Message Freedom in MD4 and MD5 Collisions. Application to APOP.

Gaëtan Leurent

Laboratoire d'Informatique de l'École Normale Supérieure,
Département d'Informatique,
45 rue d'Ulm, Paris 75230 Cedex 05, France
Gaetan.Leurent@ens.fr

Fast Software Encryption 2007

Outline

The Post Office Protocol

POP3

- Standard protocol for remote access to a mailbox
- RFC 1460,1725,1939 (first version 1993)
- Supported by virtually every mail provider and every mail user agent
- Widely used (tend to be replaced by IMAP)

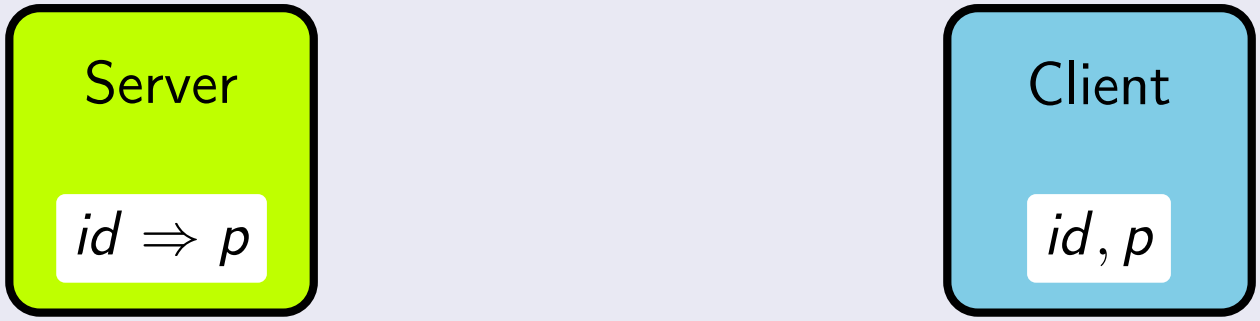
Supported authentication command

- `USER/PASS`: plaintext password
- `APOP`: "secure" authentication
- `AUTH`: any IMAP authentication mechanism: Kerberos, GSS-API, S/Key, CRAM-MD5

APOP authentication

What is APOP?

- Unilateral challenge-response authentication protocol based on a MAC: $h_k(m) = \mathrm{MD5}(m||k)$



- Challenges form: <21921.1174489729@mail.com>
- Origin authentication and replay protection

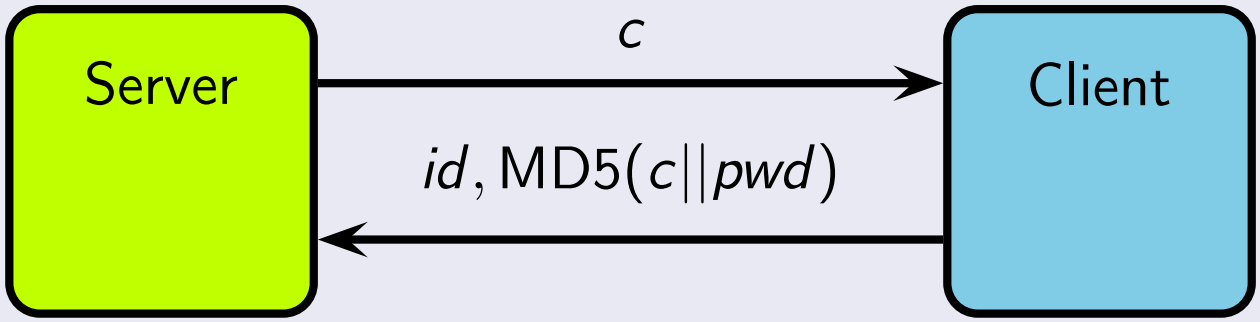
First remarks

- $h_k(m) = \mathrm{MD5}(m||k)$ is not a secure MAC: offline collisions and envelope attack.
- The protocol allows chosen-text attack. There should be some client-chosen randomness.

APOP authentication

What is APOP?

- Unilateral challenge-response authentication protocol based on a MAC: $h_k(m) = \mathrm{MD5}(m||k)$



- Challenges form: <21921.1174489729@mail.com>
- Origin authentication and replay protection

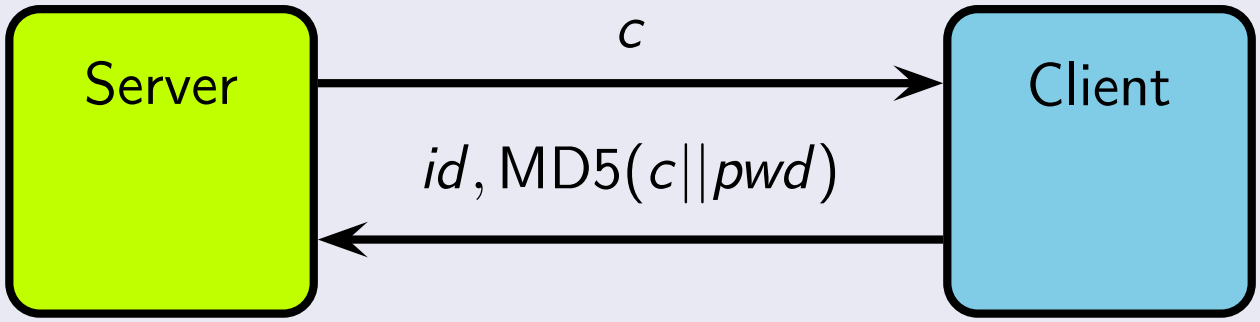
First remarks

- $h_k(m) = \mathrm{MD5}(m||k)$ is not a secure MAC: offline collisions and envelope attack.
- The protocol allows chosen-text attack. There should be some client-chosen randomness.

APOP authentication

What is APOP?

- Unilateral challenge-response authentication protocol based on a MAC: $h_k(m) = \mathrm{MD5}(m||k)$



- Challenges form: <21921.1174489729@mail.com>
- Origin authentication and replay protection

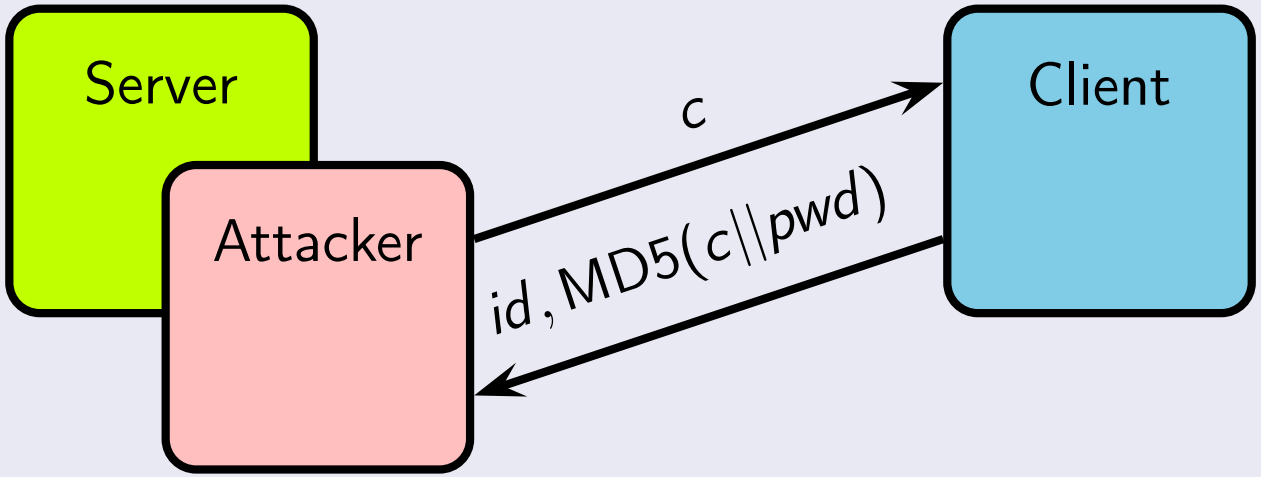
First remarks

- $h_k(m) = \mathrm{MD5}(m||k)$ is not a secure MAC: offline collisions and envelope attack.
- The protocol allows chosen-text attack. There should be some client-chosen randomness.

The APOP Attack

Attack setting

- Active attack: impersonate the server

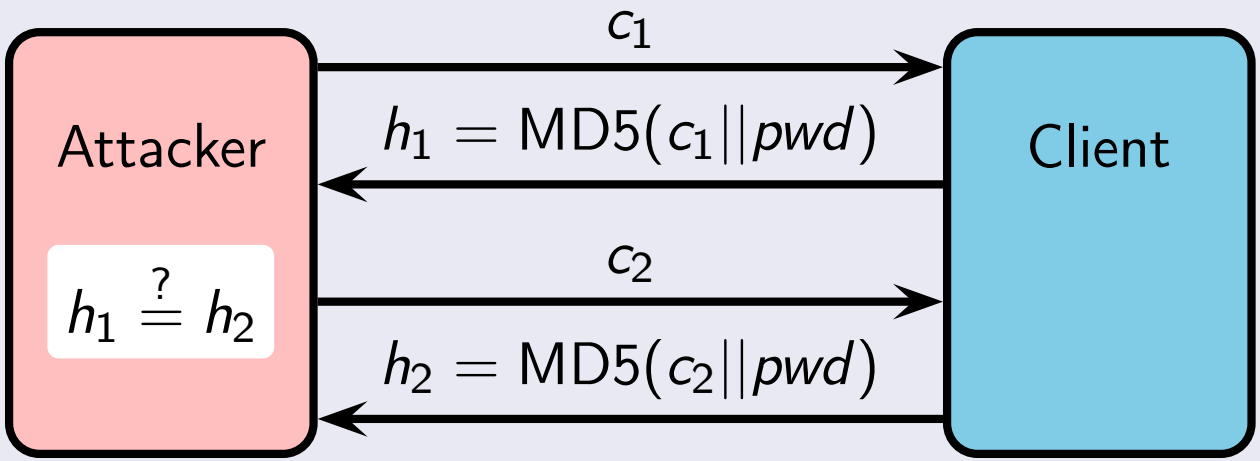


- No server authentication in POP.
- Typical scenario: open WiFi network.
- We can use the client to log on the server and access the mails, but the password should still be safe...

The APOP Attack

Basic idea

Basic idea: use collisions



- Chosen message attack: craft challenges.
- Choose the challenge size to isolate some part of the key in a block.
- Same idea as the key-recovery against the envelope method by Preneel and van Oorschot.

The APOP Attack

Basic idea

Recover the first password character

- If we have an MD5 collision with a specific format:

$M_1 =$ 

$c_1 =$ 

$M_2 =$ 

$c_2 =$ 

- We send c_1 and c_2 as challenges, and we get:

MD5 (

)

MD5 (

)

- Both hashes collide if $p_0 =$ 'x' (unlikely if $p_0 \neq$ 'x').
- We will repeat this with all 'x' until we find p_0.

The APOP Attack

Basic idea

Recover the next password character

- Then we use collisions of the form:

$M_1 =$   $c_1 =$ 

$M_2 =$   $c_2 =$ 

- When we send the challenges c_1 and c_2, we receive:

MD5 (  )

MD5 (  )

- Both hashes collide if $p_1 =$ 'y'.
- We can recover the full password in linear time.

Using Wang's collision to attack APOP

What we need to attack APOP

- We need collisions with control over the end of the last block.
- Birthday paradox too expensive.

Using Wang's collisions

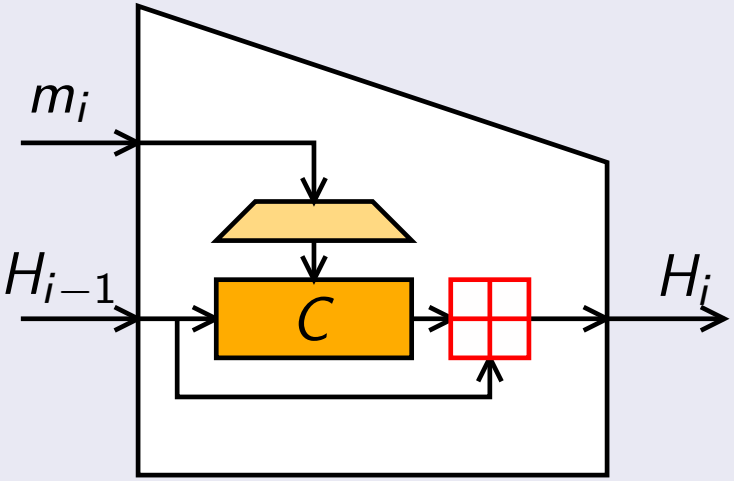
- Wang's collisions look random.
- Due to the message modifications, we loose control over the message value.
- Modify Wang's collision finding technique.

Outline

The MD4 family
Compression function

Compression Function Design

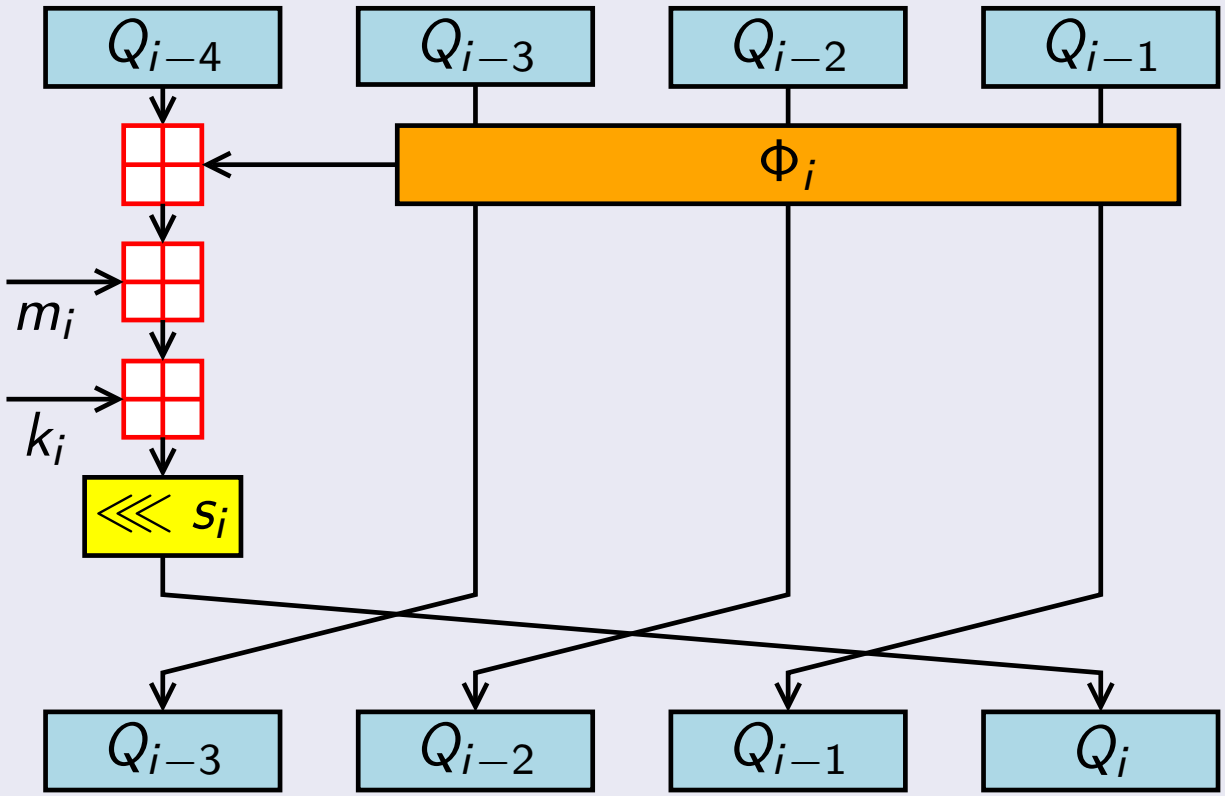
- Davies-Meyer with a Feistel-like cipher.



- Designed to be fast: 32 bit words, and operations available in hardware:
  - additions mod2^{32}: $\boxplus$
  - boolean functions: Φ_i
  - rotations $\lll s_i$
- Message expansion $M = \langle M_0, \ldots M_{15} \rangle \mapsto \langle m_0, \ldots m_{47/63} \rangle$
- 4 words of internal state Q_i updated in rounds of 16 steps

Compression Function

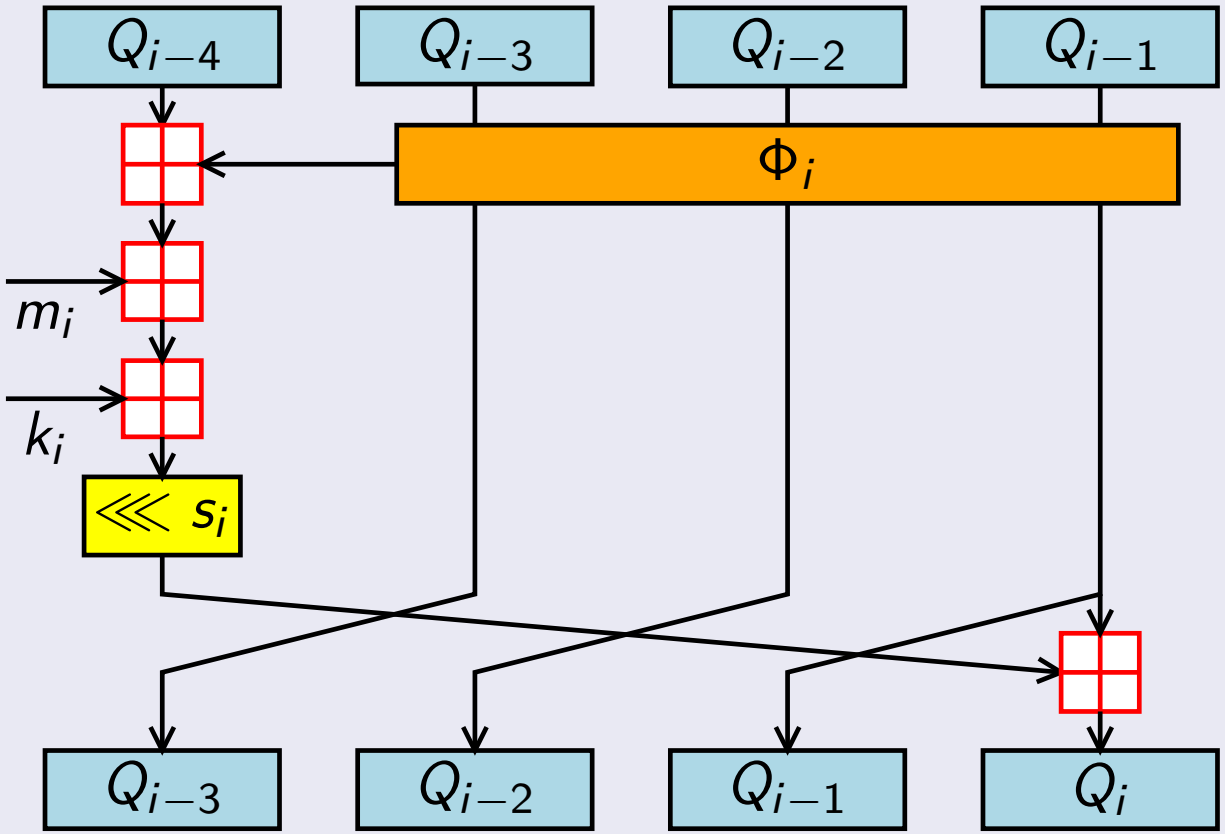
MD4 Step Update



$$Q_i = (Q_{i-4} \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3}) \boxplus m_i \boxplus k_i) \lll s_i$$

Compression Function

MD5 Step Update



$$Q_i = Q_{i-1} \boxplus (Q_{i-4} \boxplus \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3}) \boxplus m_i \boxplus k_i) \lll s_i$$

Collisions in MD4 and MD5

Wang in a nutshell

1. Precomputation:
   - Choose a message difference.
   - Compute a differential path.
   - Derive a set of sufficient conditions.
2. Collision search:
   - Find a message that satisfies the set of conditions.

