WHAT IS ANONYMITY?
WHAT IS ANONYMITY?

msg
WHAT IS ANONYMITY?

msg
WHAT IS ANONYMITY?

Who sent/received this?

msg
WHAT IS ANONYMITY?

potential senders

msg

Who sent/received this?
WHAT IS ANONYMITY?

potential senders

potential receivers

Who sent/received this?

msg
WHAT IS ANONYMITY?

**SENDER-ANONYMITY:**
An attacker overhearing communication cannot determine the **true sender** from a larger set of **potential senders**.

(The attacker might learn the receiver)
WHAT IS ANONYMITY?

SENDER-ANONYMITY:
An attacker overhearing communication cannot determine the **true sender** from a larger set of **potential senders**.

(The attacker might learn the receiver)
WHAT IS ANONYMITY?

**SENDER-ANONYMITY:**
An attacker overhearing communication cannot determine the **true sender** from a larger set of **potential senders**.

(The attacker might learn the receiver)

**RECEIVER-ANONYMITY:**
An attacker overhearing communication cannot determine the **true receiver** from a larger set of **potential receivers**.

(The attacker might learn the sender)
WHAT IS ANONYMITY?

SEND-ANONYMITY:
An attacker overhearing communication cannot determine the true sender from a larger set of potential senders.
(The attacker might learn the receiver)

RECEIVER-ANONYMITY:
An attacker overhearing communication cannot determine the true receiver from a larger set of potential receivers.
(The attacker might learn the sender)
WHAT IS ANONYMITY?

SENDER-ANONYMITY:
An attacker overhearing communication cannot determine the **true sender** from a larger set of **potential senders**.
(The attacker might learn the receiver)

RECEIVER-ANONYMITY:
An attacker overhearing communication cannot determine the **true receiver** from a larger set of **potential receivers**.
(The attacker might learn the sender)

**Number stations:** Spies use small FM transmitters to broadcast encoded messages; the set of potential receivers is **everyone** within broadcast range.
WHAT IS ANONYMITY?

**SENDER-ANONYMITY:**
An attacker overhearing communication cannot determine the true sender from a larger set of potential senders.
(The attacker might learn the receiver)

**RECEIVER-ANONYMITY:**
An attacker overhearing communication cannot determine the true receiver from a larger set of potential receivers.
(The attacker might learn the sender)

**SENDER-RECEIVER-ANONYMITY:**
An attacker overhearing communication cannot determine the communicating pair from a larger set of potential pairs.
(The attacker might learn the sender or the receiver, but not both)
To quantify “how anonymous” a system / protocol is, we think of how large the anonymity set is: the set of other potential users / computers that could have performed the action.

**Intuition:**
The more other people it *might have been*, the less likely they can pin it to any individual user.

**Example:**
In a densely populated area, the anonymity set of a number station can be tens of millions.
ANONYMITY IS NOT PRIVACY (NOT EXACTLY)

Both of these are fungible terms, but generally speaking...

PRIVACY:
Maintaining confidentiality about an entity’s personally-identifying information (PII)

ANONYMITY:
Maintaining confidentiality about with whom (or whether) an entity communicates

The connection is complicated:
With whom you communicate is a form of PII

Sharing PII can de-anonymize your communication

This lecture: we will focus on anonymous communication
THE DINING CRYPTOGRAPHER’S PROBLEM

Each individual knows 2 bits \( b_{\text{left}} \) and \( b_{\text{right}} \)
THE DINING CRYPTOGRAPHER’S PROBLEM

Each individual knows 2 bits $b_{\text{left}}$ and $b_{\text{right}}$

PROBLEM:
One person has a message $m$ to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?
THE DINING CRYPTOGRAPHER'S PROBLEM

PROBLEM:
One person has a message \( m \) to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

PROTOCOL

Each individual knows 2 bits
\( b_{\text{left}} \) and \( b_{\text{right}} \)
THE DINING CRYPTOGRAPHER'S PROBLEM

PROBLEM:
One person has a message $m$ to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

PROTOCOL
Each pair props up a menu
Private communication channel

Each individual knows 2 bits
$b_{\text{left}}$ and $b_{\text{right}}$
**THE DINING CRYPTOGRAPHER’S PROBLEM**

**PROBLEM:**
One person has a message $m$ to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

**PROTOCOL**
Each pair props up a menu
Private communication channel

Each individual knows 2 bits $b_{\text{left}}$ and $b_{\text{right}}$
**THE DINING CRYPTOGRAPHER’S PROBLEM**

