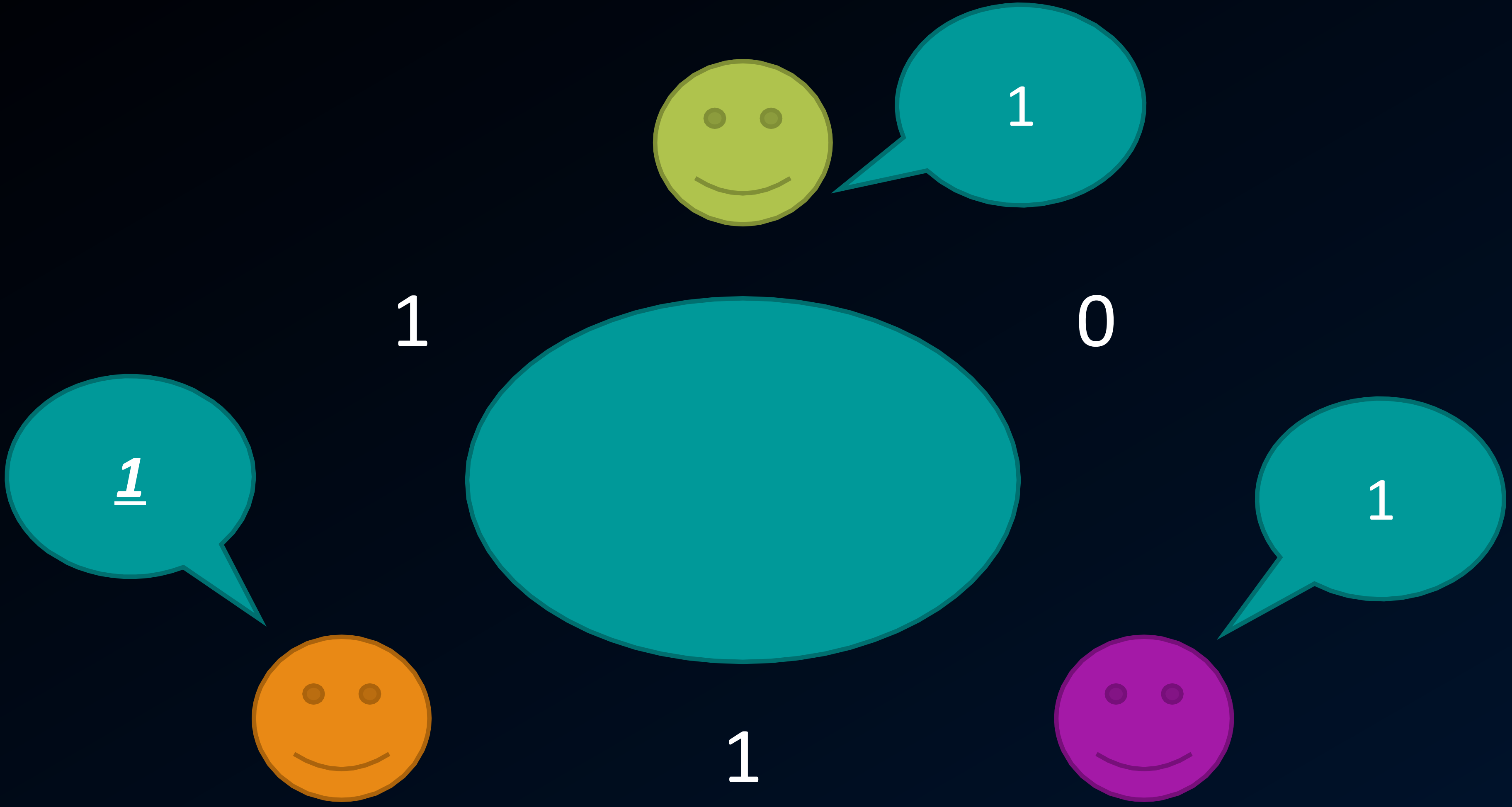
The Premise – The Cryptographers at Dinner