Main result

We know a difference Δ and a set of conditions on the internal state variables Q_i's, such that:

> *If all the conditions are satisfied by the internal state variable in the computation of $H(M)$, then $H(M) = H(M + \Delta)$.*

Collision search

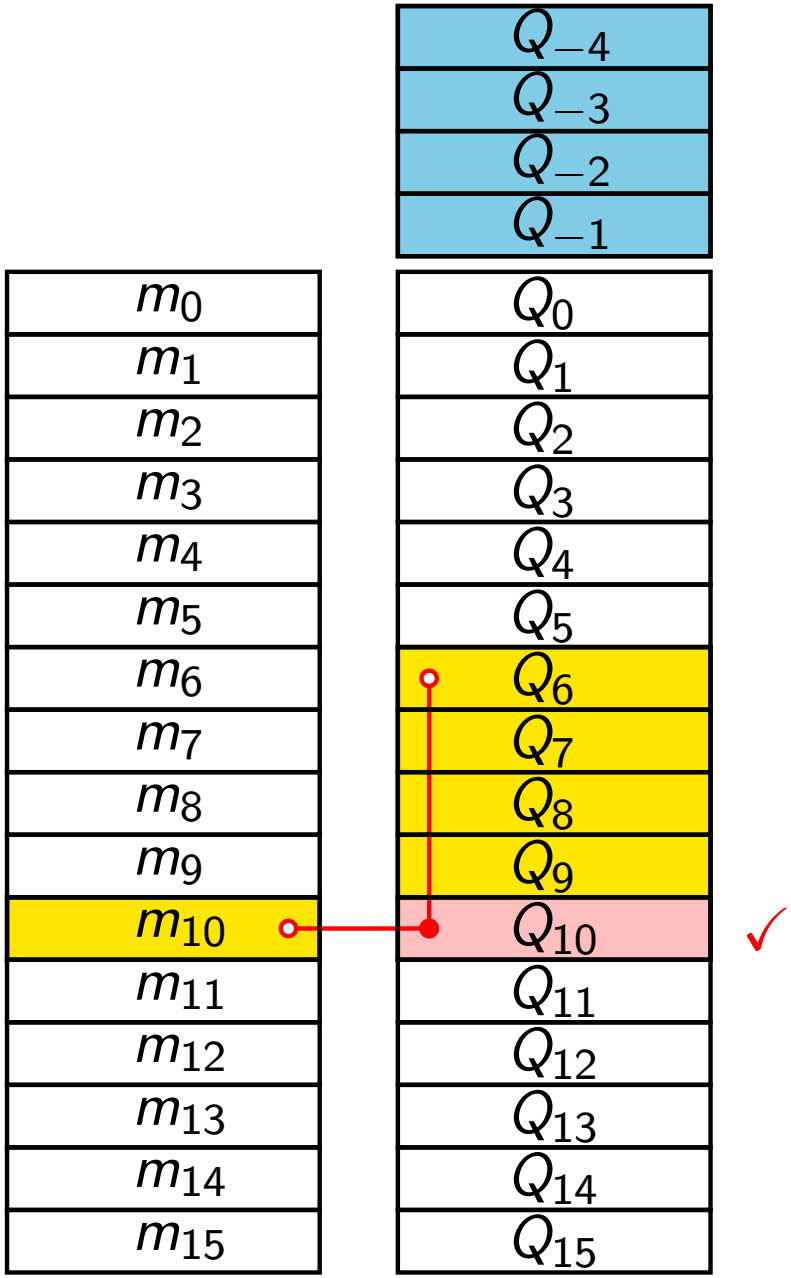
	Q_{-4}
	Q_{-3}
	Q_{-2}
	Q_{-1}
m_0	$1c$ Q_0
m_1	$3c$ Q_1
m_2	$3c$ Q_2
m_3	$5c$ Q_3
m_4	$5c$ Q_4
m_5	$5c$ Q_5
m_6	$6c$ Q_6
m_7	$4c$ Q_7
m_8	$4c$ Q_8
m_9	$4c$ Q_9
m_{10}	$4c$ Q_{10}
m_{11}	$5c$ Q_{11}
m_{12}	$6c$ Q_{12}
m_{13}	$6c$ Q_{13}
m_{14}	$6c$ Q_{14}
m_{15}	$6c$ Q_{15}

	Q_{12}
	Q_{13}
	Q_{14}
	Q_{15}
m_0	$3c$ Q_{16}
m_4	$3c$ Q_{17}
m_8	$3c$ Q_{18}
m_{12}	$2c$ Q_{19}
m_1	$2c$ Q_{20}
m_5	$1c$ Q_{21}
m_9	$2c$ Q_{22}
m_{13}	Q_{23}
m_2	Q_{24}
m_6	Q_{25}
m_{10}	Q_{26}
m_{14}	Q_{27}

Goal

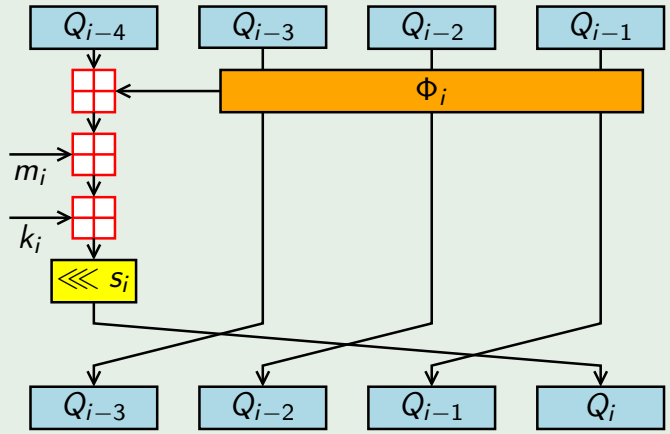
Given the set of conditions, find a message (here on MD4).

Collision search



✓

Basic equations

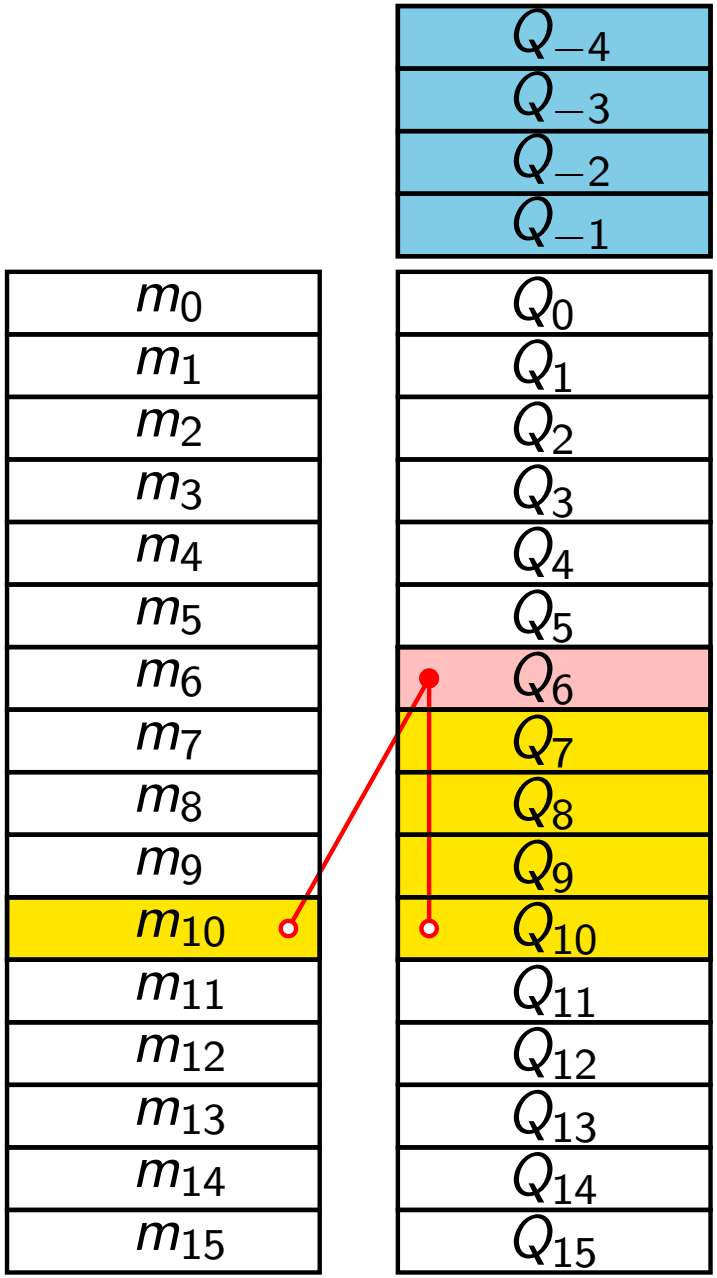


$$Q_i = (Q_{i-4} \boxplus \Phi_i \boxplus m_i \boxplus k_i) \lll s_i$$

$$Q_{i-4} = Q_i \ggg s_i \boxminus \Phi_i \boxminus m_i \boxminus k_i$$

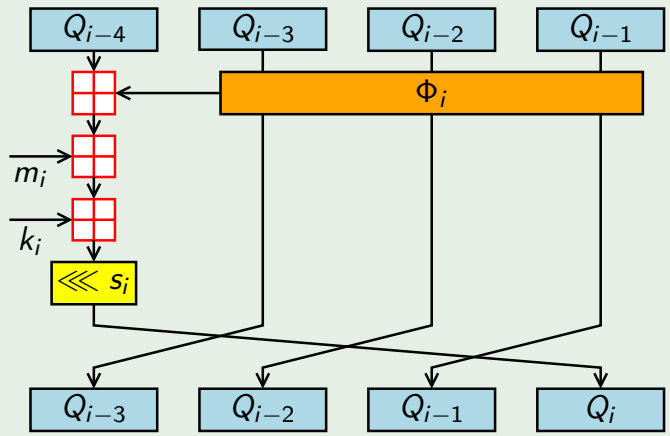
$$m_i = Q_i \ggg s_i \boxminus Q_{i-4} \boxminus \Phi_i \boxminus k_i$$

Collision search



✓

Basic equations

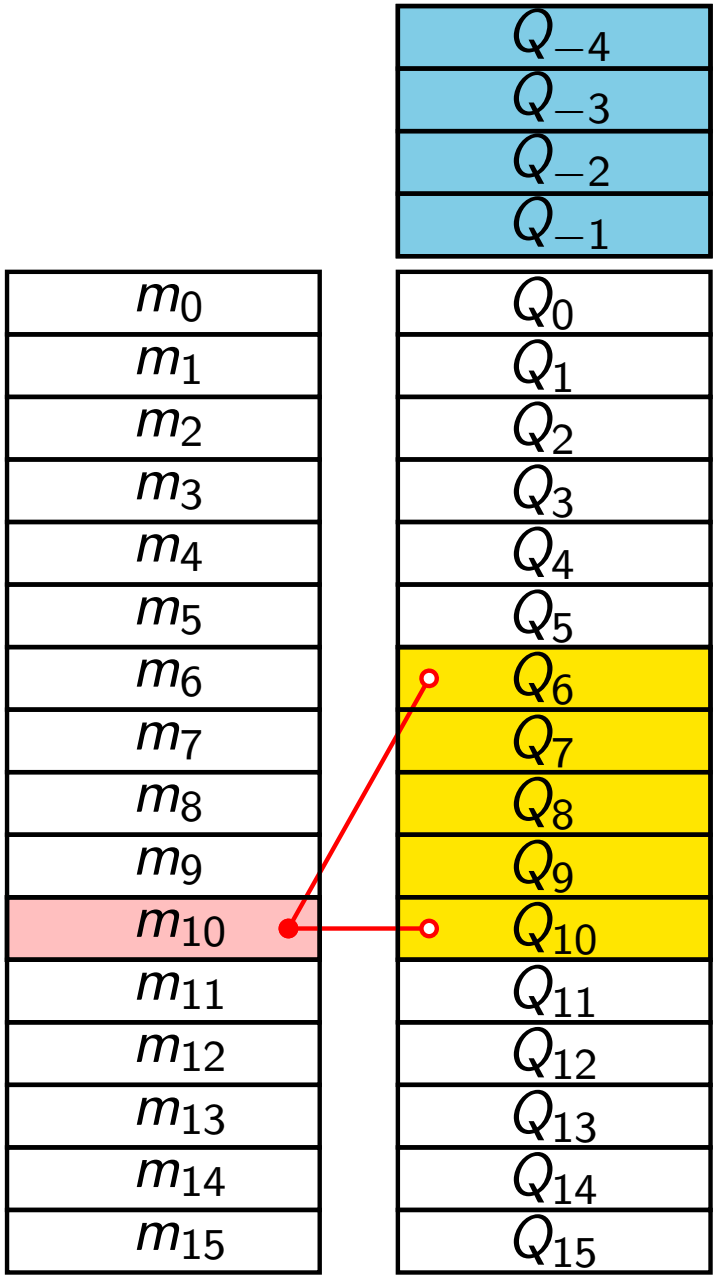


$$Q_i = (Q_{i-4} \boxplus \Phi_i \boxplus m_i \boxplus k_i) \lll s_i$$

$$Q_{i-4} = Q_i \ggg s_i \boxminus \Phi_i \boxminus m_i \boxminus k_i$$

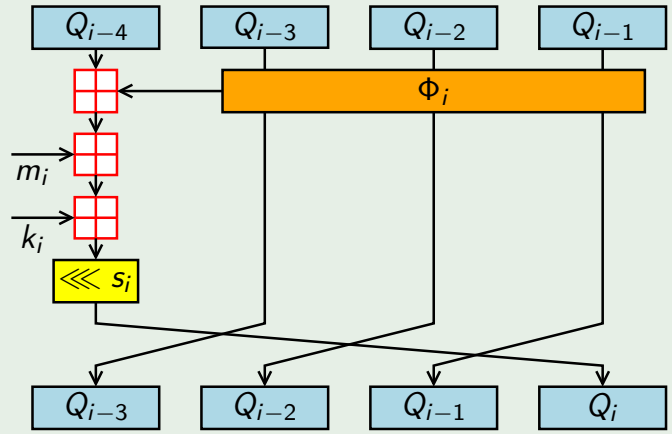
$$m_i = Q_i \ggg s_i \boxminus Q_{i-4} \boxminus \Phi_i \boxminus k_i$$

Collision search



✓

Basic equations



$$Q_i = (Q_{i-4} \boxplus \Phi_i \boxplus m_i \boxplus k_i) \lll s_i$$

$$Q_{i-4} = Q_i \ggg s_i \boxminus \Phi_i \boxminus m_i \boxminus k_i$$

$$m_i = Q_i \ggg s_i \boxminus Q_{i-4} \boxminus \Phi_i \boxminus k_i$$

How to satisfy conditions in the first round (Wang)

	Q_{-4}
	Q_{-3}
	Q_{-2}
	Q_{-1}
m_0	Q_0
m_1	Q_1
m_2	Q_2
m_3	Q_3
m_4	Q_4
m_5	Q_5
m_6	Q_6
m_7	Q_7
m_8	Q_8
m_9	Q_9
m_{10}	Q_{10}
m_{11}	Q_{11}
m_{12}	Q_{12}
m_{13}	Q_{13}
m_{14}	Q_{14}
m_{15}	Q_{15}

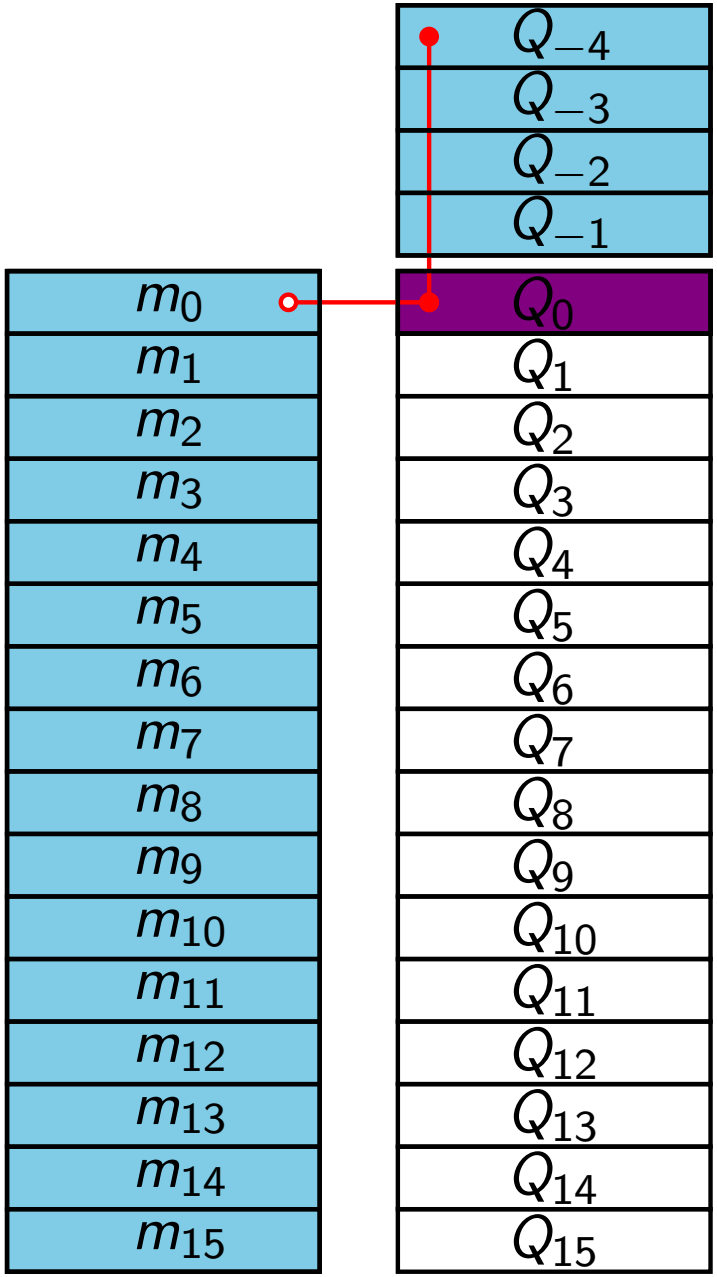
Message modification

- Pick a message.
- Compute Q_i.
- Modify Q_i; recompute m_i. Alternatively, modify m_i.

Remark

Each message modification depends on the previous ones.

How to satisfy conditions in the first round (Wang)



✓

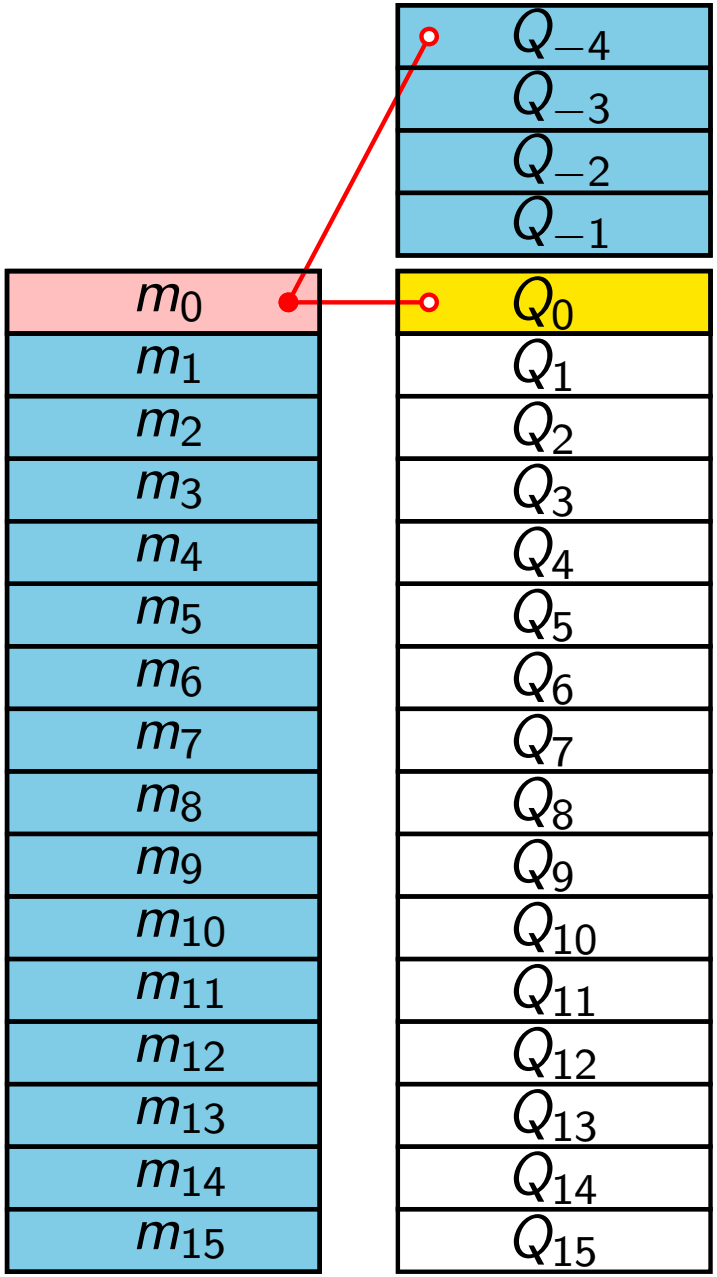
Message modification

- Pick a message.
- Compute Q_i.
- Modify Q_i; recompute m_i. Alternatively, modify m_i.

Remark

Each message modification depends on the previous ones.

