Signed messages

What was wrong with the oral messages?

Signed messages

What was wrong with the oral messages?

Signed messages

What was wrong with the oral messages?

Answer: because **traitors lie**, they **alter the contents** of the messages they receive and send.

Signed messages

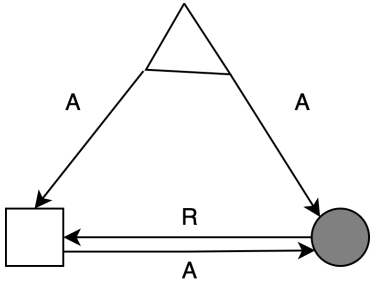
What was wrong with the oral messages?

Answer: because **traitors lie**, they **alter the contents** of the messages they receive and send.

Signed messages

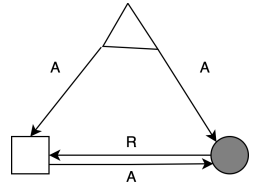
What was wrong with the oral messages?

Answer: because **traitors lie**, they **alter the contents** of the messages they receive and send.



What was wrong with the oral messages?

Answer: because **traitors lie**, they **alter the contents** of the messages they receive and send.



Signed messages

What was wrong with the oral messages?

Answer: because **traitors lie**, they **alter the contents** of the messages they receive and send.

Let's make messages that can not be lied (forged) or any alterations could be detected

What was wrong with the oral messages?
Answer: because **traitors lie**, they **alter the contents** of the messages they receive and send.
Let's make messages that can not be lied (forged) or any alterations could be detected

Formal **&abstract** Assumptions regarding Messages!

A1. Every Message that is sent is delivered correctly.
A2. There receiver of a message knows who sent it.
A3. The absence of a message can be detected.
} → **Oral messages**

A4. (a) A loyal general's signature cannot be forged (**or changed!!**) and any alteration of the contents of his signed messages can be detected
(b) Anyone can verify the authenticity of a general's signature

↓

Signed messages

BUT, in case of assuming A4, What is the purpose of a traitor?????

Formal **&abstract** Assumptions regarding Messages!

A1. Every Message that is sent is delivered correctly.
A2. There receiver of a message knows who sent it.
A3. The absence of a message can be detected.
} → **Oral messages**

A4. (a) A loyal general's signature cannot be forged (**or changed!!**) and any alteration of the contents of his signed messages can be detected
(b) Anyone can verify the authenticity of a general's signature

↓

Signed messages

BUT, in case of assuming A4, What is the purpose of a traitor?????

Signed messages algorithm

Note that we know **the number of m** traitors when running the Alg.

Signed messages algorithm

Note that we know **the number of m** traitors when running the Alg.

Signed messages algorithm

1. Commander sends a **message** having a value v and a signature (which is a sequence of IDs) $\rightarrow$ $\boldsymbol{v:0}$

Signed messages algorithm

1. Commander sends a **message** having a value v and a signature (which is a sequence of IDs) $\rightarrow$ $\boldsymbol{v:0}$

Signed messages algorithm

1. Commander sends a **message** having a value v and a signature (which is a sequence of IDs) $\rightarrow$ $\boldsymbol{v:0}$
2. each lieutenant i receives the message of length k, **adds the v to a V_i set**, **adds his ID** to the message and **sends it to those child lieutenants not having received this message before**

Signed messages algorithm

1. Commander sends a **message** having a value v and a signature (which is a sequence of IDs) $\rightarrow$ $\boldsymbol{v:0}$
2. each lieutenant i receives the message of length k, **adds the v to a V_i set**, **adds his ID** to the message and **sends it to those child lieutenants not having received this message before**

Signed messages algorithm

1. Commander sends a **message** having a value v and a signature (which is a sequence of IDs) $\rightarrow$ $\boldsymbol{v:0}$
2. each lieutenant i receives the message of length k, **adds the v to a V_i set**, **adds his ID** to the message and **sends it to those child lieutenants not having received this message before**
3. when lieutenant i receives no more messages, the lieutenant i applies the **a Choice function** to V_i in order to retrieve an order.

Signed messages algorithm

1. Commander sends a **message** having a value v and a signature (which is a sequence of IDs) $\rightarrow$ $\boldsymbol{v:0}$
2. each lieutenant i receives the message of length k, **adds the v to a V_i set**, **adds his ID** to the message and **sends it to those child lieutenants not having received this message before**
3. when lieutenant i receives no more messages, the lieutenant i applies the **a Choice function** to V_i in order to retrieve an order.

Signed messages algorithm

1. Commander sends a **message** having a value v and a signature (which is a sequence of IDs) $\rightarrow$ $\boldsymbol{v:0}$
2. each lieutenant i receives the message of length k, **adds the v to a V_i set**, **adds his ID** to the message and **sends it to those child lieutenants not having received this message before**
3. when lieutenant i receives no more messages, the lieutenant i applies the **a Choice function** to V_i in order to retrieve an order.

What is that **Choice function?**

Signed messages algorithm

1. Commander sends a **message** having a value v and a signature (which is a sequence of IDs) $\rightarrow$ $\boldsymbol{v:0}$
2. each lieutenant i receives the message of length k, **adds the v to a V_i set**, **adds his ID** to the message and **sends it to those child lieutenants not having received this message before**
3. when lieutenant i receives no more messages, the lieutenant i applies the **a Choice function** to V_i in order to retrieve an order.