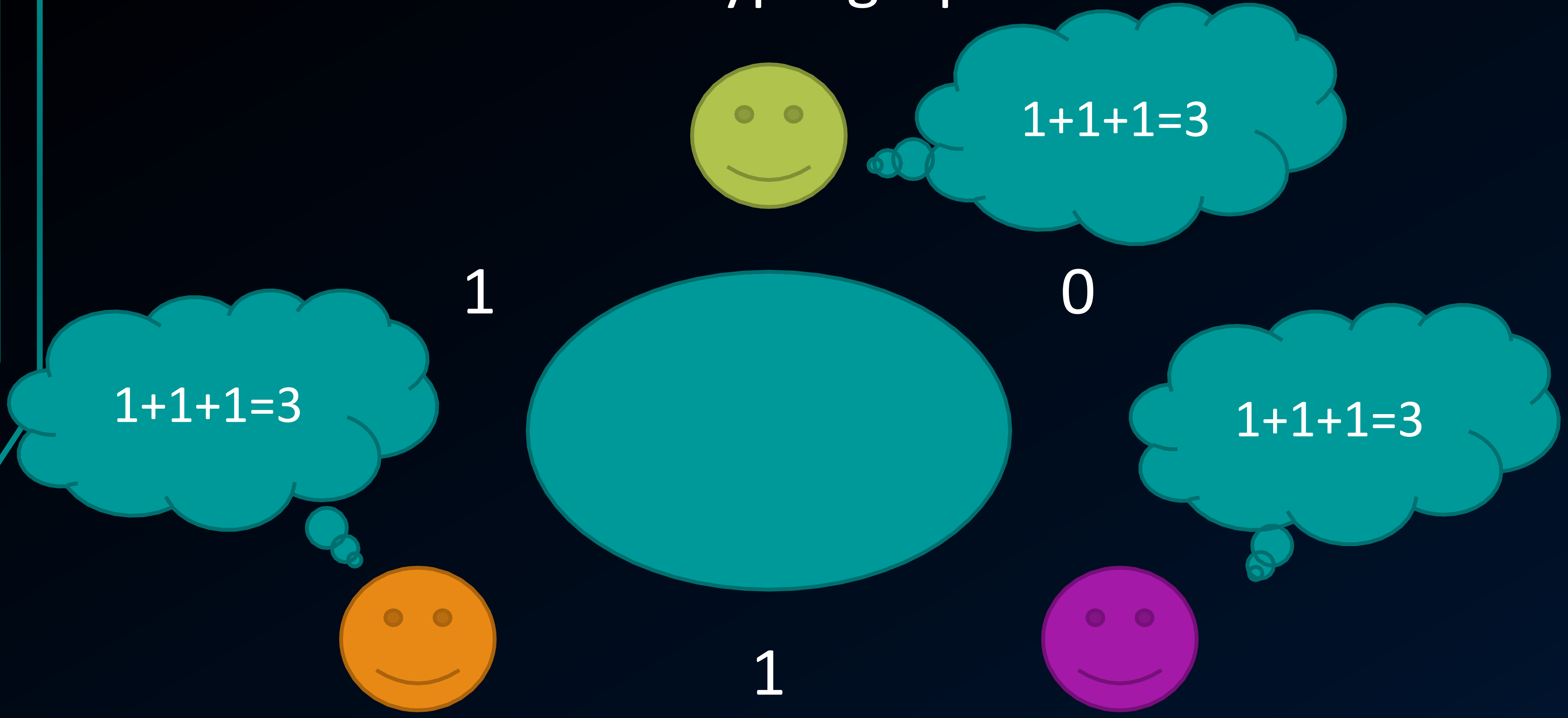
The Premise – The Cryptographers at Dinner