How to satisfy conditions in the first round (Wang)



✓

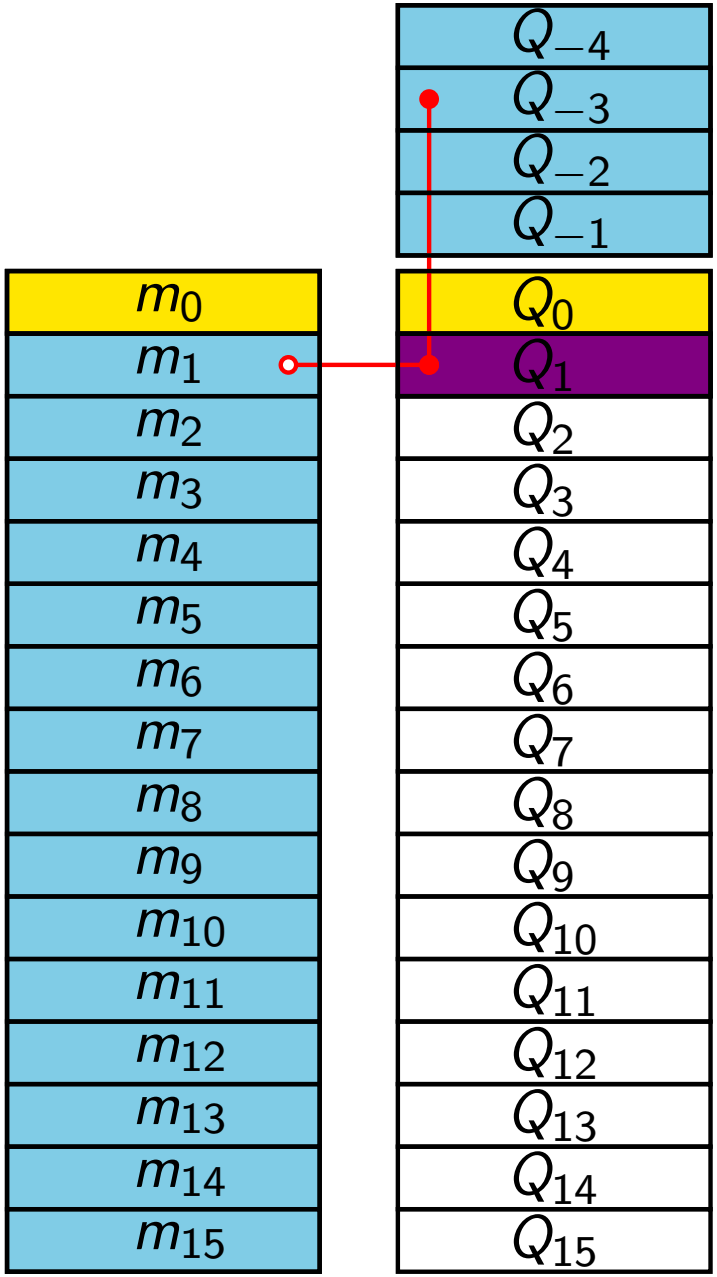
Message modification

- Pick a message.
- Compute Q_i.
- Modify Q_i; recompute m_i. Alternatively, modify m_i.

Remark

Each message modification depends on the previous ones.

How to satisfy conditions in the first round (Wang)



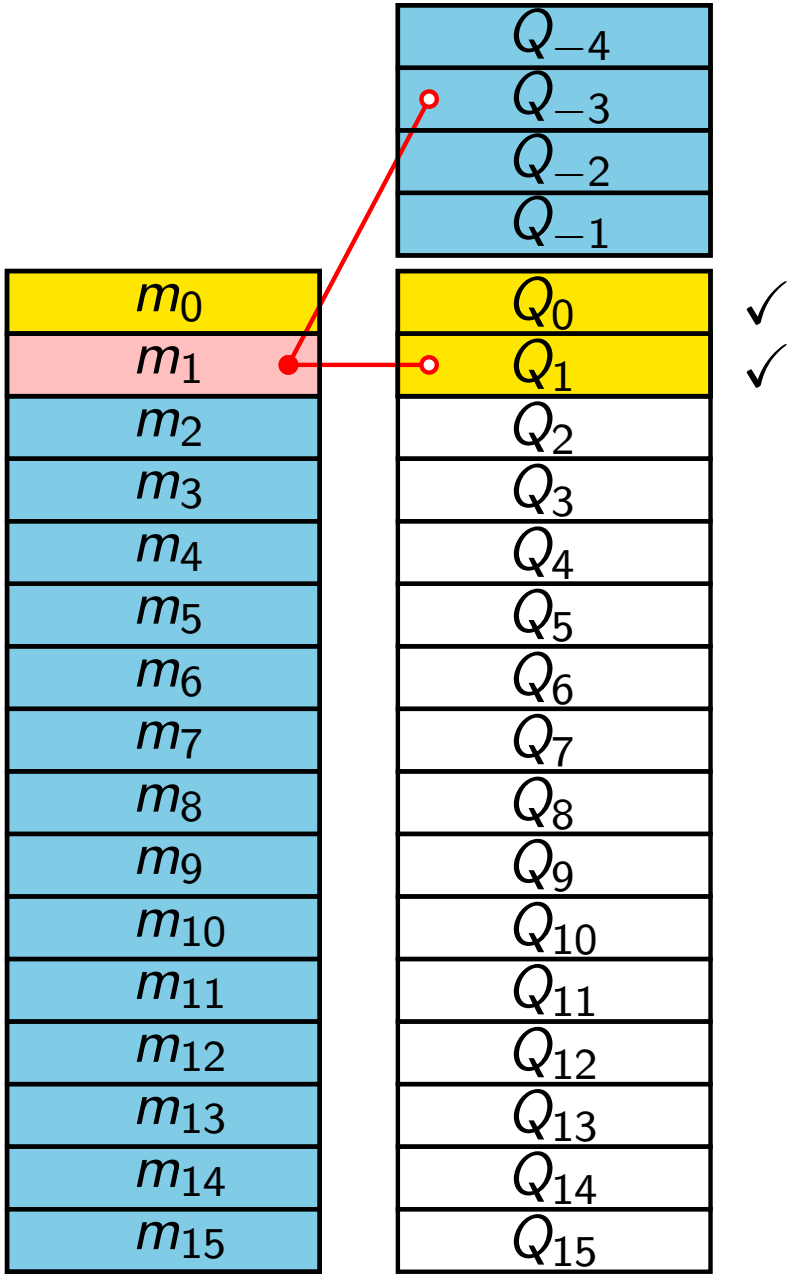
Message modification

- Pick a message.
- Compute Q_i.
- Modify Q_i; recompute m_i. Alternatively, modify m_i.

Remark

Each message modification depends on the previous ones.

How to satisfy conditions in the first round (Wang)



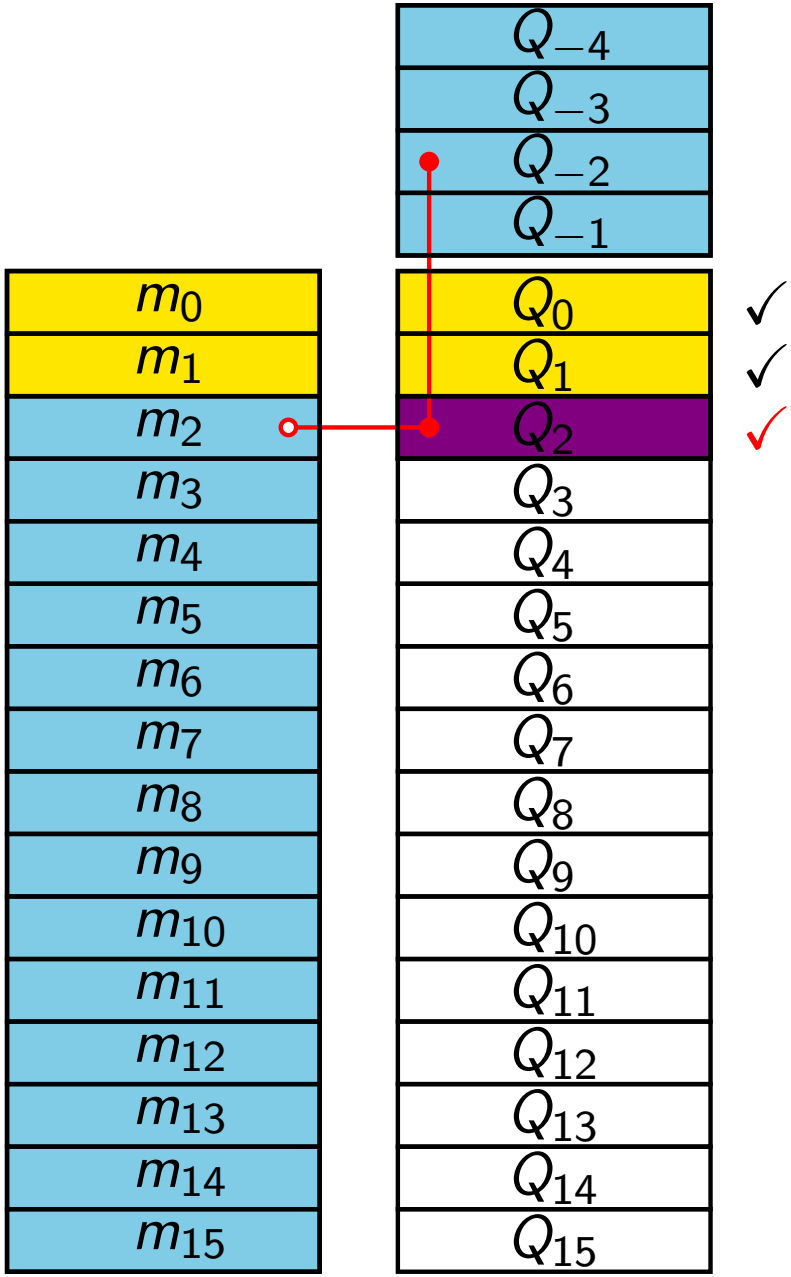
Message modification

- Pick a message.
- Compute Q_i.
- Modify Q_i; recompute m_i. Alternatively, modify m_i.

Remark

Each message modification depends on the previous ones.

How to satisfy conditions in the first round (Wang)



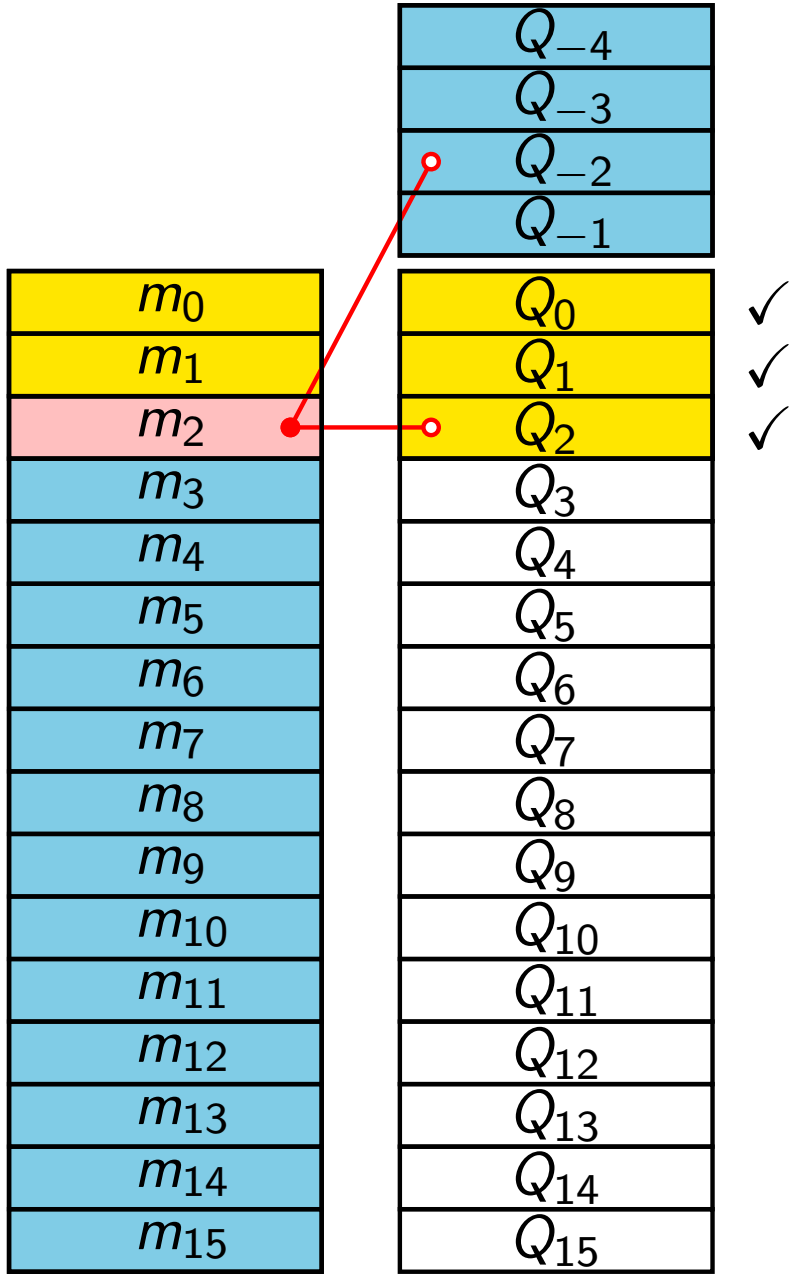
Message modification

- Pick a message.
- Compute Q_i.
- Modify Q_i; recompute m_i. Alternatively, modify m_i.

Remark

Each message modification depends on the previous ones.

How to satisfy conditions in the first round (Wang)



Message modification

- Pick a message.
- Compute Q_i.
- Modify Q_i; recompute m_i. Alternatively, modify m_i.

Remark

Each message modification depends on the previous ones.

How to satisfy conditions in the first round (Wang)

	Q_{-4}
	Q_{-3}
	Q_{-2}
	Q_{-1}
m_0	Q_0 ✓
m_1	Q_1 ✓
m_2	Q_2 ✓
m_3	Q_3 ✓
m_4	Q_4 ✓
m_5	Q_5 ✓
m_6	Q_6 ✓
m_7	Q_7 ✓
m_8	Q_8 ✓
m_9	Q_9 ✓
m_{10}	Q_{10} ✓
m_{11}	Q_{11} ✓
m_{12}	Q_{12} ✓
m_{13}	Q_{13} ✓
m_{14}	Q_{14} ✓
m_{15}	Q_{15} ✓

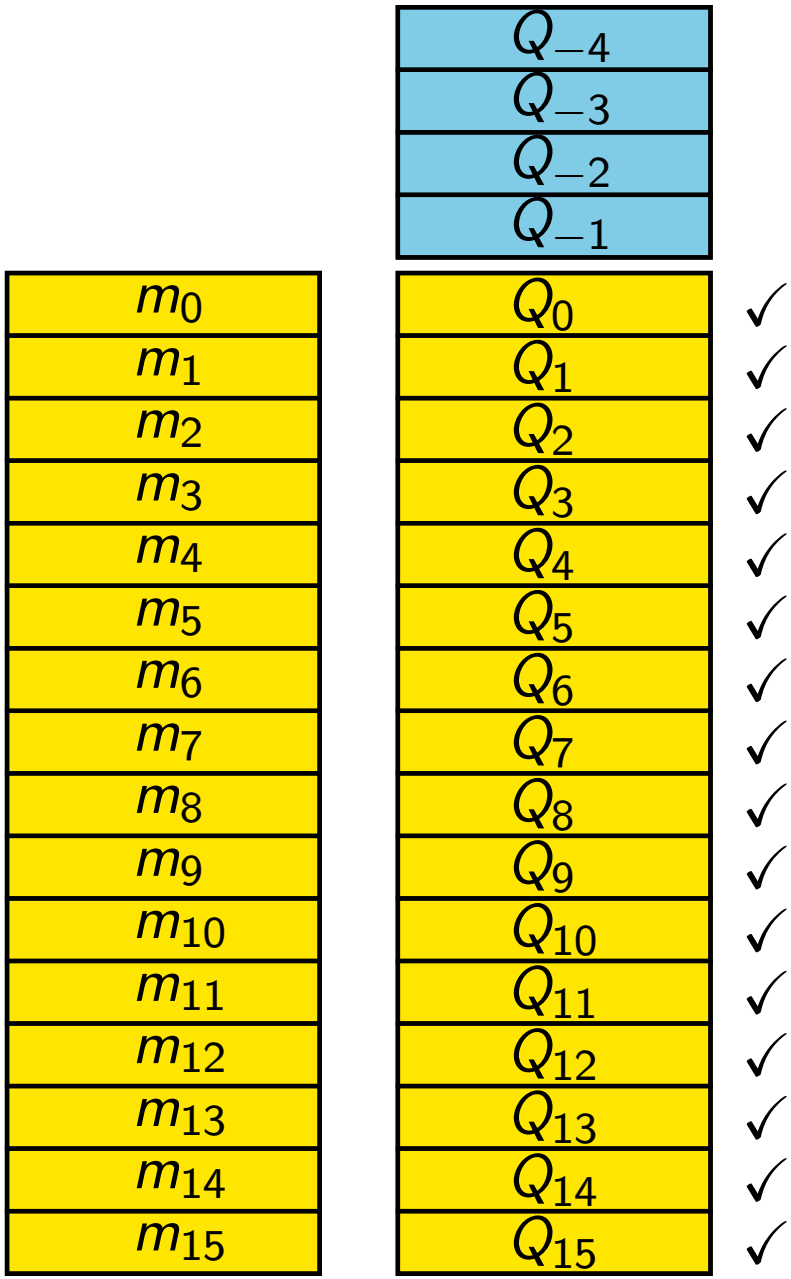
Message modification

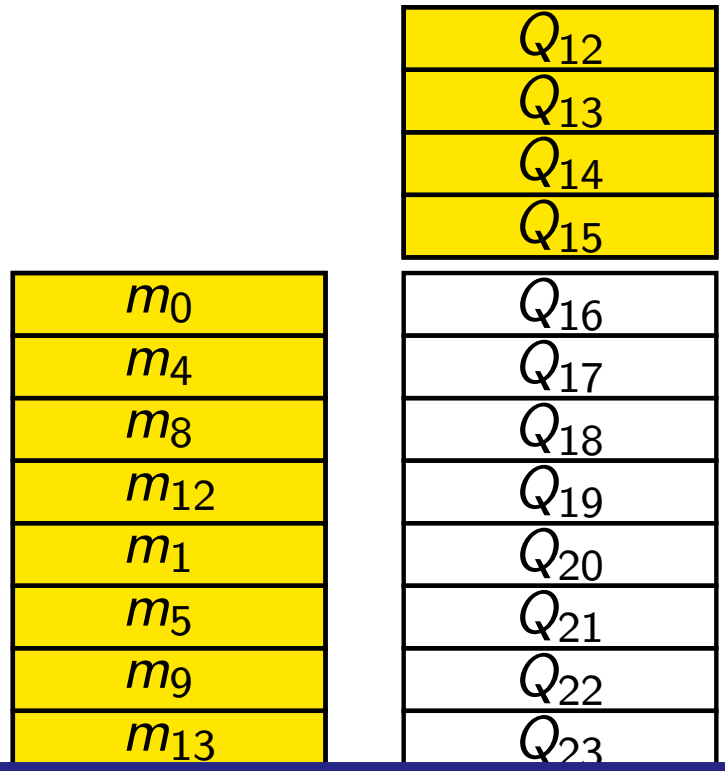
- Pick a message.
- Compute Q_i.
- Modify Q_i; recompute m_i. Alternatively, modify m_i.

Remark

Each message modification depends on the previous ones.

How to satisfy conditions in the 2nd round (Wang)

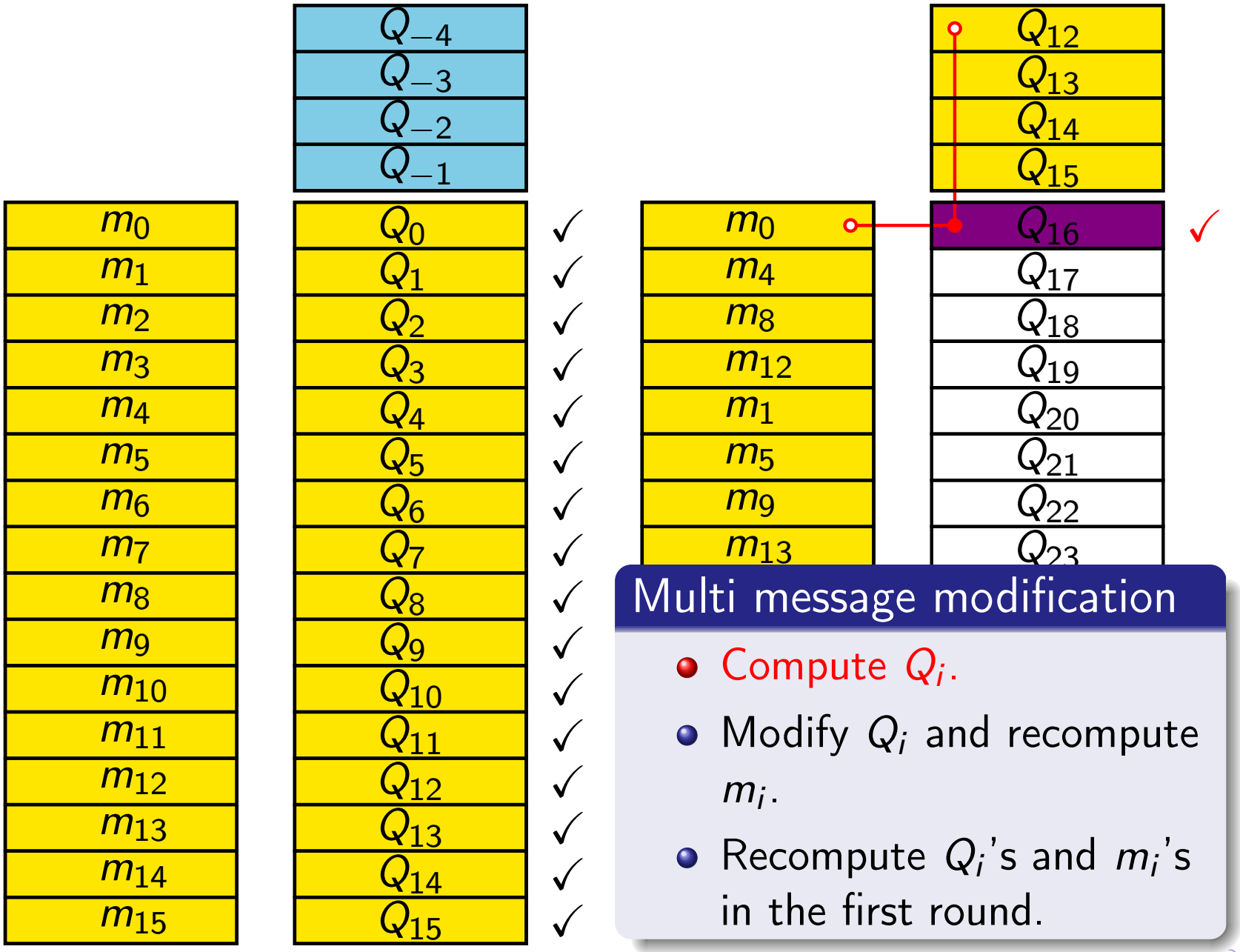




Multi message modification

- Compute Q_i.
- Modify Q_i and recompute m_i.
- Recompute Q_i's and m_i's in the first round.