**PROBLEM:**
One person has a message $m$ to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

**PROTOCOL**
Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
$b_{\text{left}}$ and $b_{\text{right}}$
**PROBLEM:**
One person has a message $m$ to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

**PROTOCOL:**
Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits $b_{\text{left}}$ and $b_{\text{right}}$
THE DINING CRYPTOGRAPHER’S PROBLEM

PROBLEM:
One person has a message $m$ to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

PROTOCOL

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
$b_{\text{left}}$ and $b_{\text{right}}$

Every individual says a bit
Broadcasts:
If they have $m$: $m \oplus b_{\text{left}} \oplus b_{\text{right}}$
Otherwise: $b_{\text{left}} \oplus b_{\text{right}}$
**THE DINING CRYPTOGRAPHER’S PROBLEM**

**PROBLEM:**
One person has a message $m$ to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

**PROTOCOL**

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits $b_{\text{left}}$ and $b_{\text{right}}$

Every individual says a bit
Broadcasts:

- If they have $m$:
  $$m \oplus b_{\text{left}} \oplus b_{\text{right}}$$
- Otherwise:
  $$b_{\text{left}} \oplus b_{\text{right}}$$

XOR all messages to recover $m$
THE DINING CRYPTOGRAPHER’S PROBLEM

PROBLEM:
One person has a message \( m \) to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

PROTOCOL

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
\( b_{\text{left}} \) and \( b_{\text{right}} \)

Every individual says a bit
Broadcasts:

If they have \( m \):
\[ m \oplus b_{\text{left}} \oplus b_{\text{right}} \]

Otherwise:
\[ b_{\text{left}} \oplus b_{\text{right}} \]

XOR all messages to recover \( m \)
**THE DINING CRYPTOGRAPHER’S PROBLEM**

**PROBLEM:**
One person has a message $m$ to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

**PROTOCOL**

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
$b_{\text{left}}$ and $b_{\text{right}}$

Every individual says a bit
Broadcasts:

If they have $m$:
\[ m \oplus b_{\text{left}} \oplus b_{\text{right}} \]

Otherwise:
\[ b_{\text{left}} \oplus b_{\text{right}} \]

XOR all messages to recover $m$
THE DINING CRYPTOGRAPHER'S PROBLEM

PROBLEM:
One person has a message $m$ to send (let's say it's a bit)
Can this person reveal that bit without revealing their identity?

PROTOCOL

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
$b_{\text{left}}$ and $b_{\text{right}}$

Every individual says a bit
Broadcasts:

If they have $m$:
$$m \oplus b_{\text{left}} \oplus b_{\text{right}}$$

Otherwise:
$$b_{\text{left}} \oplus b_{\text{right}}$$

XOR all messages to recover $m$
**THE DINING CRYPTOGRAPHER’S PROBLEM**

**PROBLEM:**
One person has a message $m$ to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

**PROTOCOL**

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
$b_{left}$ and $b_{right}$

Every individual says a bit
Broadcasts:

If they have $m$:
$m \oplus b_{left} \oplus b_{right}$

Otherwise:
$b_{left} \oplus b_{right}$

XOR all messages to recover $m$
THE DINING CRYPTOGRAPHER’S PROBLEM

PROBLEM:
One person has a message $m$ to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

PROTOCOL

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits $b_{\text{left}}$ and $b_{\text{right}}$

Every individual says a bit
Broadcasts:
If they have $m$:
$m \oplus b_{\text{left}} \oplus b_{\text{right}}$
Otherwise:
$b_{\text{left}} \oplus b_{\text{right}}$

XOR all messages to recover $m$
**THE DINING CRYPTOGRAPHER’S PROBLEM**

**PROBLEM:**
One person has a message $m$ to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

**PROTOCOL**

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits $b_{\text{left}}$ and $b_{\text{right}}$

Every individual says a bit
Broadcasts:
- If they have $m$: $m \oplus b_{\text{left}} \oplus b_{\text{right}}$
- Otherwise: $b_{\text{left}} \oplus b_{\text{right}}$

XOR all messages to recover $m$
THE DINING CRYPTOGRAPHER’S PROBLEM

PROBLEM:
One person has a message $m$ to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

PROTOCOL
Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
$b_{\text{left}}$ and $b_{\text{right}}$

Every individual says a bit
broadcasts:
If they have $m$:
$m \oplus b_{\text{left}} \oplus b_{\text{right}}$