What is that **Choice function?**

Choice function

The Choice function could be any **aggregate function** (such as median, average, etc) BUT, it needs to have two essential properties:

Choice function

The Choice function could be any **aggregate function** (such as median, average, etc) BUT, it needs to have two essential properties:

Choice function

The Choice function could be any **aggregate function** (such as median, average, etc) BUT, it needs to have two essential properties:

1. if Set V_i consists of single value v Then, $Choice(V_i) = v$

Choice function

The Choice function could be any **aggregate function** (such as median, average, etc) BUT, it needs to have two essential properties:

1. if Set V_i consists of single value v Then, $Choice(V_i) = v$

Choice function

The Choice function could be any **aggregate function** (such as median, average, etc) BUT, it needs to have two essential properties:

1. if Set V_i consists of single value v Then, $Choice(V_i) = v$
2. $Choice(\emptyset) = RETREAT$

Choice function

The Choice function could be any **aggregate function** (such as median, average, etc) BUT, it needs to have two essential properties:

1. if Set V_i consists of single value v Then, $Choice(V_i) = v$
2. $Choice(\emptyset) = RETREAT$

Formal statement of Signed messages $SM(m)$

Initially $V_i = \emptyset$
(1) The commander signs and sends message $v : 0$ to all lieutenants
(2) For each i:
(A) If Lieutenant i receives a message of the form $v : 0$ from the commander and he hasn't receieved any order, then:
(i) he lets $V_i = v$
(ii) he sends the message $v : 0 : i$ to every other lieutenant.

(B) If Lieutenant i receives a message of the form $v : 0 : j_1 \cdots : j_k$ **and** $v \notin V_i$ then:
(i) he adds v to V_i;
(ii) if $k < m$, then he sends the message $v : 0 : j_1 \cdots : j_k : i$ to every lieutenant other than $j_1, \ldots, j_k$

(3) For each i: When Lieutenant i will receive no more messages, he obeys the order $Choice(V_i)$

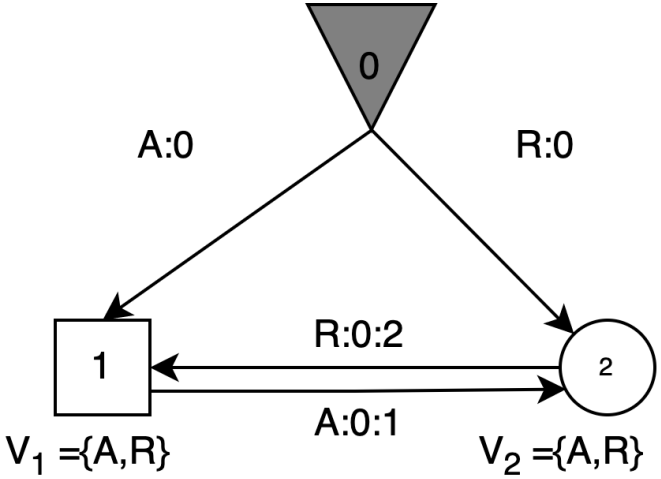
Formal statement of Signed messages $SM(m)$

Initially $V_i = \emptyset$
(1) The commander signs and sends message $v : 0$ to all lieutenants
(2) For each i:
(A) If Lieutenant i receives a message of the form $v : 0$ from the commander and he hasn't receieved any order, then:
(i) he lets $V_i = v$
(ii) he sends the message $v : 0 : i$ to every other lieutenant.

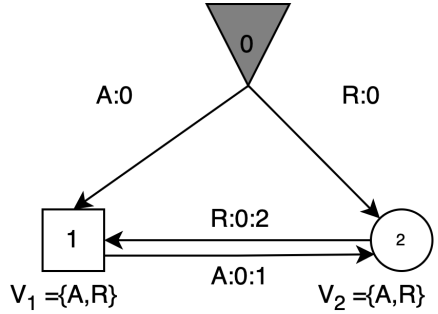
(B) If Lieutenant i receives a message of the form $v : 0 : j_1 \cdots : j_k$ **and** $v \notin V_i$ then:
(i) he adds v to V_i;
(ii) if $k < m$, then he sends the message $v : 0 : j_1 \cdots : j_k : i$ to every lieutenant other than $j_1, \ldots, j_k$

(3) For each i: When Lieutenant i will receive no more messages, he obeys the order $Choice(V_i)$

The impossible case revisited



The impossible case revisited



- The proposed algorithm **increased fault tolerance**, but it is still **expensive** $\rightarrow O(n!)$

- The proposed algorithm **increased fault tolerance**, but it is still **expensive** $\rightarrow O(n!)$

- The proposed algorithm **increased fault tolerance**, but it is still **expensive** $\rightarrow O(n!)$
- In what follows, we give a description of a more **practical and efficient algorithm** which can tolerate Byzantine-faults (i.e. arbitrary messages and faults in nodes) in asynchronous systems, proposed by *Castro M. & Liskov B.*

- The proposed algorithm **increased fault tolerance**, but it is still **expensive** $\rightarrow O(n!)$
- In what follows, we give a description of a more **practical and efficient algorithm** which can tolerate Byzantine-faults (i.e. arbitrary messages and faults in nodes) in asynchronous systems, proposed by *Castro M. & Liskov B.*

- The proposed algorithm **increased fault tolerance**, but it is still **expensive** $\rightarrow O(n!)$
- In what follows, we give a description of a more **practical and efficient algorithm** which can tolerate Byzantine-faults (i.e. arbitrary messages and faults in nodes) in asynchronous systems, proposed by *Castro M. & Liskov B.*
- but first let us define what is meant by **synchronous and asynchronous systems.**

- The proposed algorithm **increased fault tolerance**, but it is still **expensive** $\rightarrow O(n!)$
- In what follows, we give a description of a more **practical and efficient algorithm** which can tolerate Byzantine-faults (i.e. arbitrary messages and faults in nodes) in asynchronous systems, proposed by *Castro M. & Liskov B.*
- but first let us define what is meant by **synchronous and asynchronous systems.**