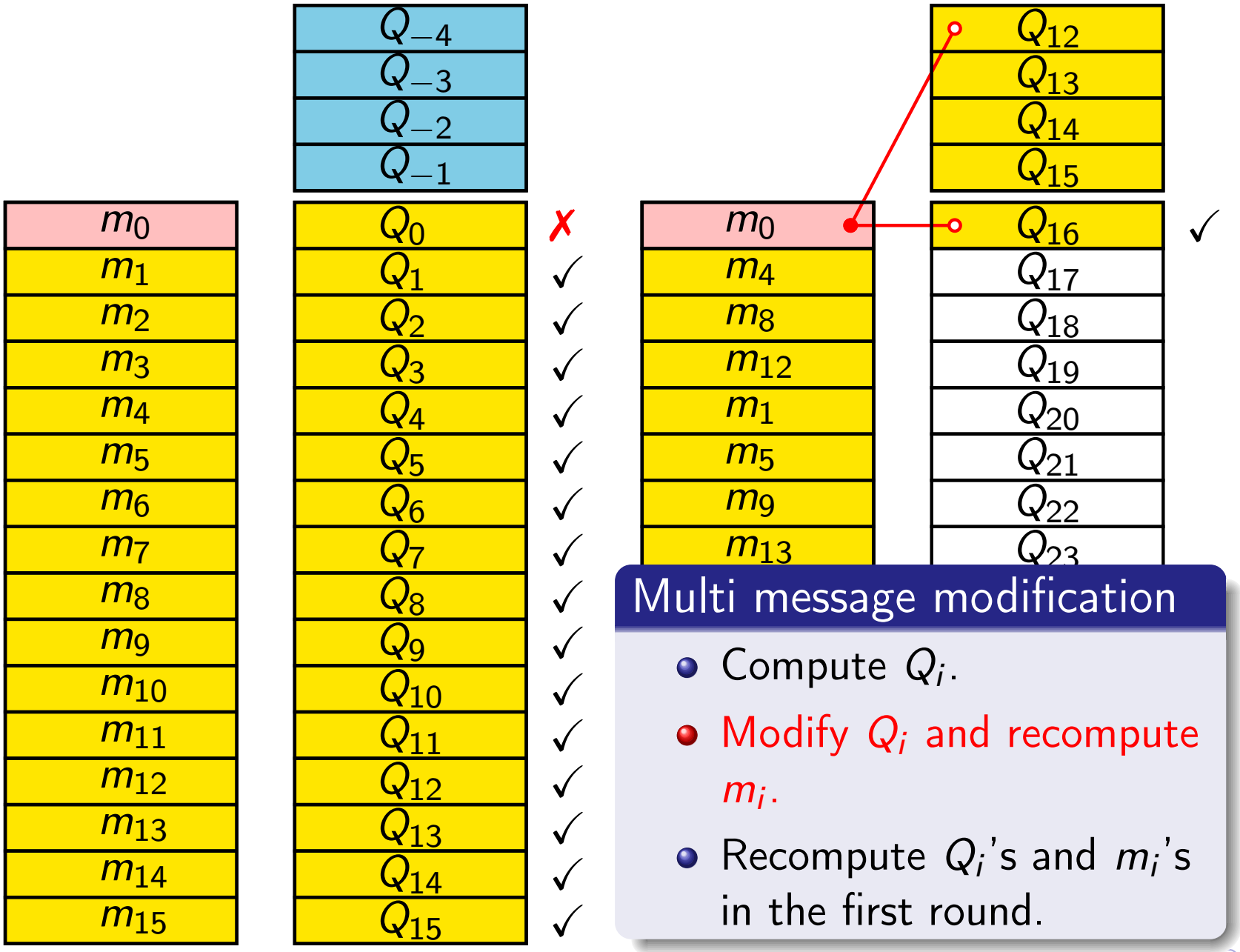
How to satisfy conditions in the 2nd round (Wang)



Multi message modification

- Compute Q_i.
- Modify Q_i and recompute m_i.
- Recompute Q_i's and m_i's in the first round.

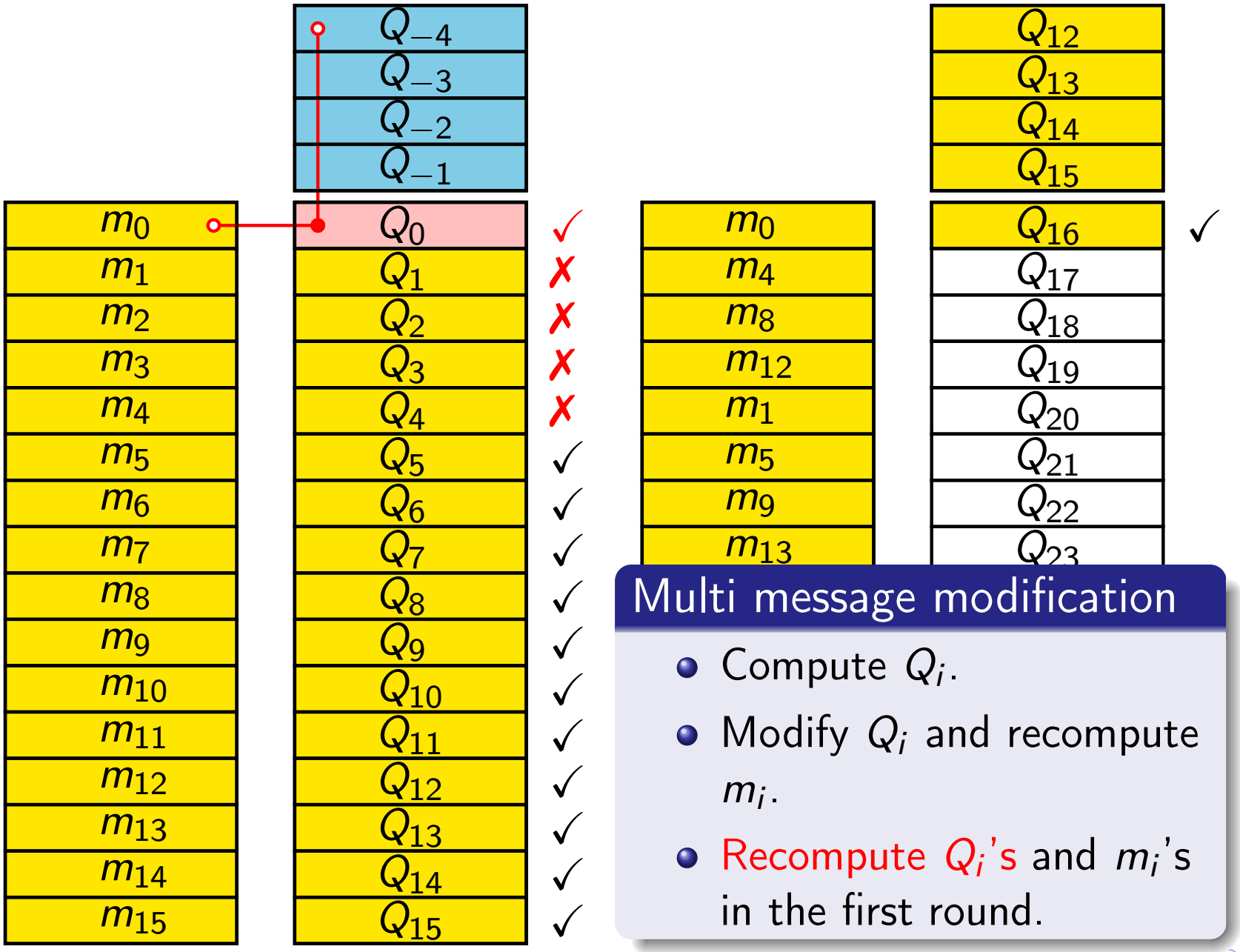
How to satisfy conditions in the 2nd round (Wang)



Multi message modification

- Compute Q_i.
- Modify Q_i and recompute m_i.
- Recompute Q_i's and m_i's in the first round.

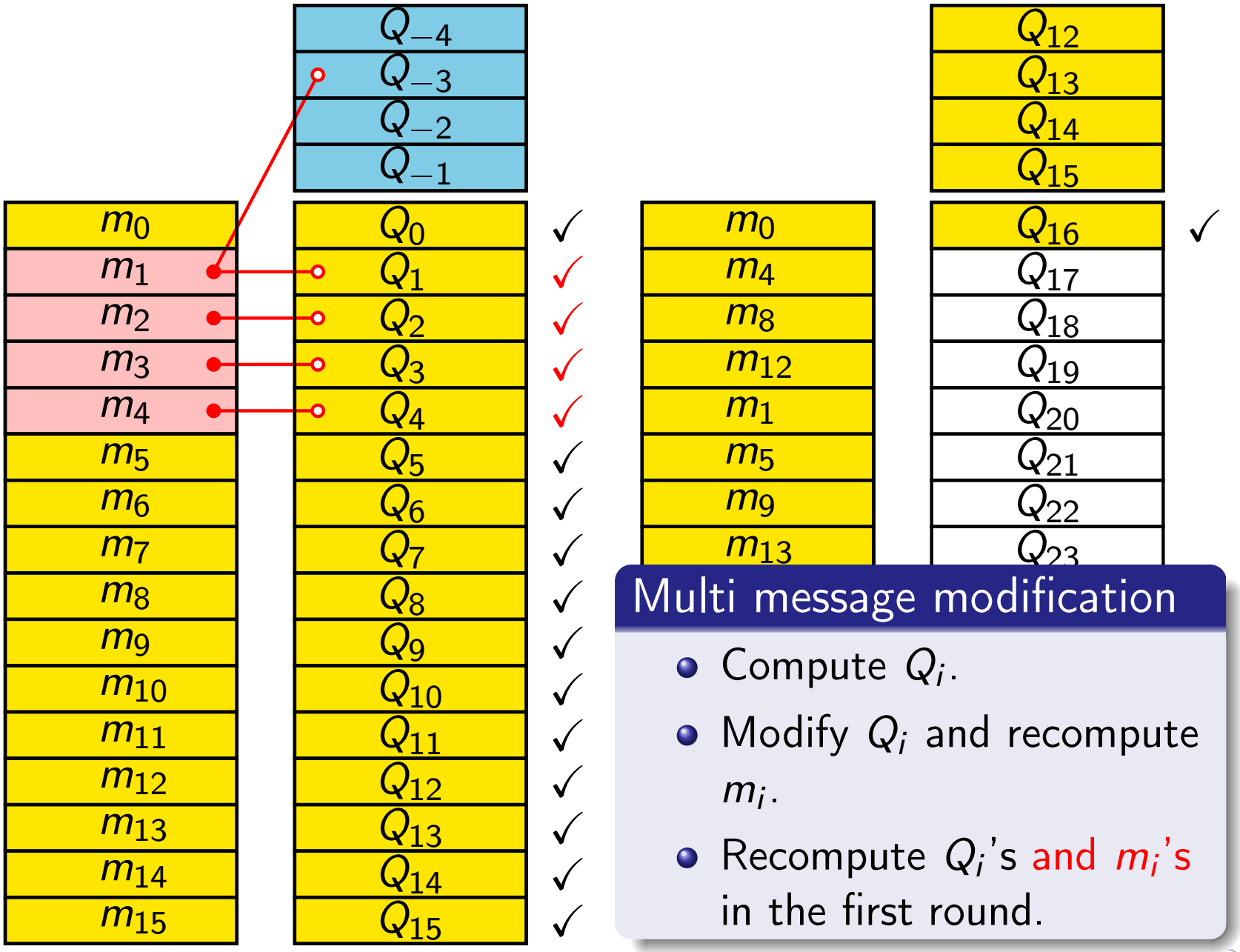
How to satisfy conditions in the 2nd round (Wang)



Multi message modification

- Compute Q_i.
- Modify Q_i and recompute m_i.
- Recompute Q_i's and m_i's in the first round.

How to satisfy conditions in the 2nd round (Wang)



Multi message modification

- Compute Q_i.
- Modify Q_i and recompute m_i.
- Recompute Q_i's and m_i's in the first round.

Revisiting Wang

Outline

- Message modification in the second round hard to find.
- Little message freedom due to message modifications.
- We propose a new way to satisfy the conditions.

How to satisfy conditions in the first round (New)

	Q_{-4}
	Q_{-3}
	Q_{-2}
	Q_{-1}
m_0	Q_0
m_1	Q_1
m_2	Q_2
m_3	Q_3
m_4	Q_4
m_5	Q_5
m_6	Q_6
m_7	Q_7
m_8	Q_8
m_9	Q_9
m_{10}	Q_{10}
m_{11}	Q_{11}
m_{12}	Q_{12}
m_{13}	Q_{13}
m_{14}	Q_{14}
m_{15}	Q_{15}

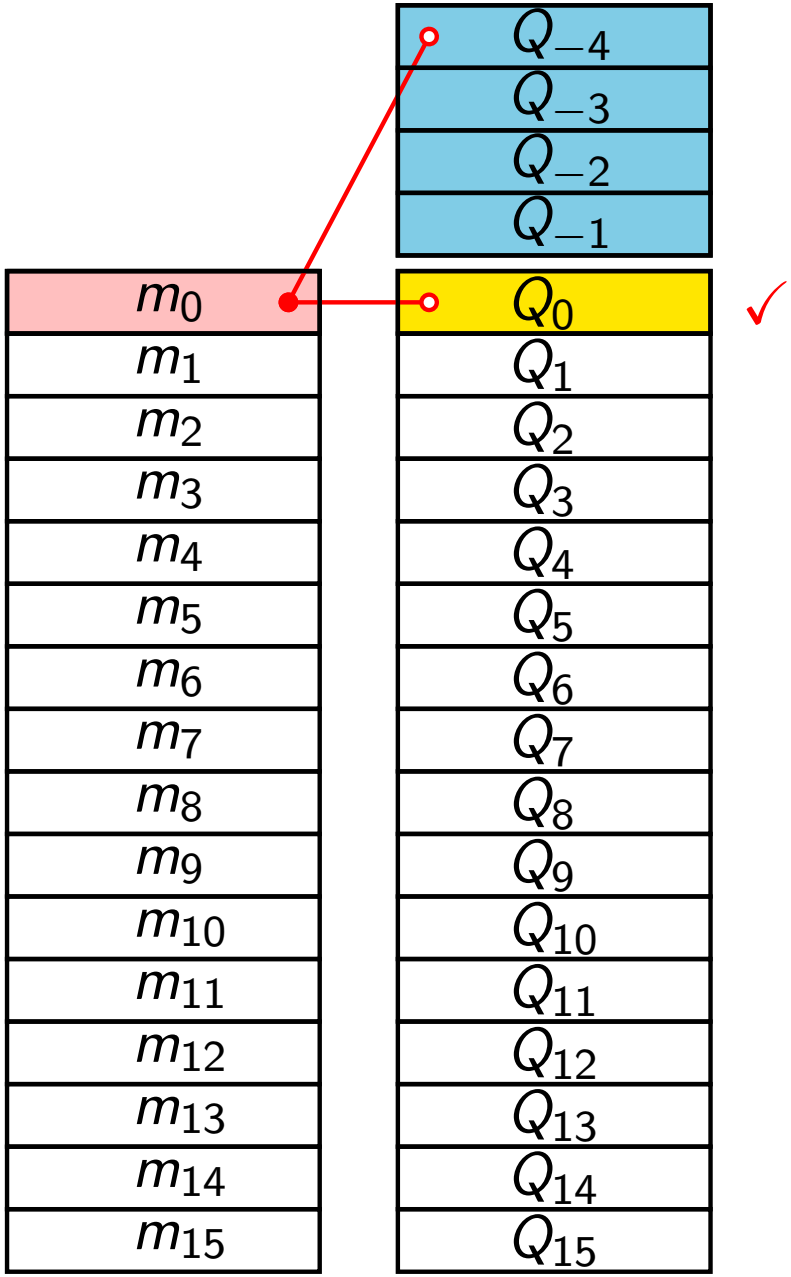
Wang revisited

- Choose Q_i.
- Compute m_i.

Remark

The Q_i's in the first round can be chosen in any order.

How to satisfy conditions in the first round (New)



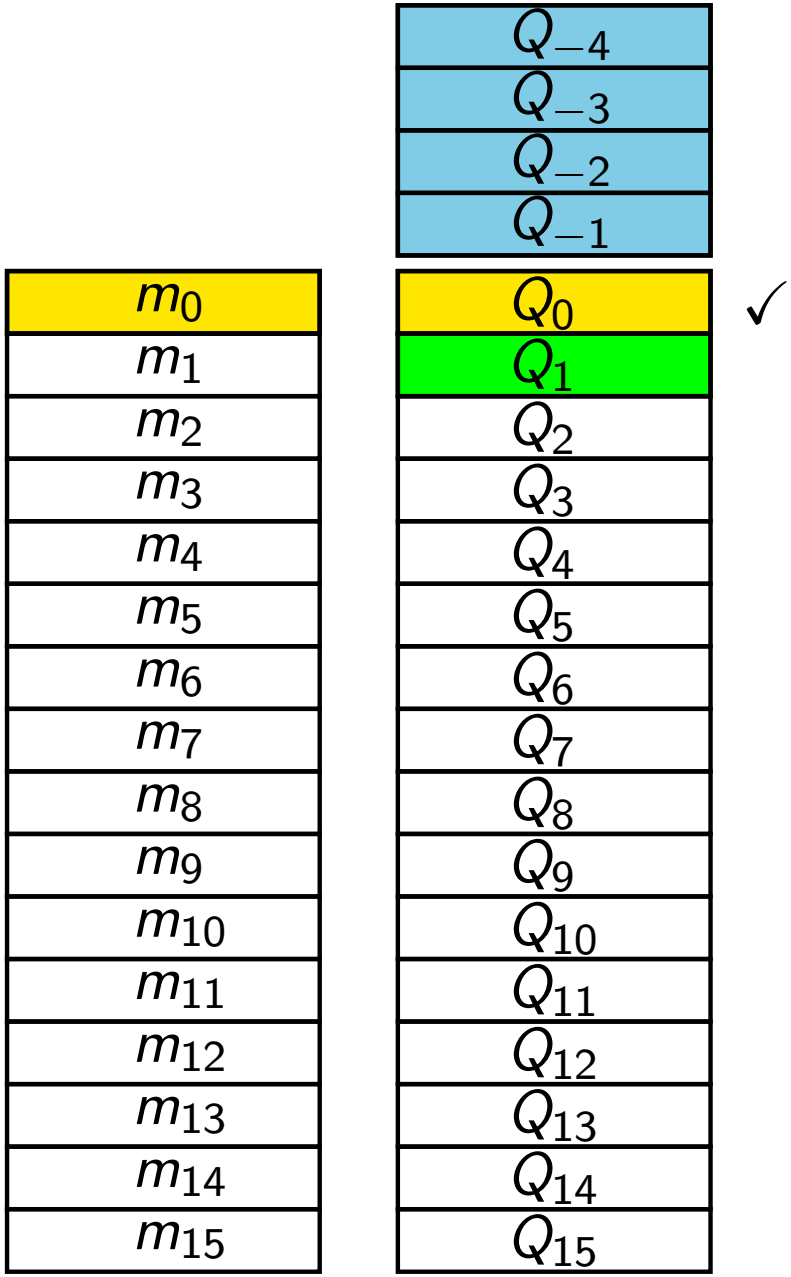
Wang revisited

- Choose Q_i.
- Compute m_i.

Remark

The Q_i's in the first round can be chosen in any order.