The Premise – The Cryptographers at Dinner



The Premise – The Cryptographers at Dinner



1+1+1=3

1

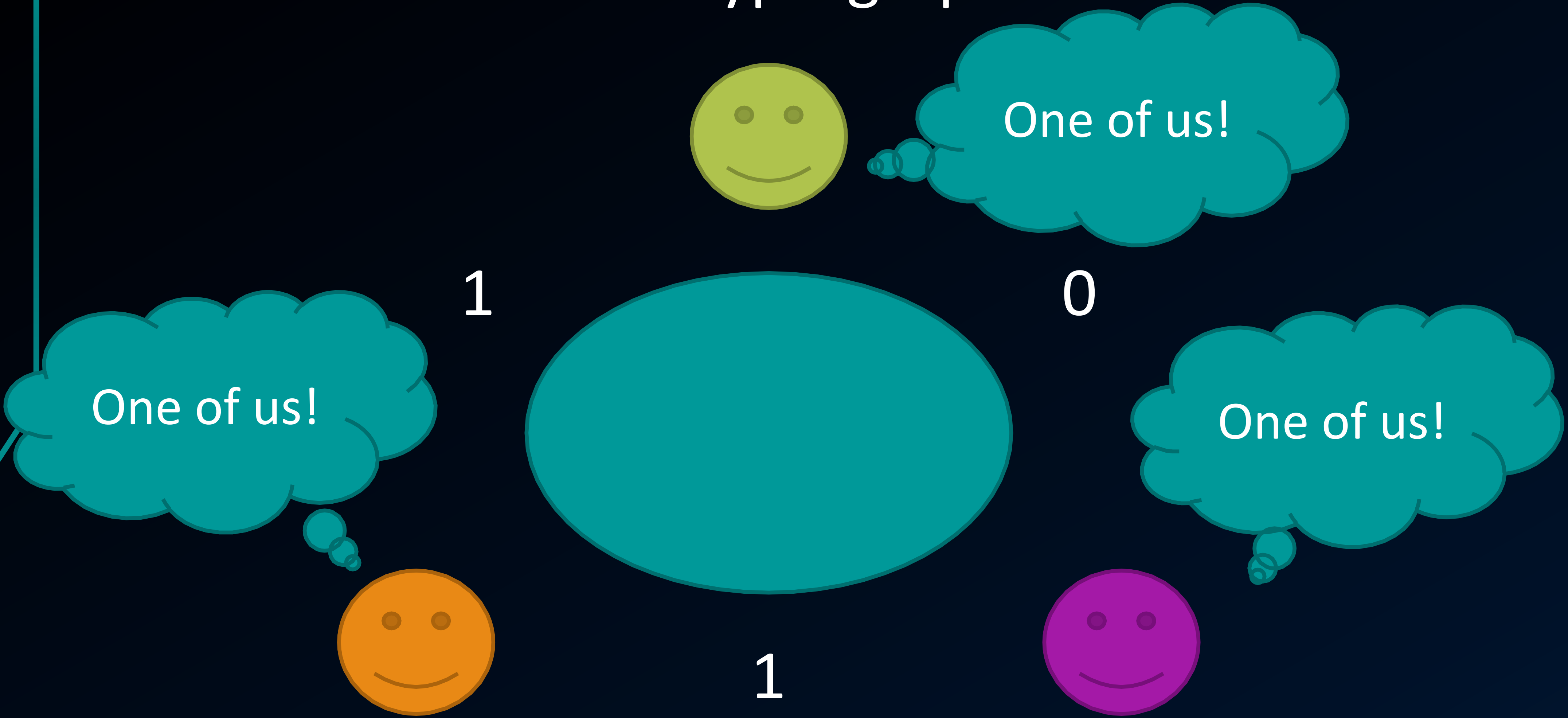
0

1+1+1=3

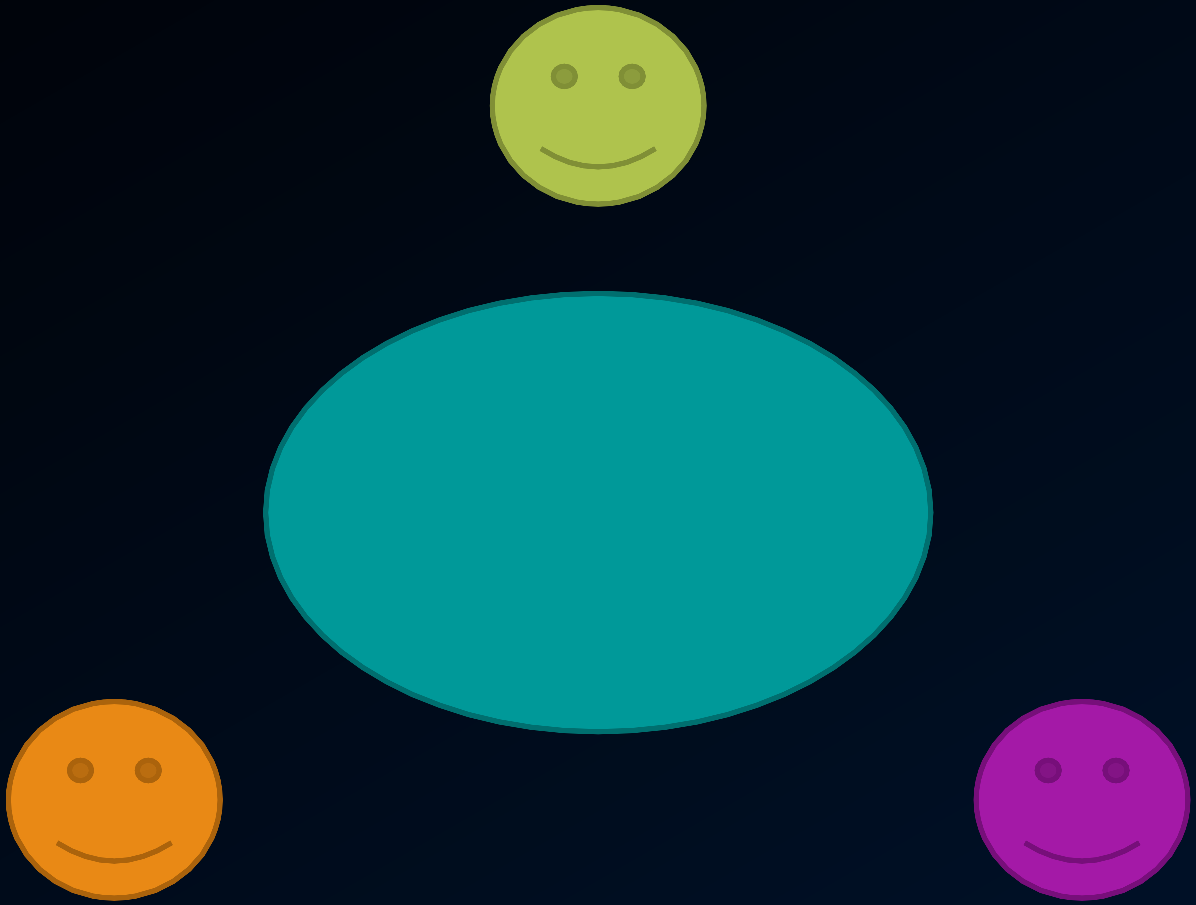