Otherwise:
$b_{\text{left}} \oplus b_{\text{right}}$

XOR all messages to recover $m$
THE DINING CRYPTOGRAPHER’S PROBLEM

PROBLEM:
One person has a message $m$ to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

PROTOCOL

Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits $b_{\text{left}}$ and $b_{\text{right}}$

Every individual says a bit
Broadcasts:
If they have $m$:
$$m \oplus b_{\text{left}} \oplus b_{\text{right}}$$
Otherwise:
$$b_{\text{left}} \oplus b_{\text{right}}$$

XOR all messages to recover $m$
**THE DINING CRYPTOGRAPHER’S PROBLEM**

**PROBLEM:**
One person has a message $m$ to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

**PROTOCOL**
Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits $b_{left}$ and $b_{right}$

Every individual says a bit
Broadcasts:
- If they have $m$: $m \oplus b_{left} \oplus b_{right}$
- Otherwise: $b_{left} \oplus b_{right}$

XOR all messages to recover $m$
PROBLEM:
One person has a message \( m \) to send (let’s say it’s a bit)
Can this person reveal that bit without revealing their identity?

\[
0 \oplus 1 \oplus 0 = 1
\]

PROTOCOL
Each pair props up a menu
Private communication channel

Each pair flips a coin
Gets a shared random bit

Each individual knows 2 bits
\( b_{\text{left}} \) and \( b_{\text{right}} \)

Every individual says a bit
Broadcasts:

If they have \( m \):
\[
m \oplus b_{\text{left}} \oplus b_{\text{right}}
\]

Otherwise:
\[
b_{\text{left}} \oplus b_{\text{right}}
\]

XOR all messages to recover \( m \)
WHO LEARNS WHAT?

AFTER THE PROTOCOL

Everyone knows
THEIR $b_{\text{left}}$ and $b_{\text{right}}$
The message $m$
(Whether or not they sent it)

No one learns
The remaining bit
Everyone knows THEIR $b_{\text{left}}$ and $b_{\text{right}}$

The message $m$

(Whether or not they sent it)

No one learns
The remaining bit
WHO LEARNS WHAT?

AFTER THE PROTOCOL

Everyone knows
THEIR $b_{\text{left}}$ and $b_{\text{right}}$
The message $m$
(Whether or not they sent it)

No one learns
The remaining bit
Who learns what?

After the protocol:

Everyone knows THEIR b_{left} and b_{right}
The message m
(Whether or not they sent it)

No one learns
The remaining bit

If b_{AB} = 0
If A sent m, he would have sent 0 \oplus 0 = 0
If B sent m, he would have sent 1 \oplus 0 \oplus 0 = 1
Therefore, B was the sender
WHO LEARNS WHAT?

AFTER THE PROTOCOL

Everyone knows THEIR b\textsubscript{left} and b\textsubscript{right}
The message \( m \)
(Whether or not they sent it)

No one learns
The remaining bit

If \( b_{AB} = 0 \)
- If A sent \( m \), he would have sent \( 0 \oplus 0 = 0 \)
- If B sent \( m \), he would have sent \( 1 \oplus 0 \oplus 0 = 1 \)
Therefore, B was the sender

If \( b_{AB} = 1 \)
- If A sent \( m \), he would have sent \( 1 \oplus 1 \oplus 0 = 0 \)
- If B sent \( m \), he would have sent \( 1 \oplus 0 = 1 \)
Therefore, A was the sender
WHO LEARNS WHAT?

AFTER THE PROTOCOL

Everyone knows
THEIR $b_{\text{left}}$ and $b_{\text{right}}$
The message $m$
(Whether or not they sent it)

No one learns
The remaining bit

If $b_{AB} = 0$
If $A$ sent $m$, he would have sent $0 \oplus 0 = 0$
If $B$ sent $m$, he would have sent $1 \oplus 0 \oplus 0 = 1$
Therefore, $B$ was the sender

If $b_{AB} = 1$
If $A$ sent $m$, he would have sent $1 \oplus 1 \oplus 0 = 0$
If $B$ sent $m$, he would have sent $1 \oplus 0 = 1$
Therefore, $A$ was the sender

Each of these has probability 50% (it was determined by a coin flip)
Everyone knows THEIR b\(_{\text{left}}\) and b\(_{\text{right}}\)

The message \( m \) (Whether or not they sent it)

No one learns

The remaining bit

No one learns

Any information about who sent the message

If \( b_{AB} = 0 \)