How to satisfy conditions in the first round (New)



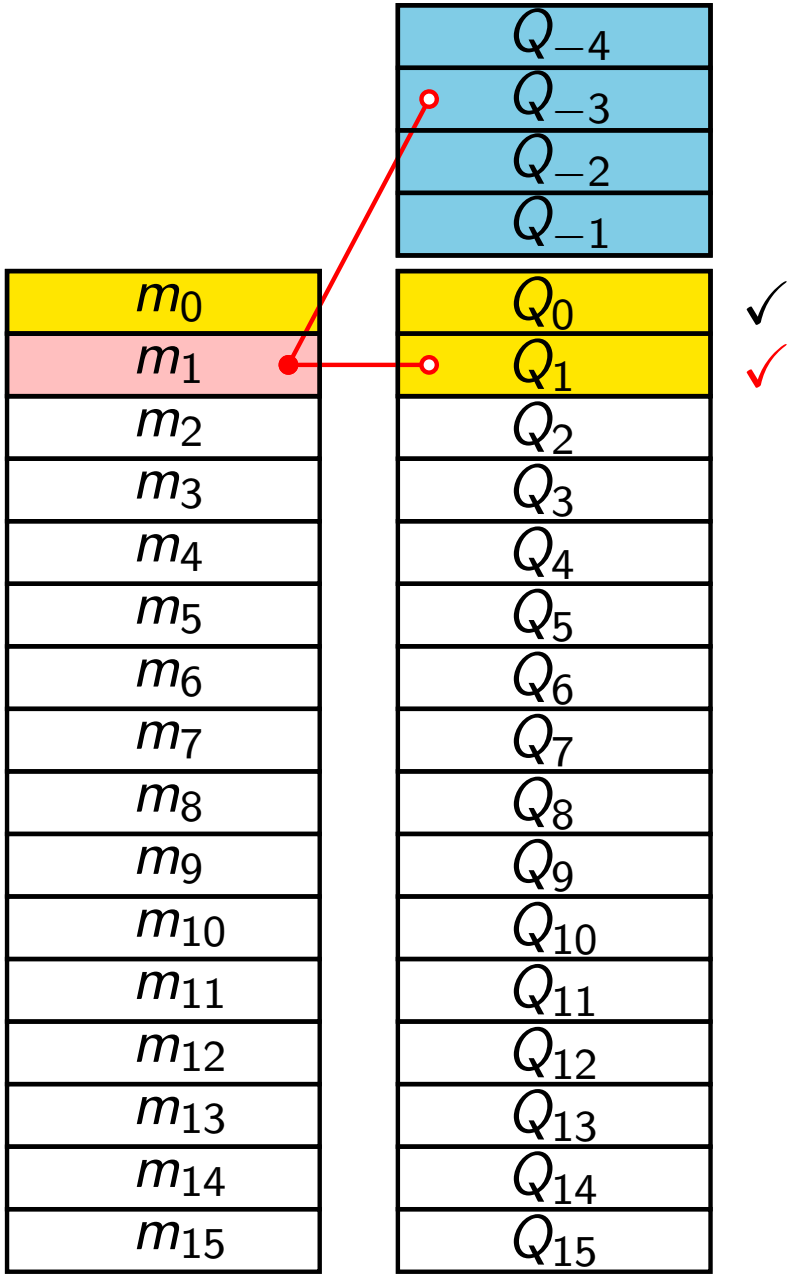
Wang revisited

- Choose Q_i.
- Compute m_i.

Remark

The Q_i's in the first round can be chosen in any order.

How to satisfy conditions in the first round (New)



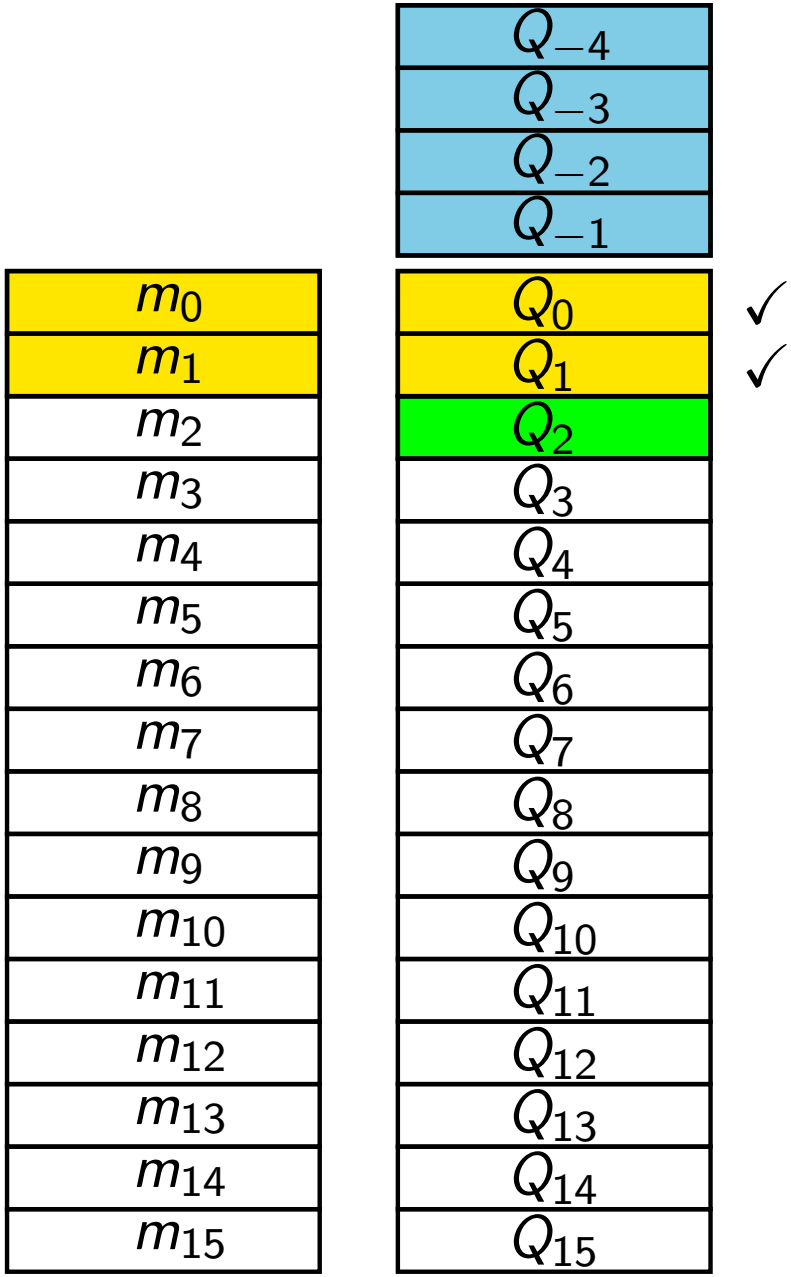
Wang revisited

- Choose Q_i.
- Compute m_i.

Remark

The Q_i's in the first round can be chosen in any order.

How to satisfy conditions in the first round (New)



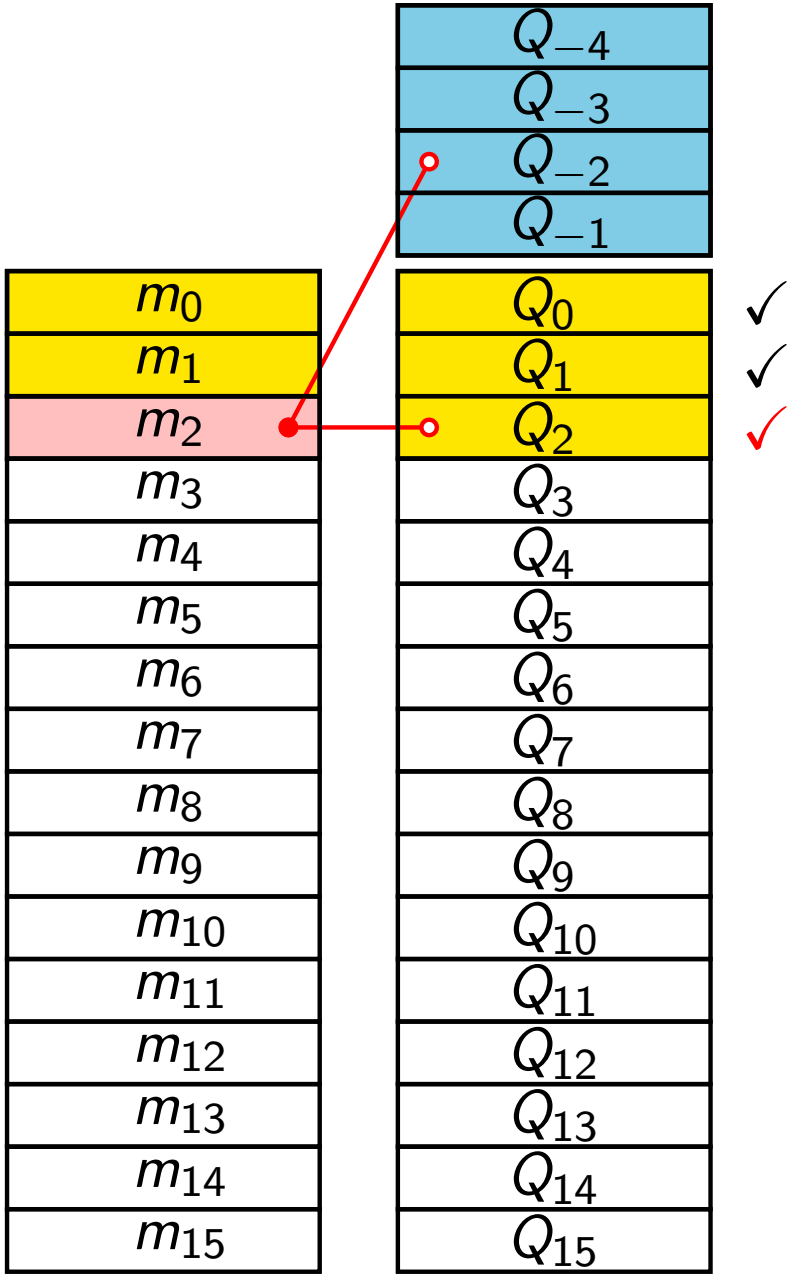
Wang revisited

- Choose Q_i.
- Compute m_i.

Remark

The Q_i's in the first round can be chosen in any order.

How to satisfy conditions in the first round (New)



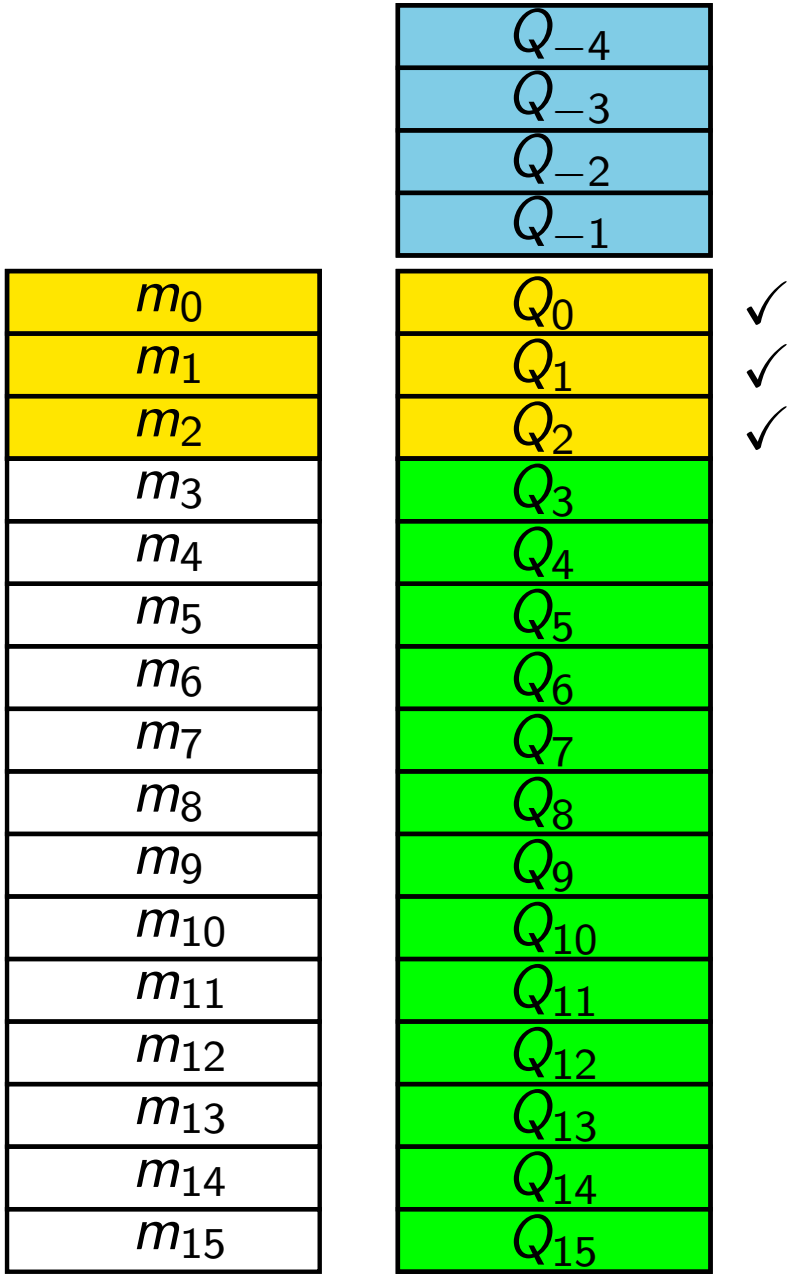
Wang revisited

- Choose Q_i.
- Compute m_i.

Remark

The Q_i's in the first round can be chosen in any order.

How to satisfy conditions in the first round (New)



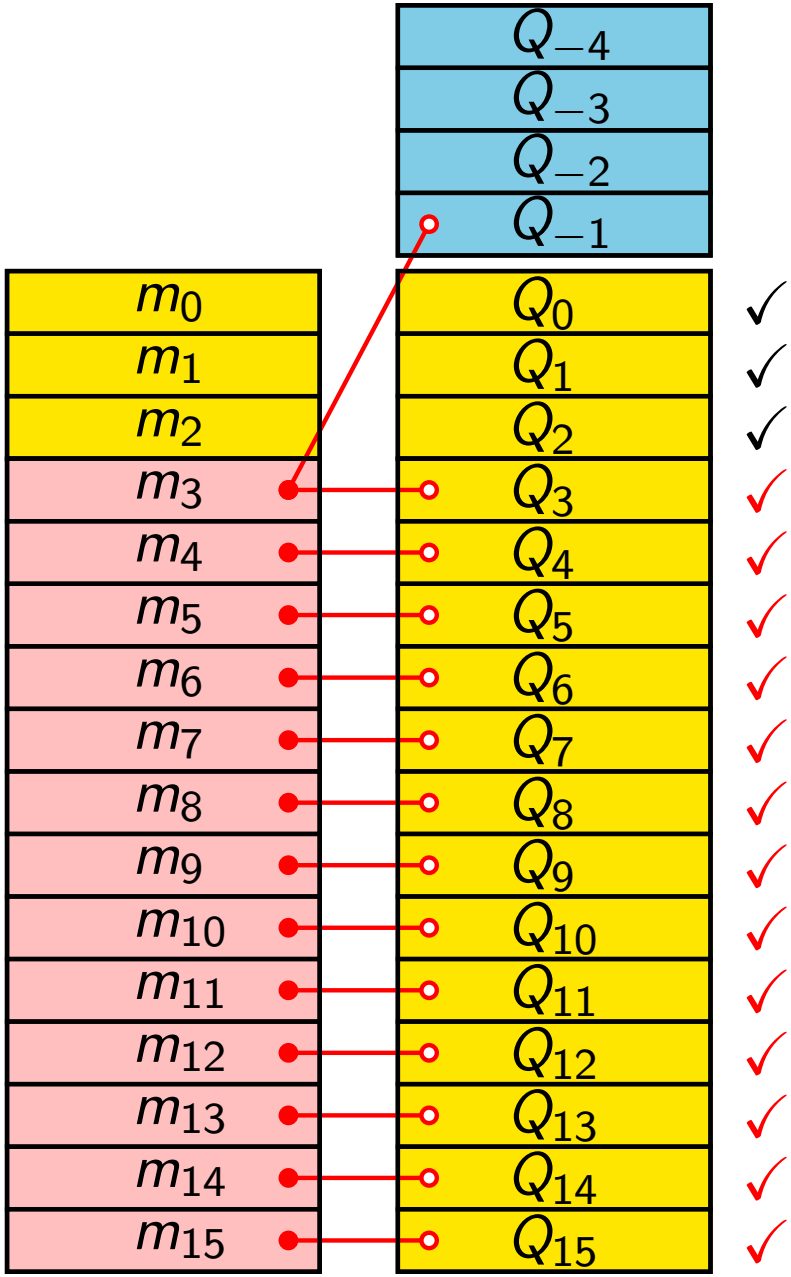
Wang revisited

- Choose Q_i.
- Compute m_i.

Remark

The Q_i's in the first round can be chosen in any order.

How to satisfy conditions in the first round (New)



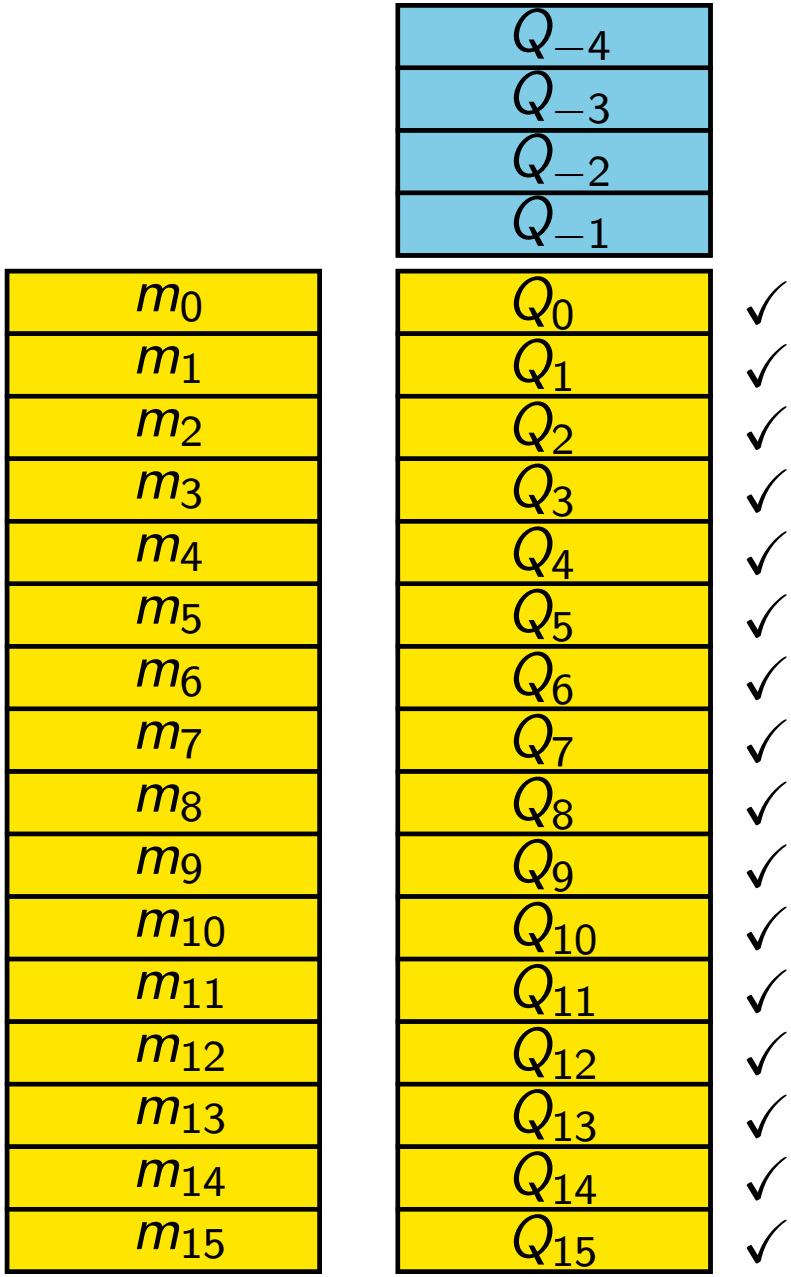
Wang revisited

- Choose Q_i.
- Compute m_i.

Remark

The Q_i's in the first round can be chosen in any order.

How to satisfy conditions in the first round (New)



Wang revisited

- Choose Q_i.
- Compute m_i.

Remark

The Q_i's in the first round can be chosen in any order.

How to satisfy conditions in the 2nd round (New)

	Q_{-4}
	Q_{-3}
	Q_{-2}
	Q_{-1}
m_0	Q_0
m_1	Q_1
m_2	Q_2
m_3	Q_3
m_4	Q_4
m_5	Q_5
m_6	Q_6
m_7	Q_7
m_8	Q_8
m_9	Q_9
m_{10}	Q_{10}
m_{11}	Q_{11}
m_{12}	Q_{12}
m_{13}	Q_{13}
m_{14}	Q_{14}
m_{15}	Q_{15}

	Q_{12}
	Q_{13}
	Q_{14}
	Q_{15}
m_0	Q_{16}
m_4	Q_{17}
m_8	Q_{18}
m_{12}	Q_{19}
m_1	Q_{20}
m_5	Q_{21}
m_9	Q_{22}
m_{13}	Q_{23}

New method

- Choose the end of the first round.
- Choose m_i to satisfy both rounds.
- Fill the first round.