1+1+1=3

1

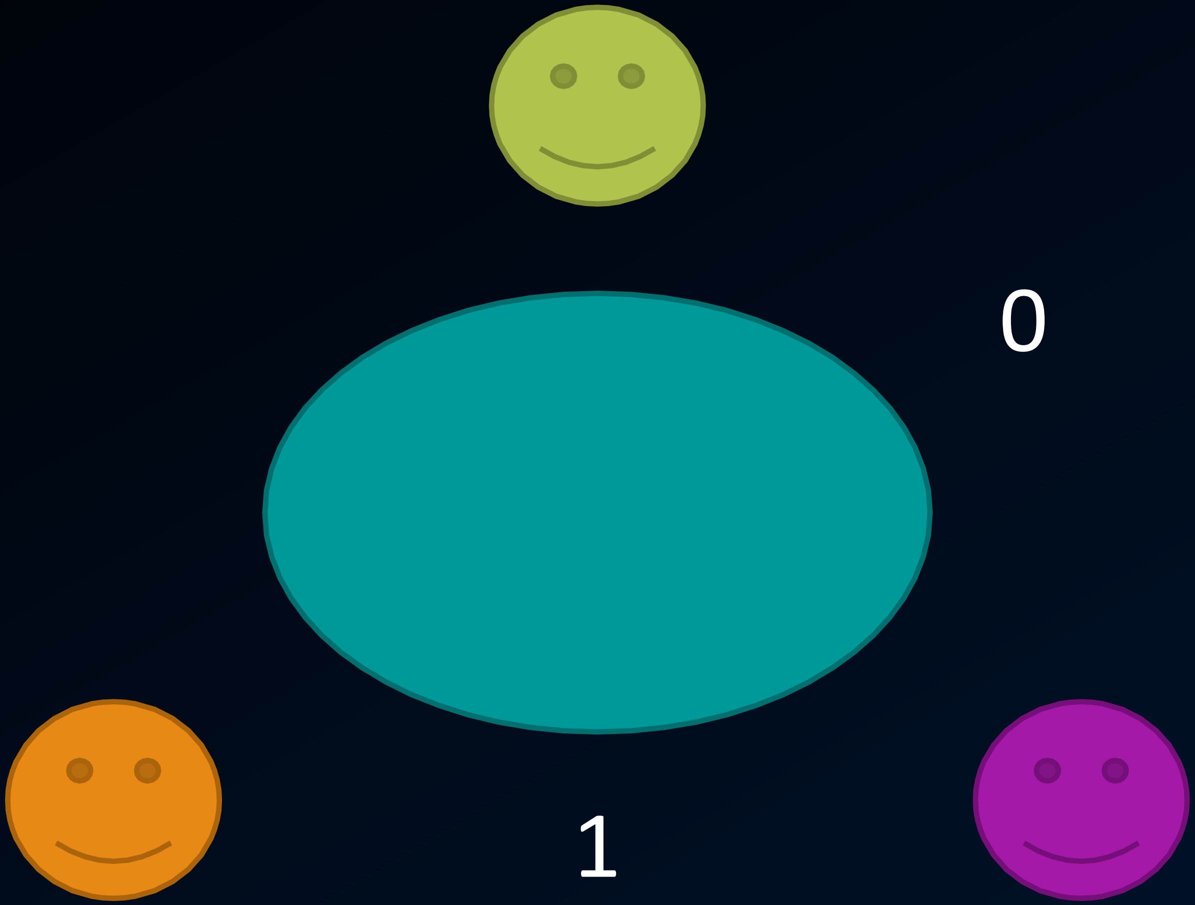
The Premise – The Cryptographers at Dinner



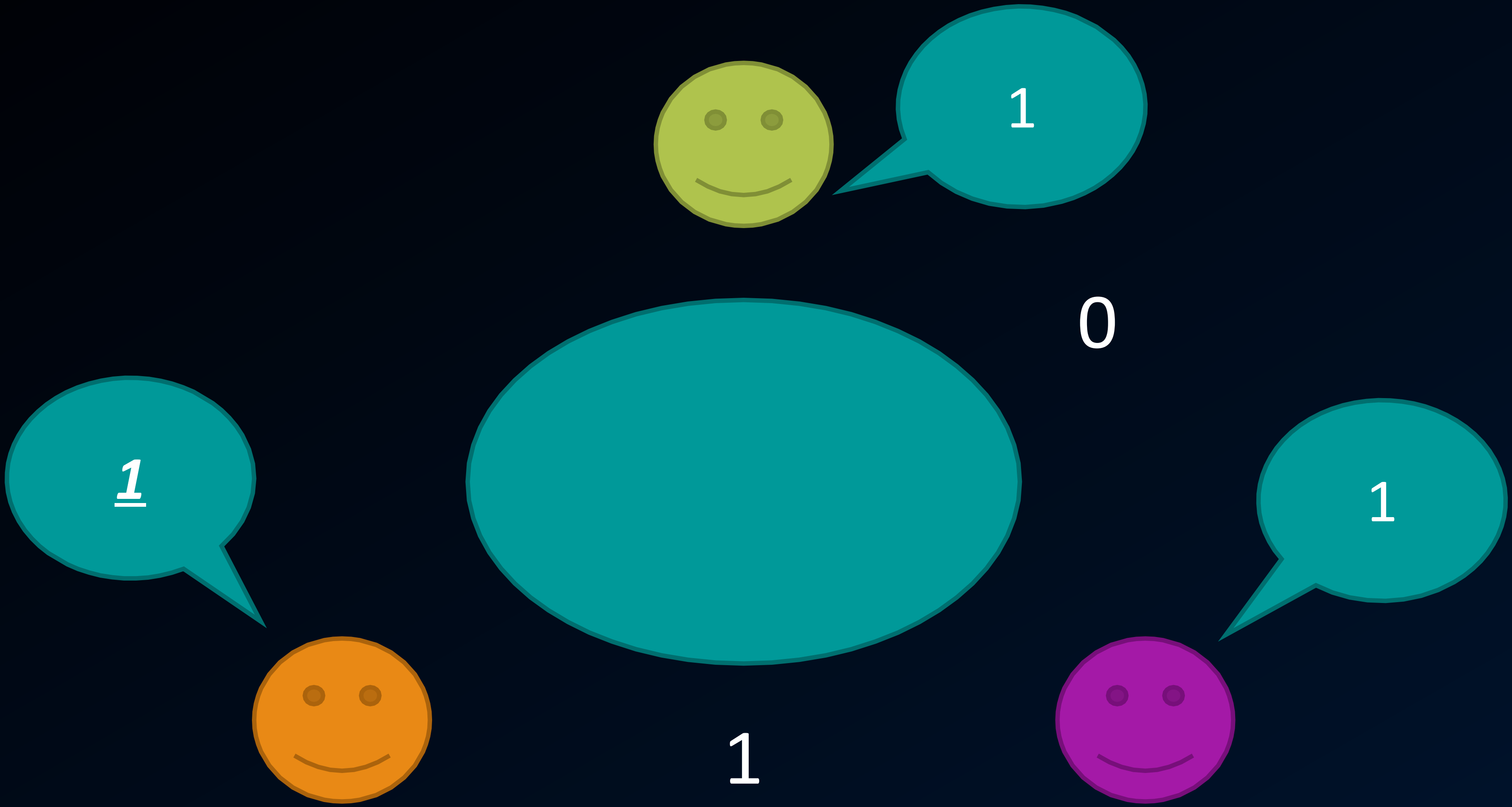
Why does this work?