If A sent \( m \), he would have sent \( 0 \oplus 0 = 0 \)
If B sent \( m \), he would have sent \( 1 \oplus 0 \oplus 0 = 1 \)

Therefore, B was the sender

If \( b_{AB} = 1 \)

If A sent \( m \), he would have sent \( 1 \oplus 1 \oplus 0 = 0 \)
If B sent \( m \), he would have sent \( 1 \oplus 0 = 1 \)

Therefore, A was the sender

Each of these has probability 50% (it was determined by a coin flip)
INSTEAD OF SENDING BITS
Send streams of packets; flip multiple coins

HOW CAN MORE THAN ONE PERSON NEED TO SEND A MESSAGE?
Take turns? But what happens when two try to send at once?

DIFFICULT BUT NOT IMPOSSIBLE TO SCALE UP
In practice we use something else…
The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability

David Chaum
Center for Mathematics and Computer Science, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

Abstract: Keeping confidential who sends what messages, in a world where any physical transmission can be traced to its origin, seems impossible. The solution presented here is unconditionally cryptographically secure, depending on whether it is based on one-time-use keys or on public keys, respectively. It can be adapted to address efficiently a wide variety of practical considerations.

Key words: Untraceability, Unconditional Security, Pseudonyms.

Introduction

Three cryptographers are sitting down to dinner at their favorite three-star restaurant. Their waiters inform them that arrangements have been made with the maître d'hôtel 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:

Each cryptographer flips an unbiased coin behind his back, between him and the cryptographer on his right, so that only the two of them can see the outcome. Each cryptographer then states aloud whether the two coins he can see—the one he flipped and the one his left-hand neighbor flipped—are the same or different on different sides. If one of the cryptographers is the payer, the state of which he sees is the opposite of what he sees. An odd number of differences uttered at the table indicates that a cryptographer is paying; an even number indicates that NSA is paying (assuming that the dinner was paid for only once). Yet if a cryptographer is paying, neither of the other two learns anything about which cryptographer is the payer.

To see why the protocol is unconditionally secure, consider the dilemma of a cryptographer who is not the payer and who wishes to find out which cryptographer is (if NSA pays, there is no anonymity problem). There are two cases:

1. In case (1) the two coins he sees are the same. One of the other cryptographers said “different,” and the other one said “same.” If the hidden outcome was the same as the one the cryptographer who said “different” is the payer; if the outcome was different, the one who said “same” is the payer. Since the hidden coin is fair, both possibilities are equally likely. In case (2) the coins he sees are of message content for thousands of years [3]. Recently, some new solutions to the “key distribution” problem (the problem of providing each communication with a secret key known only to the communicating pair) have been suggested [1, 2], under the names of public key cryptography. Another cryptographic problem, the “traffic analysis” problem (the problem of keeping secret whose conversations are between whom, and when they converse, will become increasingly important with the growth of electronic mail. This paper presents a solution to the traffic analysis problem that is based on public key cryptography. Brain has solved the traffic analysis problem for networks [1], but requires each participant to trust a common authority. In contrast, systems based on the solution advanced here can be compromised only by an interception or encroachment of all of a set of authorities. Ideally, each participant is an authority.

The following two sections introduce the notation and assumptions. Then the basic concepts are introduced for some special cases involving a series of one or more additions. The final sections cover general purpose mail networks.

Notations

Someone becomes a user of a public key cryptosystem (like that of Diffie, Helman, and Needham [6]) by creating a pair of keys $K$ and $K'$ from a randomly generated seed. The public key $K$ is made known to the other user as an agent who knows in the other key $K'$. The signing algorithm involves the encryption of the signature generated by the encryption algorithm using key $K$. The intended utility of these algorithms over conventional algorithms results because the key keys are known to every user, in the same way that $K^e(x)$ is the same as $K(x)$.

Introduction

Cryptography is the science of secret communication. Cryptographic techniques have been providing security

Footnote: Except only with the kind permission of the author and the publisher, the data in this material cannot be used for transmission to electronic mail and the like. For further information, contact the author and the publisher, or the specific software company.

This work was partially supported by the National Science Foundation under Grant Number 86-12/251 and by the Air Force Office of Scientific Research under Contract 86-03-00572.

The author thanks Martin E. Hellman, Ronald L. Rivest, and the Computer Science Department at the University of California at Berkeley for comments on an earlier draft of this paper.

Communications of the ACM

February 1981

Volume 24

Number 2