How to satisfy conditions in the 2nd round (New)

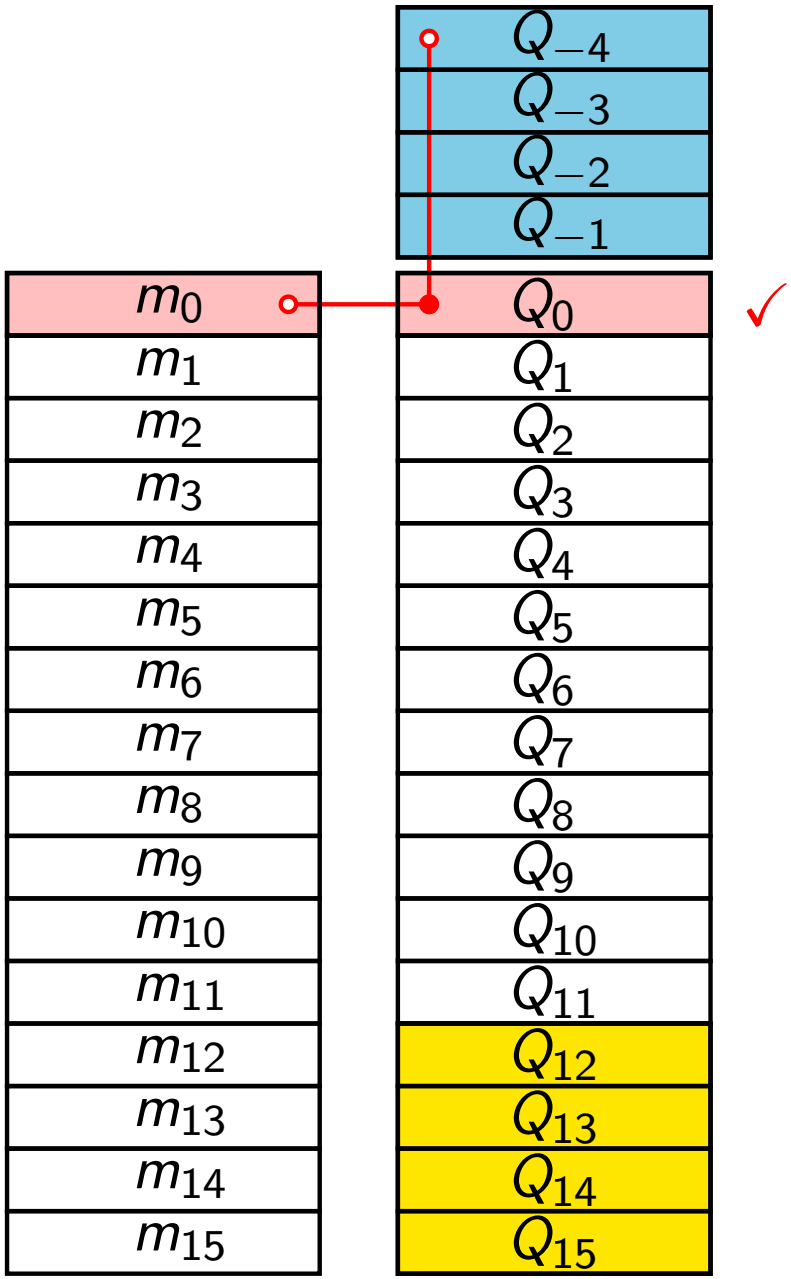
	Q_{-4}
	Q_{-3}
	Q_{-2}
	Q_{-1}
m_0	Q_0
m_1	Q_1
m_2	Q_2
m_3	Q_3
m_4	Q_4
m_5	Q_5
m_6	Q_6
m_7	Q_7
m_8	Q_8
m_9	Q_9
m_{10}	Q_{10}
m_{11}	Q_{11}
m_{12}	Q_{12}
m_{13}	Q_{13}
m_{14}	Q_{14}
m_{15}	Q_{15}

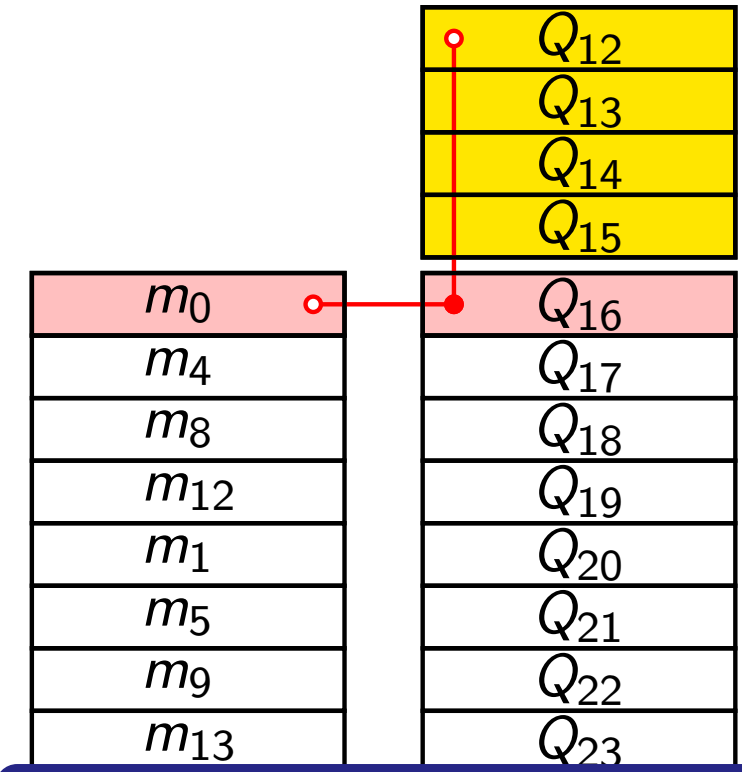
	Q_{12}
	Q_{13}
	Q_{14}
	Q_{15}
m_0	Q_{16}
m_4	Q_{17}
m_8	Q_{18}
m_{12}	Q_{19}
m_1	Q_{20}
m_5	Q_{21}
m_9	Q_{22}
m_{13}	Q_{23}

New method

- Choose the end of the first round.
- Choose m_i to satisfy both rounds.
- Fill the first round.

How to satisfy conditions in the 2nd round (New)





New method

- Choose the end of the first round.
- Choose m_i to satisfy both rounds.
- Fill the first round.

How to satisfy conditions in the 2nd round (New)

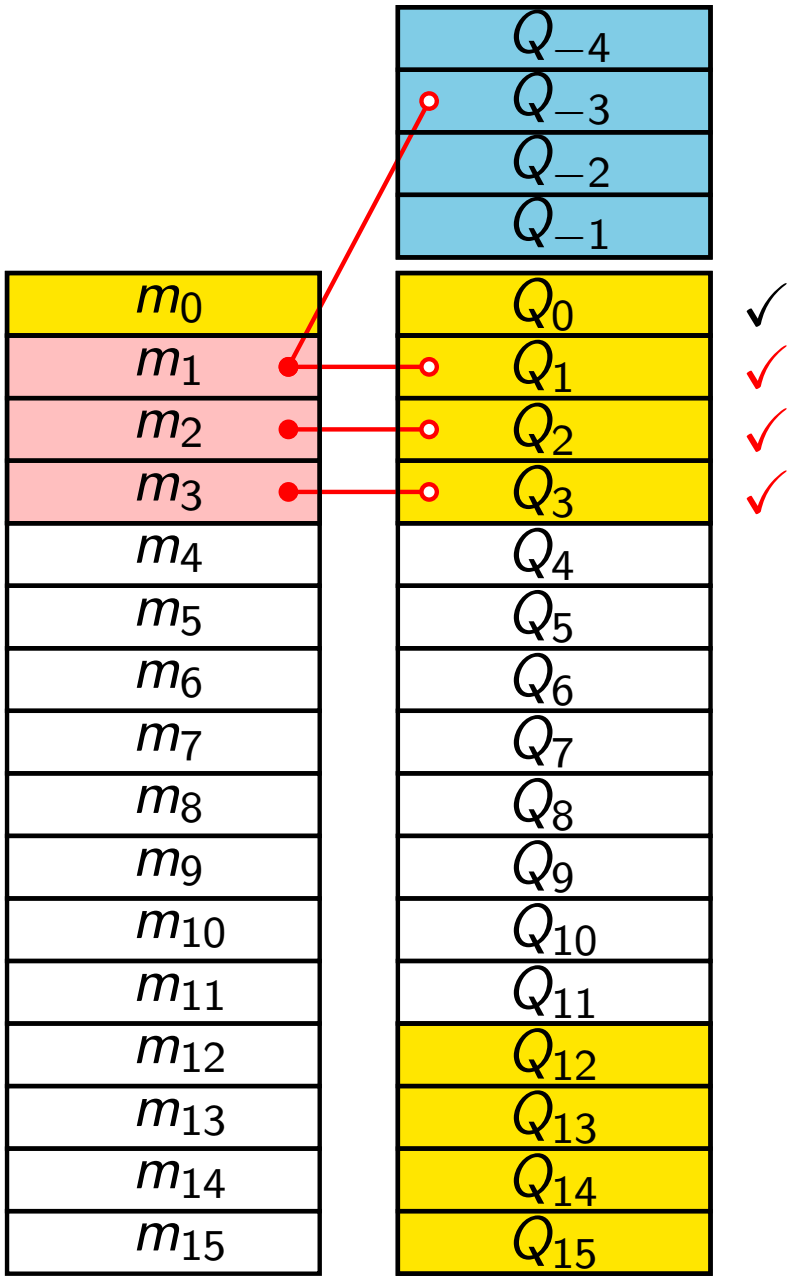
	Q_{-4}
	Q_{-3}
	Q_{-2}
	Q_{-1}
m_0	Q_0 ✓
m_1	Q_1
m_2	Q_2
m_3	Q_3
m_4	Q_4
m_5	Q_5
m_6	Q_6
m_7	Q_7
m_8	Q_8
m_9	Q_9
m_{10}	Q_{10}
m_{11}	Q_{11}
m_{12}	Q_{12}
m_{13}	Q_{13}
m_{14}	Q_{14}
m_{15}	Q_{15}

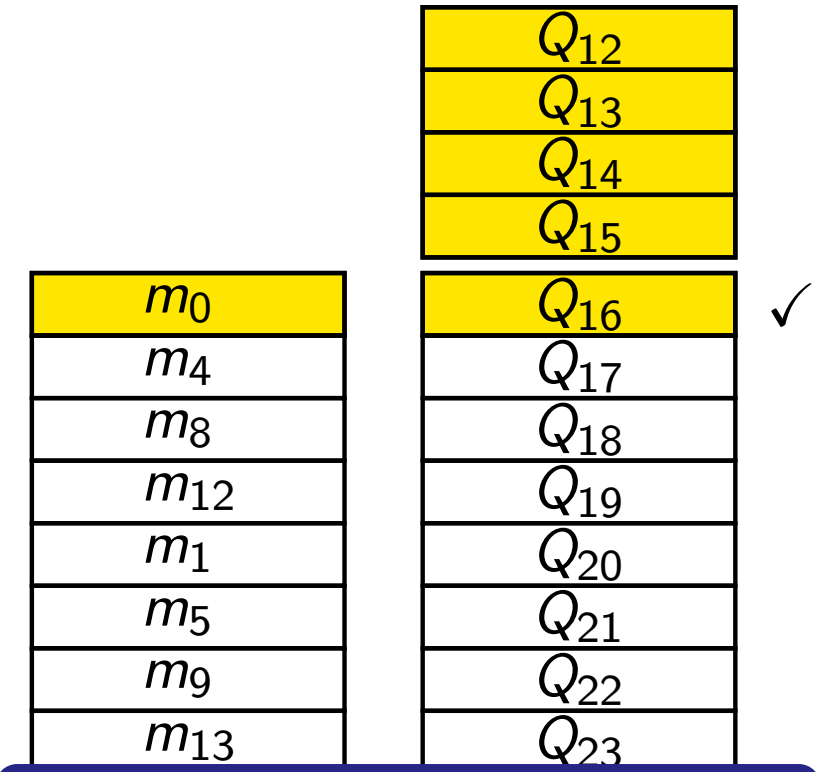
	Q_{12}
	Q_{13}
	Q_{14}
	Q_{15}
m_0	Q_{16} ✓
m_4	Q_{17}
m_8	Q_{18}
m_{12}	Q_{19}
m_1	Q_{20}
m_5	Q_{21}
m_9	Q_{22}
m_{13}	Q_{23}

New method

- Choose the end of the first round.
- Choose m_i to satisfy both rounds.
- Fill the first round.

How to satisfy conditions in the 2nd round (New)

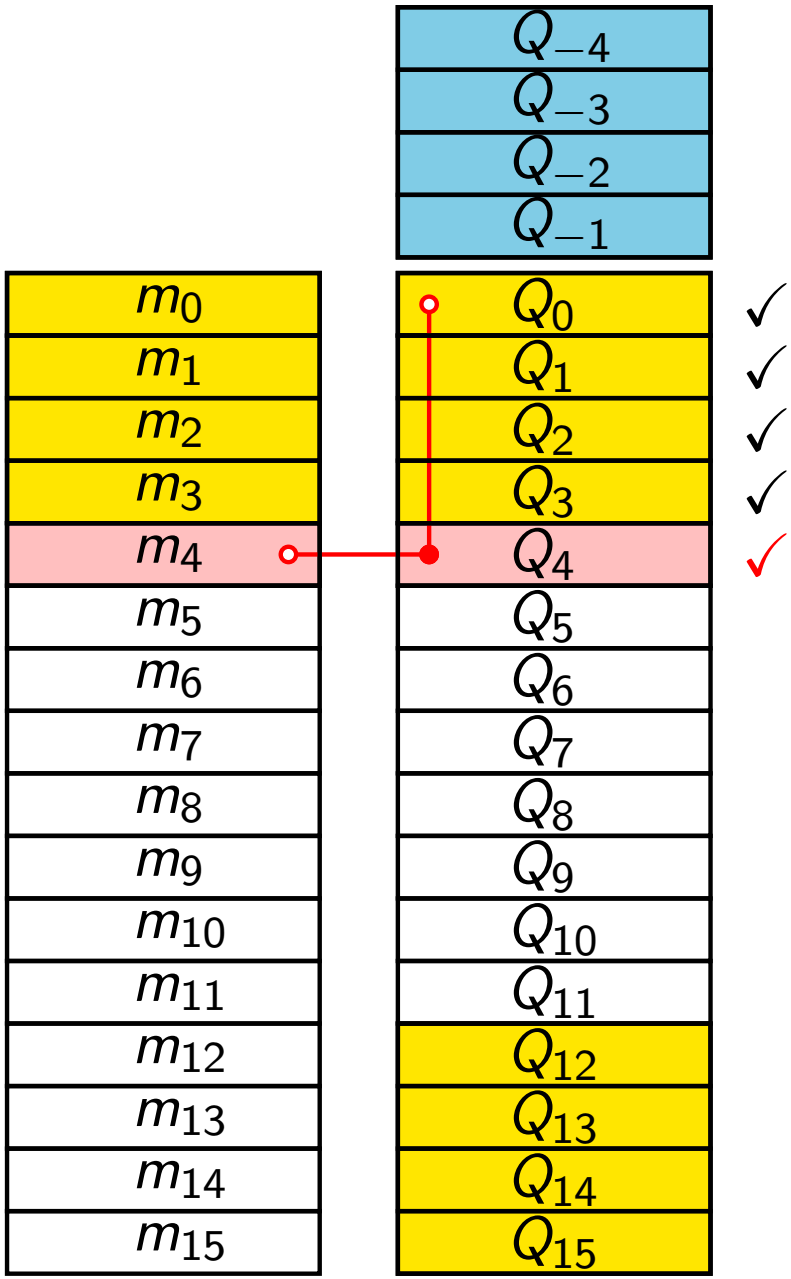


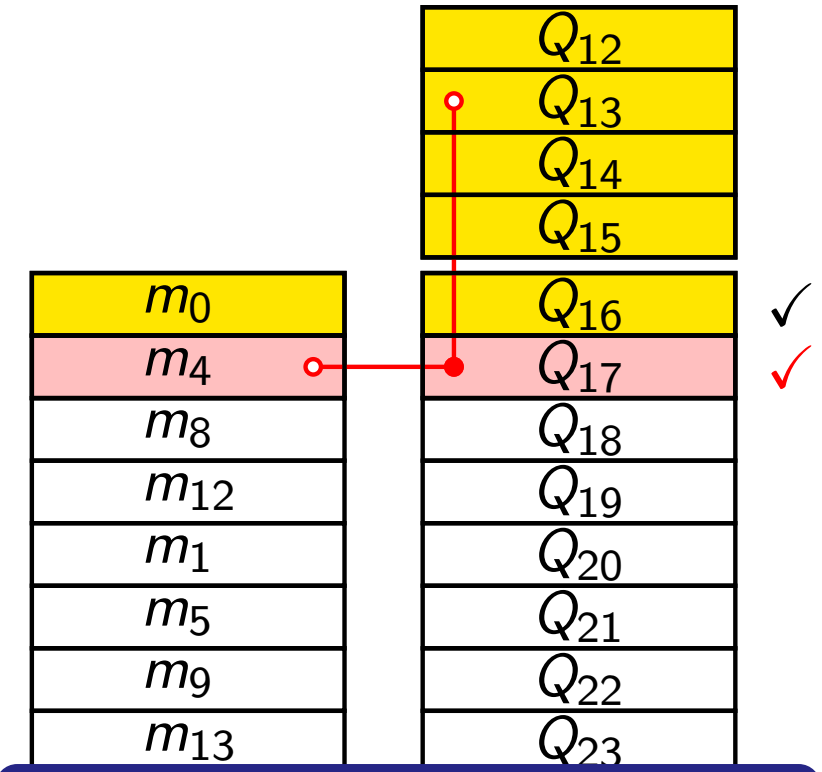


New method

- Choose the end of the first round.
- Choose m_i to satisfy both rounds.
- Fill the first round.

How to satisfy conditions in the 2nd round (New)

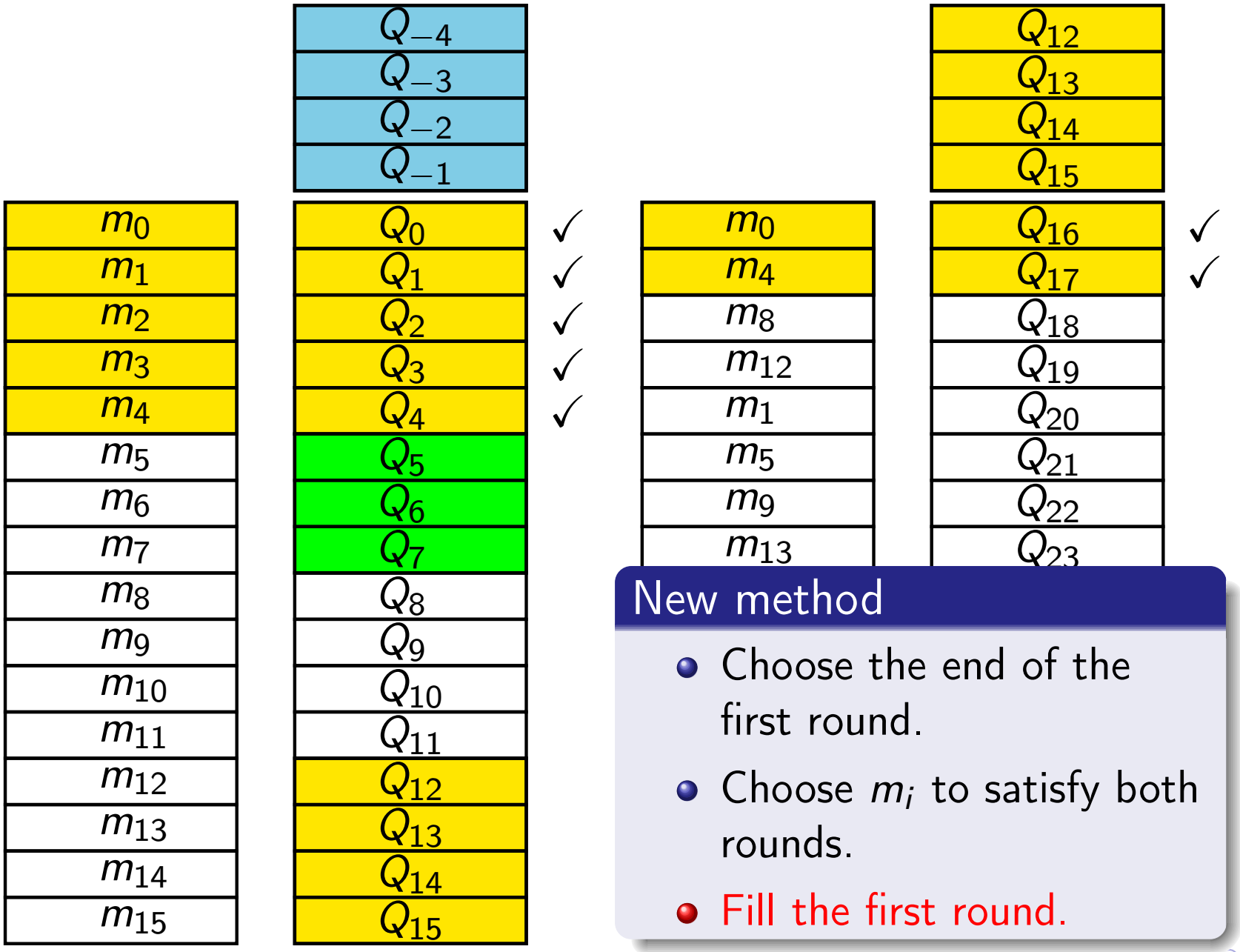




New method

- Choose the end of the first round.
- Choose m_i to satisfy both rounds.
- Fill the first round.