Why does this work?



Why does this work?



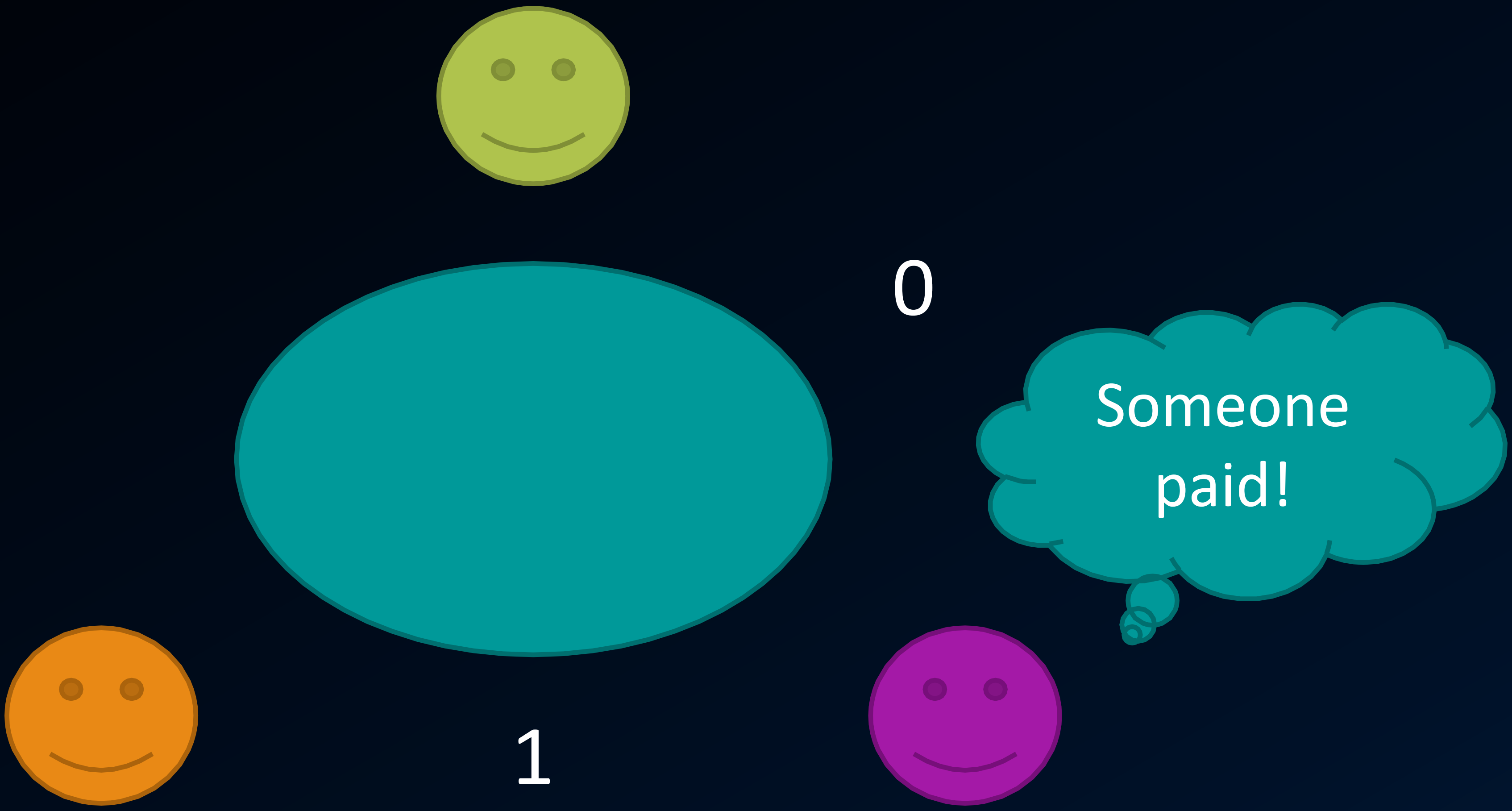
Why does this work?

0

1+1+1=3

1

Why does this work?