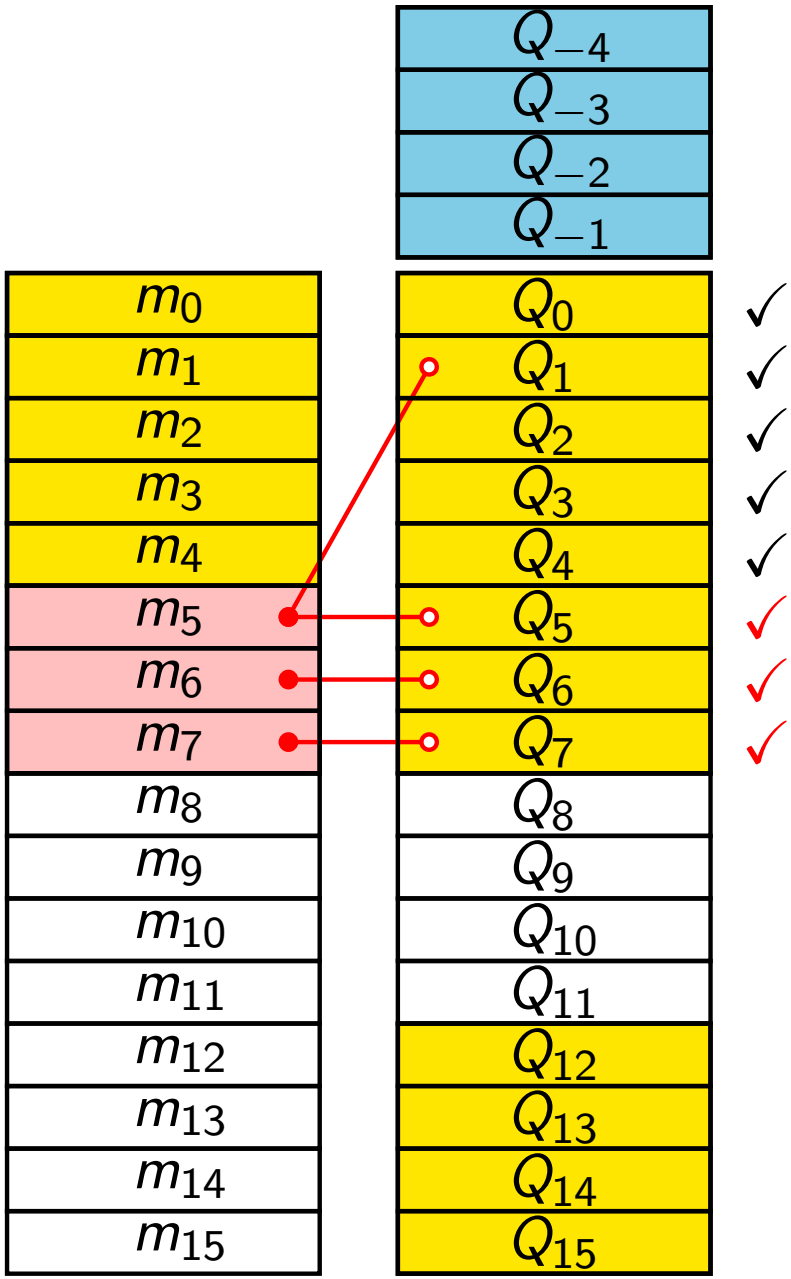
How to satisfy conditions in the 2nd round (New)

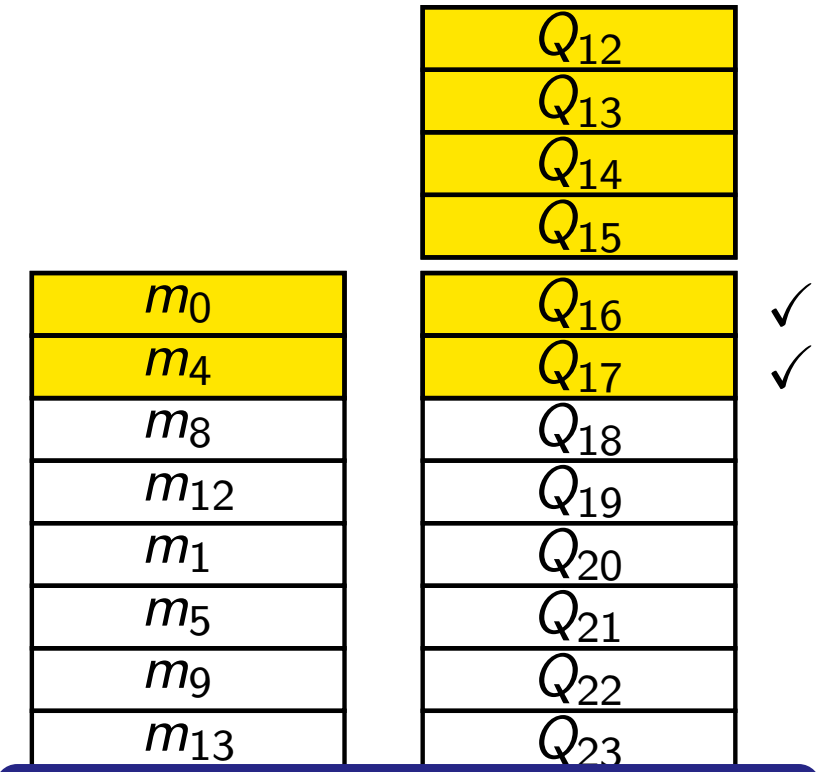


New method

- Choose the end of the first round.
- Choose m_i to satisfy both rounds.
- Fill the first round.

How to satisfy conditions in the 2nd round (New)

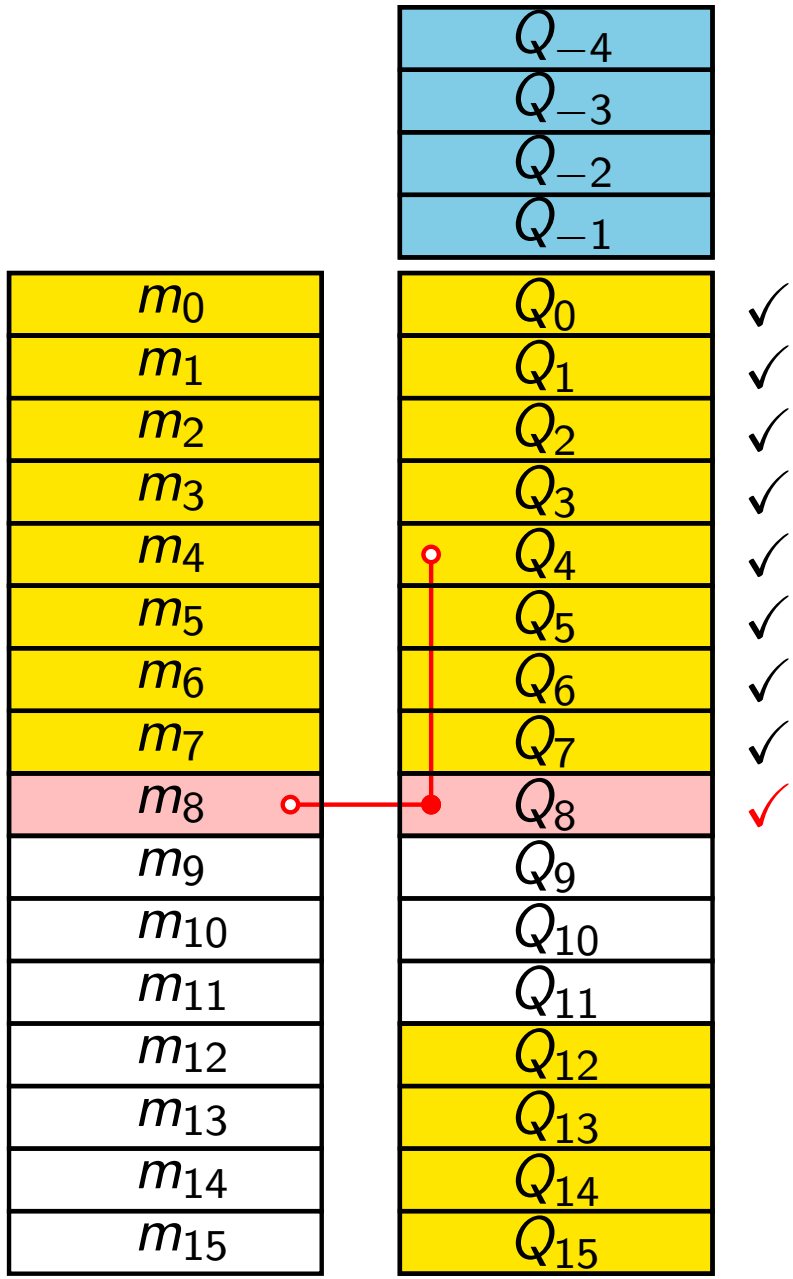


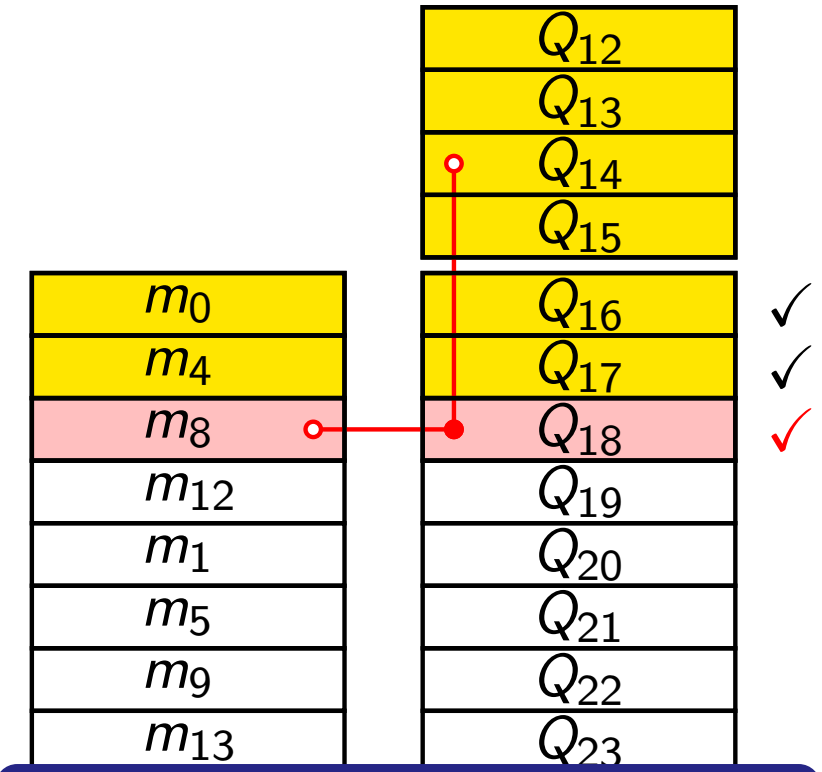


New method

- Choose the end of the first round.
- Choose m_i to satisfy both rounds.
- Fill the first round.

How to satisfy conditions in the 2nd round (New)

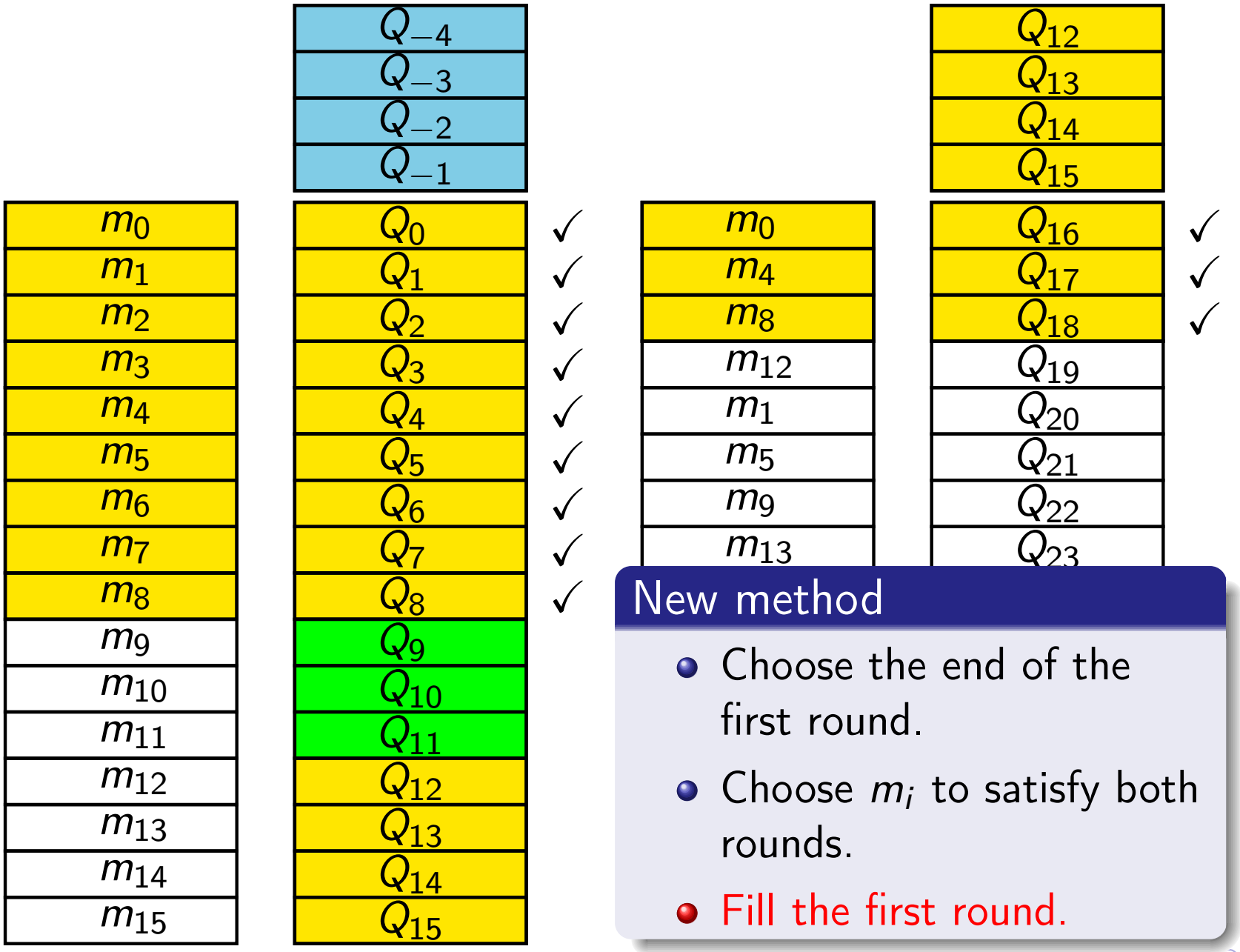




New method

- Choose the end of the first round.
- Choose m_i to satisfy both rounds.
- Fill the first round.

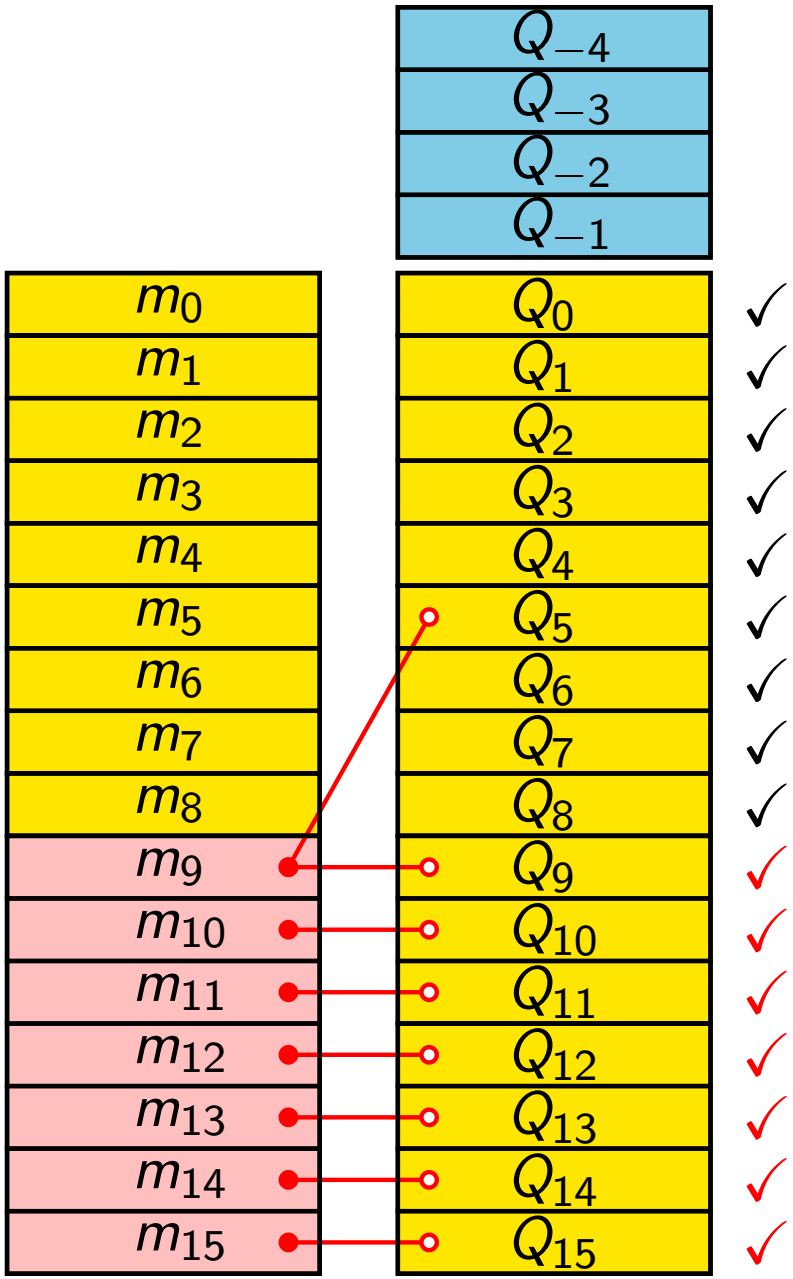
How to satisfy conditions in the 2nd round (New)

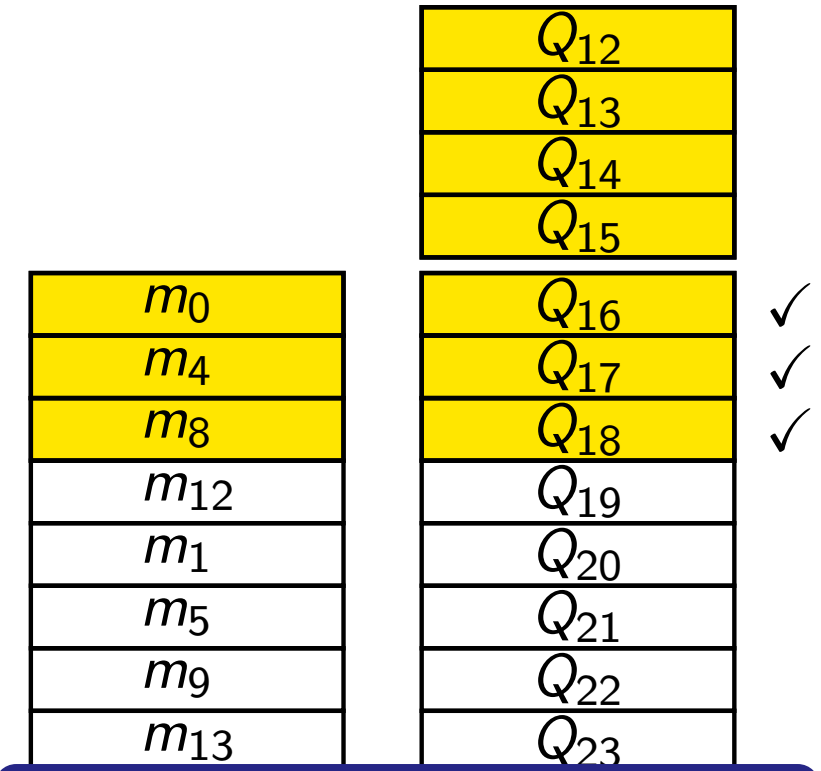


New method

- Choose the end of the first round.
- Choose m_i to satisfy both rounds.
- Fill the first round.