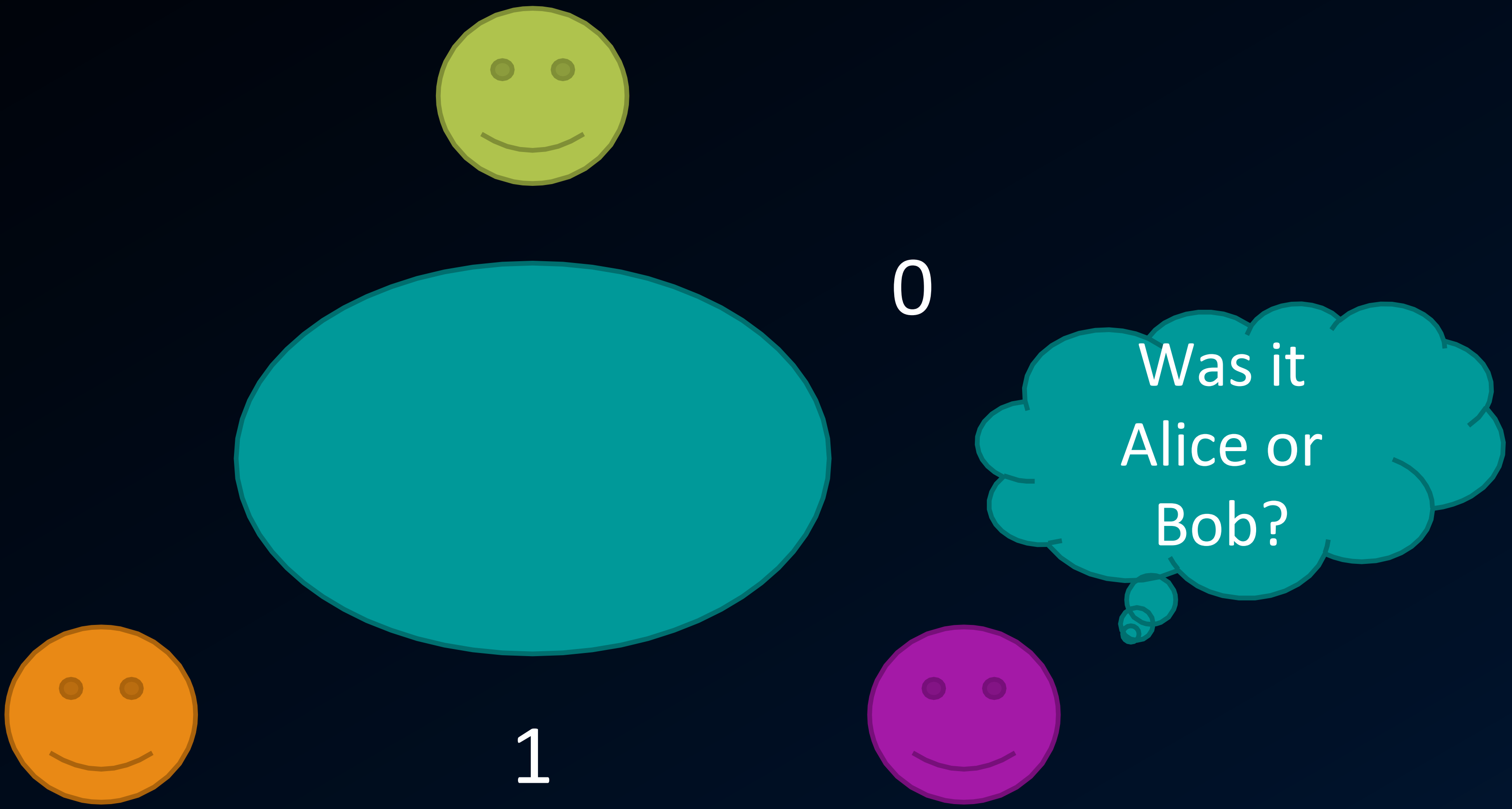


0

Someone paid!

1

Why does this work?

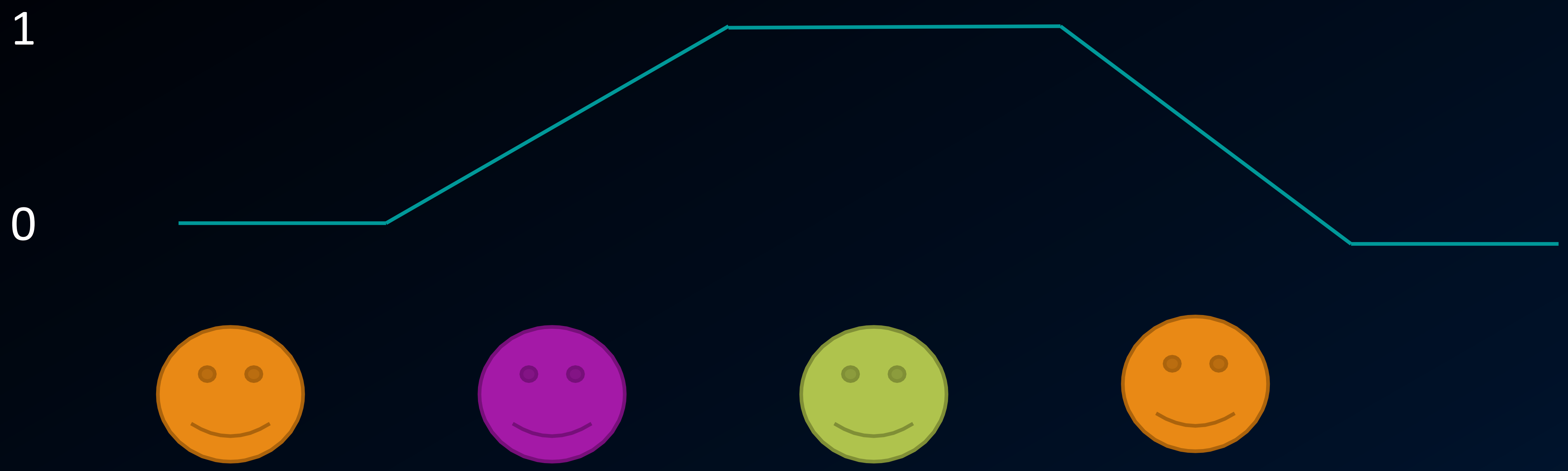


0

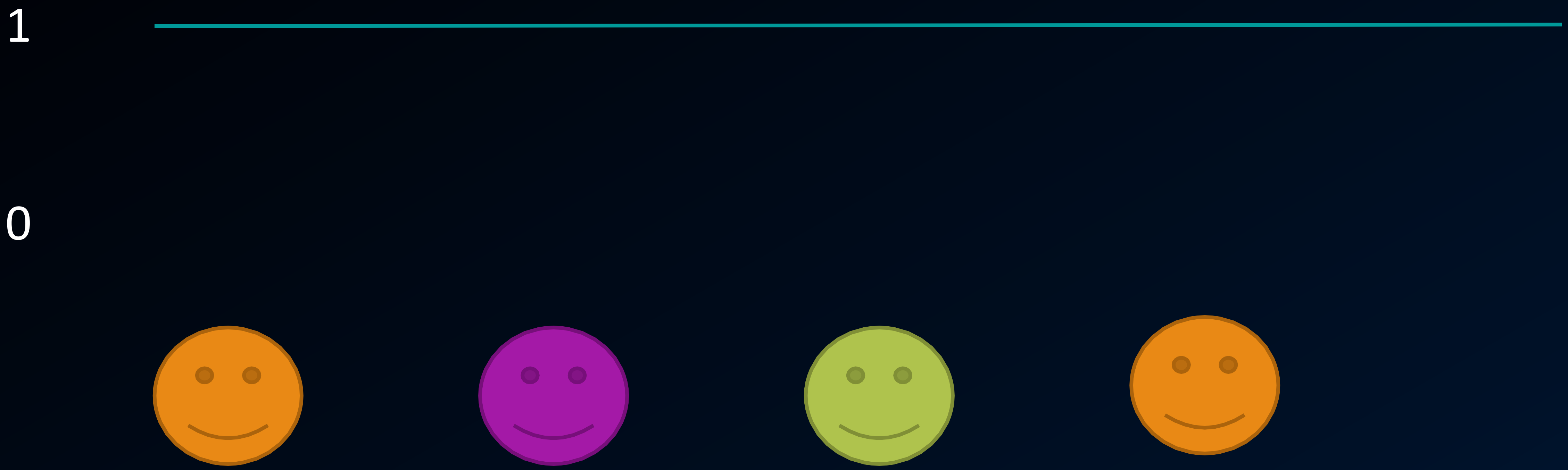
Was it Alice or Bob?

1

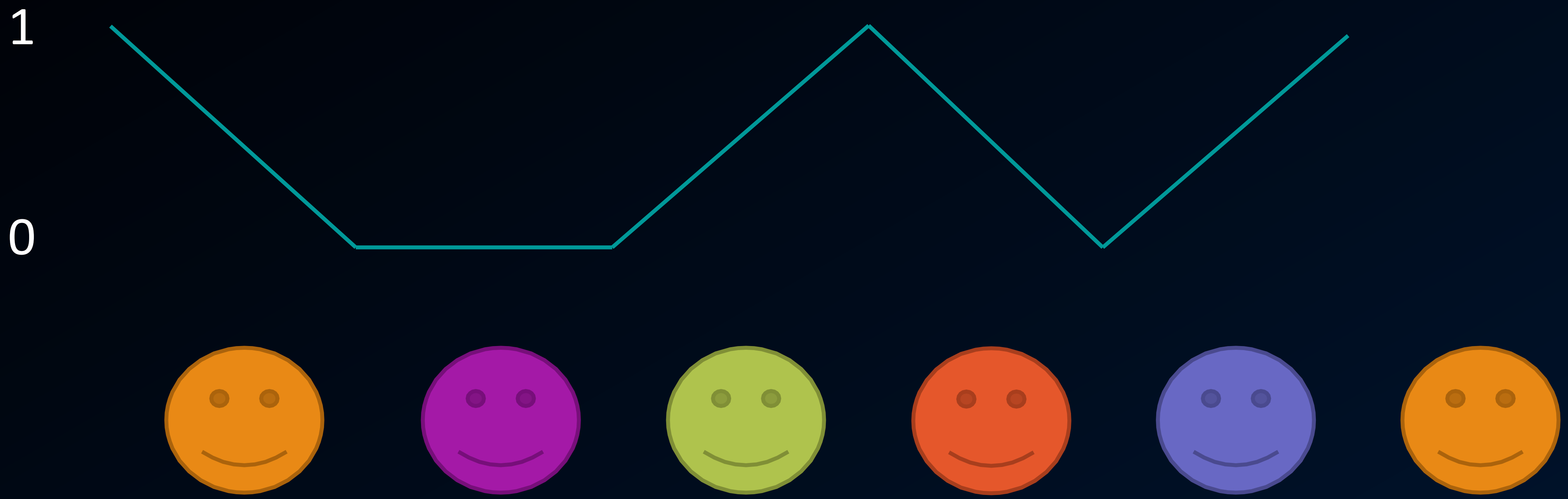
Why is it always even?



Why is it always even?



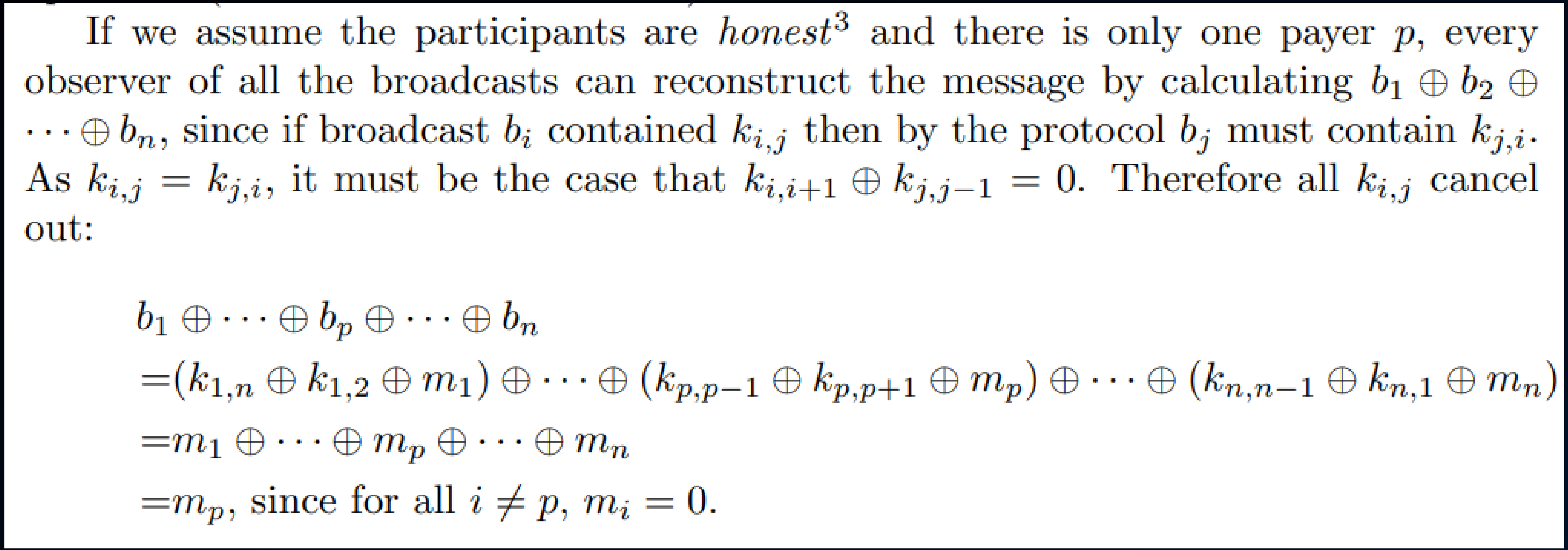
Why is it always even?



Summary of the Protocol (1 bit of information)

- Phase 1 – a 1 bit secret must be shared between each participant and their 2 neighbors, regardless of how many participants there are.
- Phase 2 – Each participant must XOR their two shares secrets and then broadcast the resulting 1 bit message.
- Phase 2’ – If you are the payer, simply flip the result of the XOR and broadcast the resulting 1 bit message
- Phase 3 – All messages are collected by all participants and if the sum of the bits is even, the message is 0, otherwise 1.

Generalizing Dining Cryptographer Networks



If we assume the participants are *honest*[3] and there is only one payer p, every observer of all the broadcasts can reconstruct the message by calculating $b_1 \oplus b_2 \oplus \cdots \oplus b_n$, since if broadcast b_i contained $k_{i,j}$ then by the protocol b_j must contain $k_{j,i}$. As $k_{i,j} = k_{j,i}$, it must be the case that $k_{i,i+1} \oplus k_{j,j-1} = 0$. Therefore all $k_{i,j}$ cancel out:

$$\begin{aligned}
& b_1 \oplus \cdots \oplus b_p \oplus \cdots \oplus b_n \\
=& (k_{1,n} \oplus k_{1,2} \oplus m_1) \oplus \cdots \oplus (k_{p,p-1} \oplus k_{p,p+1} \oplus m_p) \oplus \cdots \oplus (k_{n,n-1} \oplus k_{n,1} \oplus m_n) \\
=& m_1 \oplus \cdots \oplus m_p \oplus \cdots \oplus m_n \\
=& m_p, \text{ since for all } i \neq p,\ m_i = 0.
\end{aligned}$$

Harreveld Pg.2

Logisim Example

Issues with DCNets

- They demand a lot of **bandwidth** from the network. To propagate a message of size ***m*** with ***n*** participants it requires at a minimum n*m bits to be broadcast and 2m secret bits to be shared.
- They can only allow for **1 participant** to message the group per round, otherwise the messages interfere with each other.
- They are **easy to break** with a single un-cooperative participant.

Summary of DC Nets

- By following a strict protocol of 3 phases per round, we can establish a truly anonymous communication for any of the participants, so long as their shared secrets remain secret.
- Benefits: We could use DC Nets for CoinJoin participants who want to anonymously broadcast their outputs to the group.
- Downsides – Very fragile system, easily broken by malicious participants, and also quite a bit slower.

Discussion