How to satisfy conditions in the 2nd round (New)

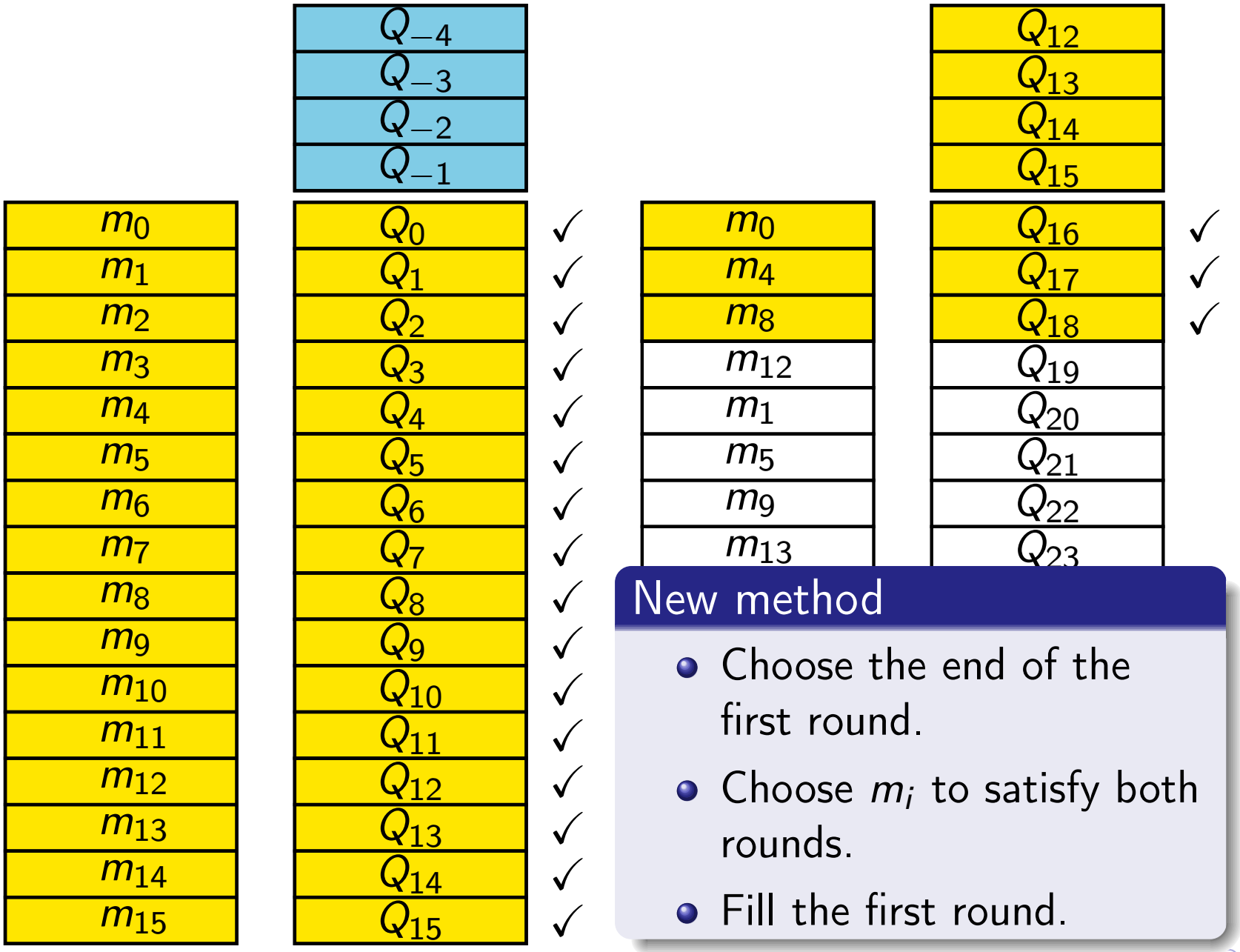




New method

- Choose the end of the first round.
- Choose m_i to satisfy both rounds.
- Fill the first round.

How to satisfy conditions in the 2nd round (New)



New method

- Choose the end of the first round.
- Choose m_i to satisfy both rounds.
- Fill the first round.

How to choose part of the message

m_0
m_1
m_2
m_3
m_4
m_5
m_6
m_7
m_8
m_9
m_{10}
m_{11}
m_{12}
m_{13}
m_{14}
m_{15}

Q_{-4}
Q_{-3}
Q_{-2}
Q_{-1}
Q_0
Q_1
Q_2
Q_3
Q_4
Q_5
Q_6
Q_7
Q_8
Q_9
Q_{10}
Q_{11}
Q_{12}
Q_{13}
Q_{14}
Q_{15}

m_0
m_4
m_8
m_{12}
m_1
m_5
m_9
m_{13}

Q_{12}
Q_{13}
Q_{14}
Q_{15}
Q_{16}
Q_{17}
Q_{18}
Q_{19}
Q_{20}
Q_{21}
Q_{22}
Q_{23}

New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

How to choose part of the message

	Q_{-4}
	Q_{-3}
	Q_{-2}
	Q_{-1}
m_0	Q_0
m_1	Q_1
m_2	Q_2
m_3	Q_3
m_4	Q_4
m_5	Q_5
m_6	Q_6
m_7	Q_7
m_8	Q_8
m_9	Q_9
m_{10}	Q_{10}
m_{11}	Q_{11}
m_{12}	Q_{12}
m_{13}	Q_{13}
m_{14}	Q_{14}
m_{15}	Q_{15}

	Q_{12}
	Q_{13}
	Q_{14}
	Q_{15}
m_0	Q_{16}
m_4	Q_{17}
m_8	Q_{18}
m_{12}	Q_{19}
m_1	Q_{20}
m_5	Q_{21}
m_9	Q_{22}
m_{13}	Q_{23}

New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

How to choose part of the message

m_0
m_1
m_2
m_3
m_4
m_5
m_6
m_7
m_8
m_9
m_{10}
m_{11}
m_{12}
m_{13}
m_{14}
m_{15}

Q_{-4}
Q_{-3}
Q_{-2}
Q_{-1}
Q_0
Q_1
Q_2
Q_3
Q_4
Q_5
Q_6
Q_7
Q_8
Q_9
Q_{10}
Q_{11}
Q_{12}
Q_{13}
Q_{14}
Q_{15}

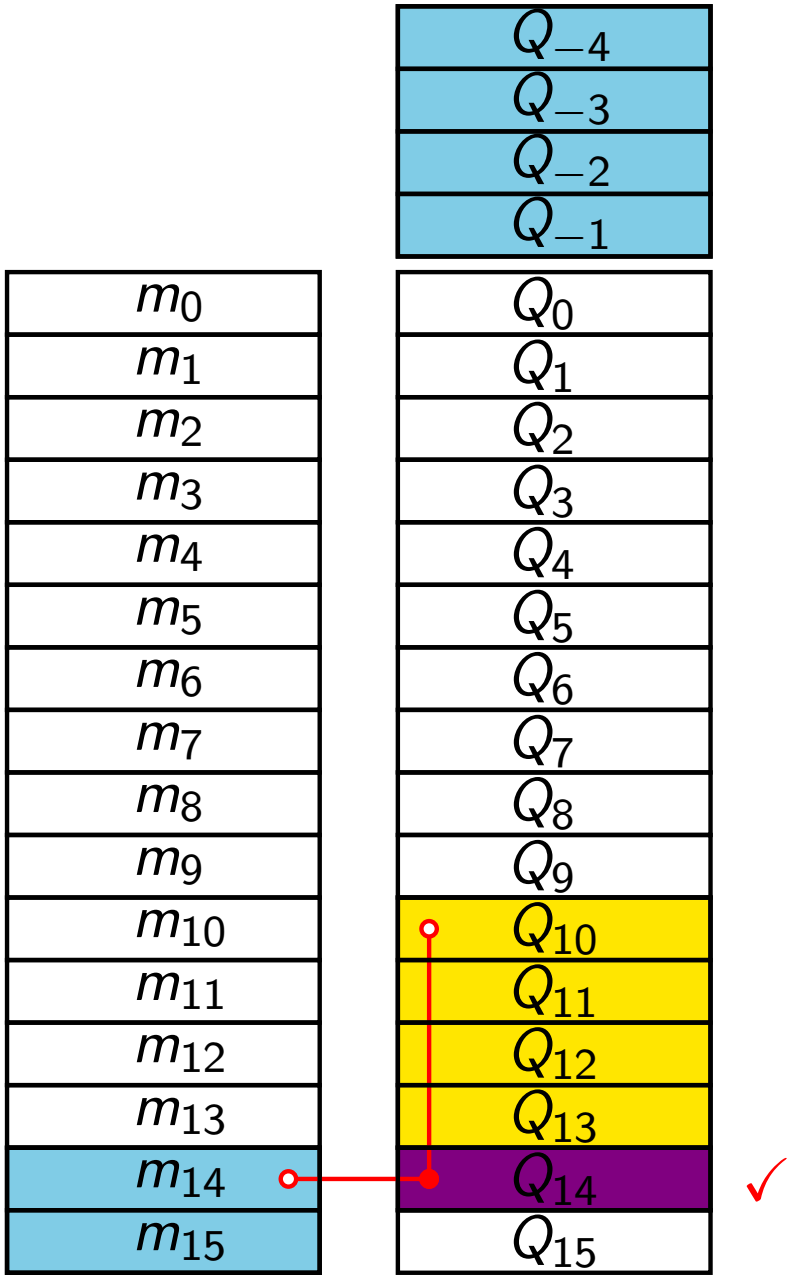
m_0
m_4
m_8
m_{12}
m_1
m_5
m_9
m_{13}

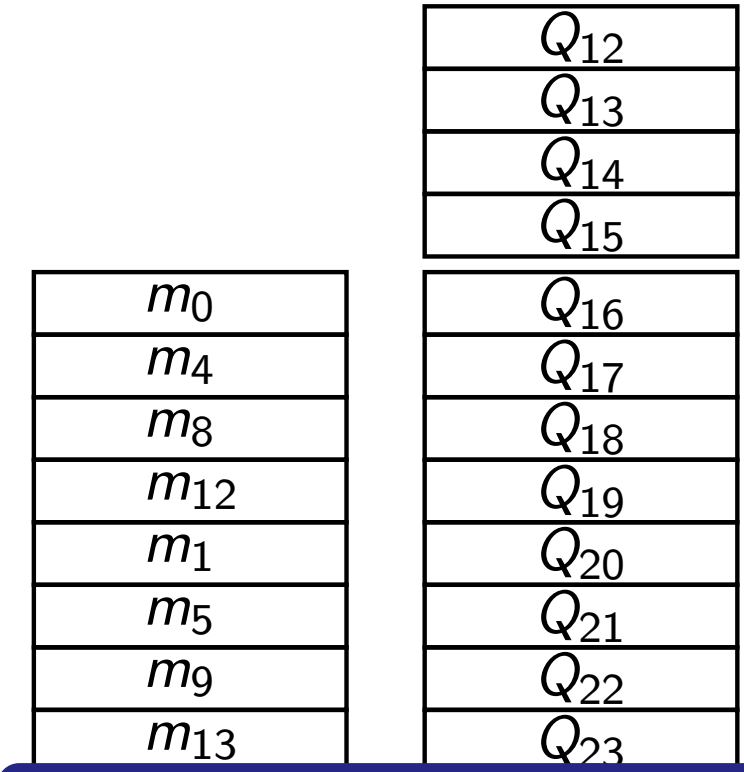
Q_{12}
Q_{13}
Q_{14}
Q_{15}
Q_{16}
Q_{17}
Q_{18}
Q_{19}
Q_{20}
Q_{21}
Q_{22}
Q_{23}

New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

How to choose part of the message

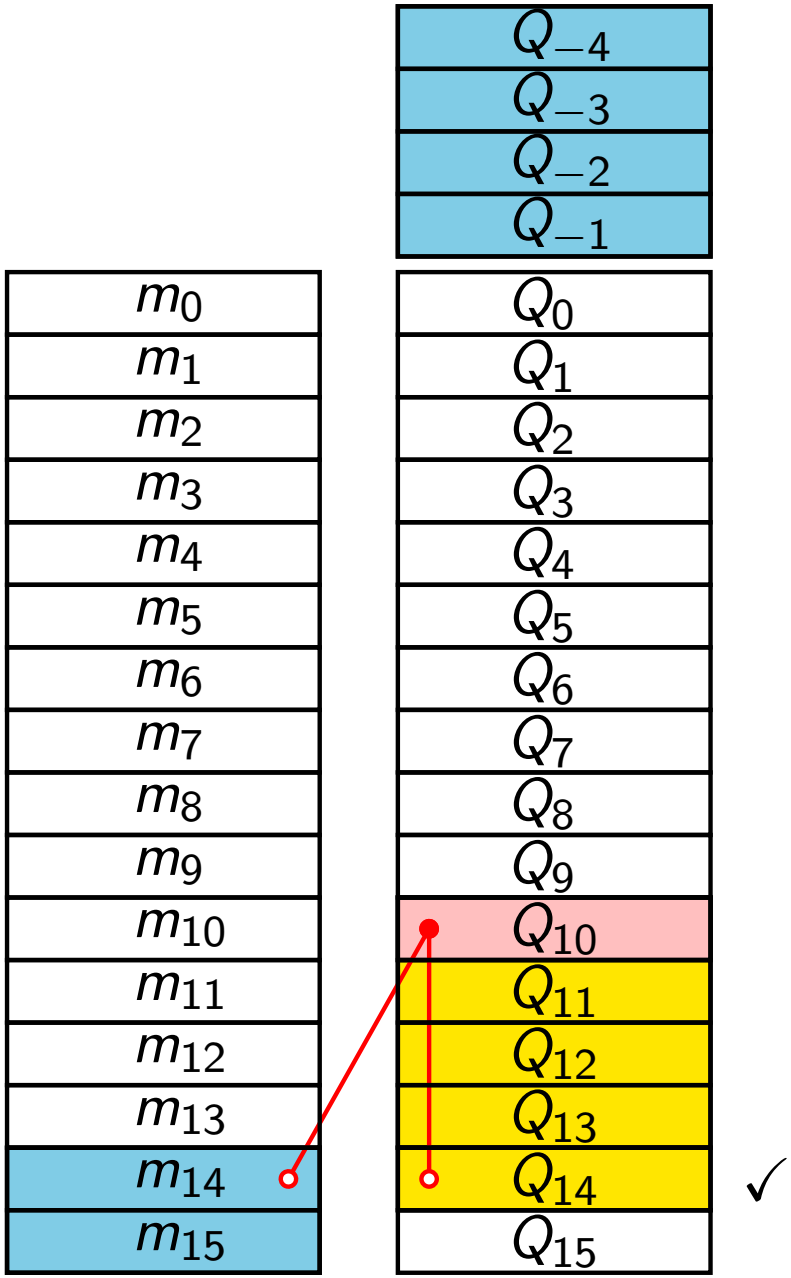




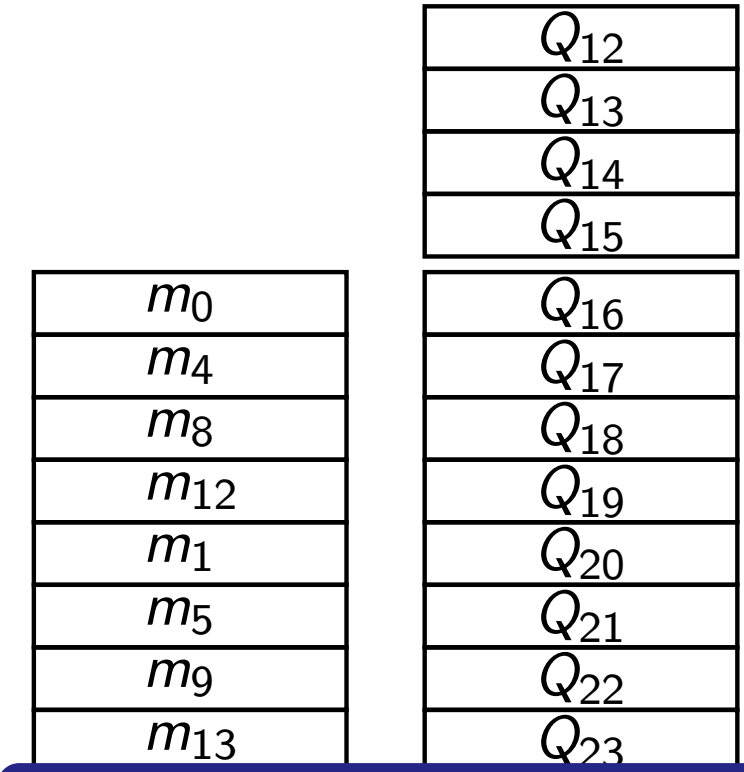
New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

How to choose part of the message



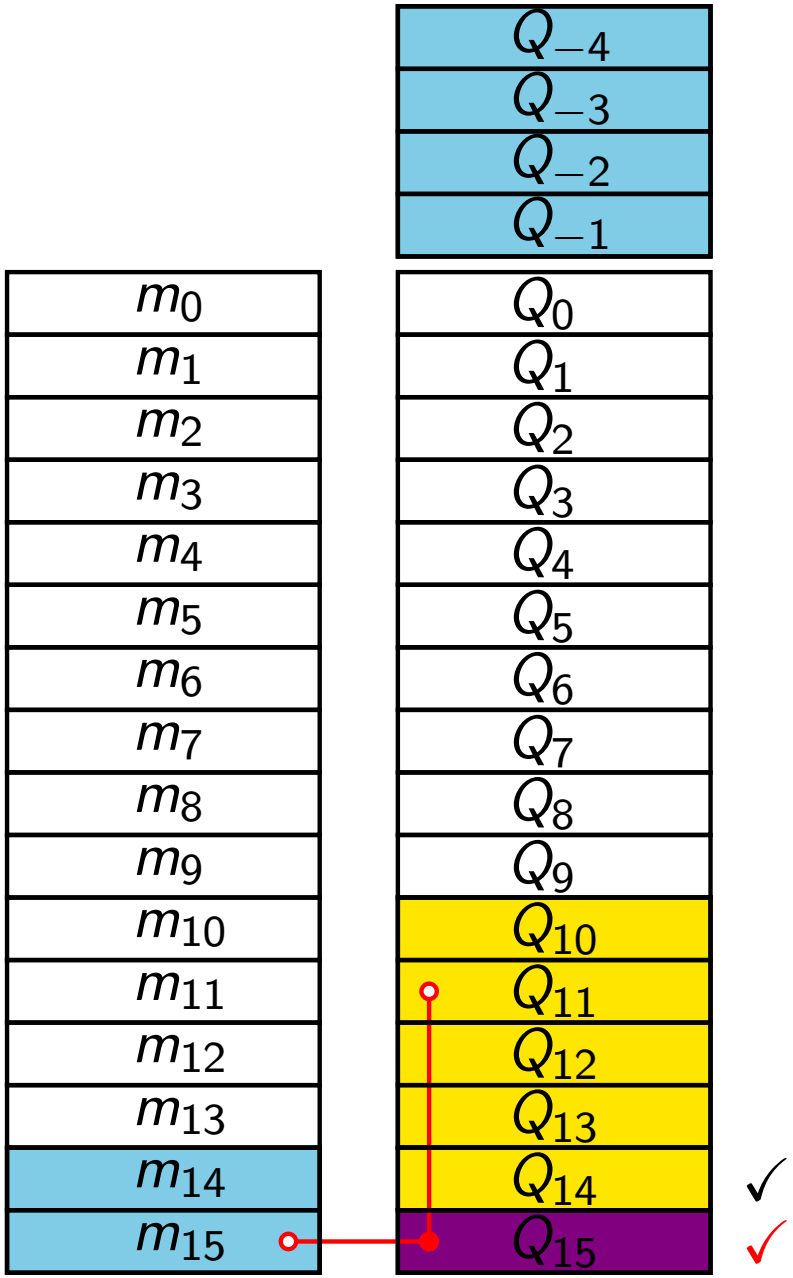
✓

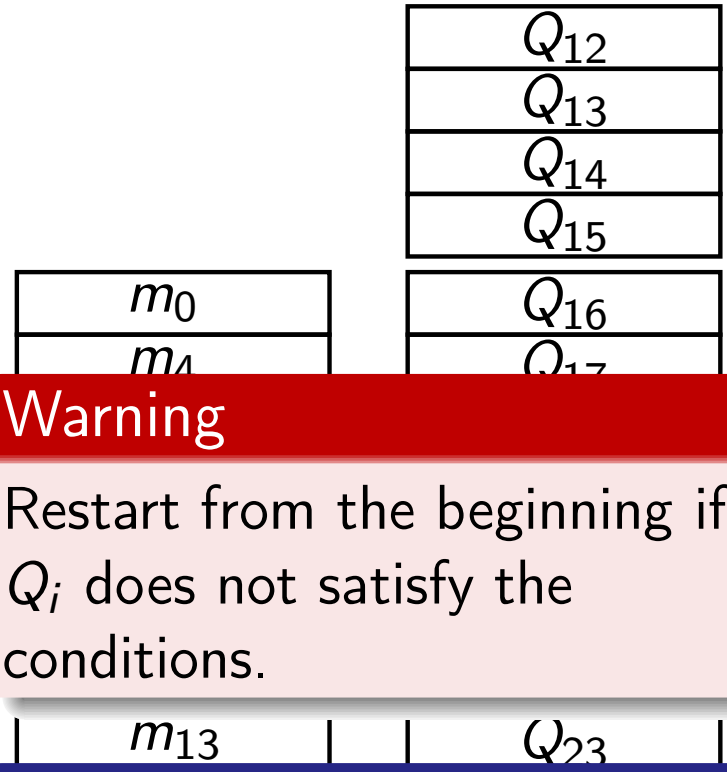


New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

How to choose part of the message





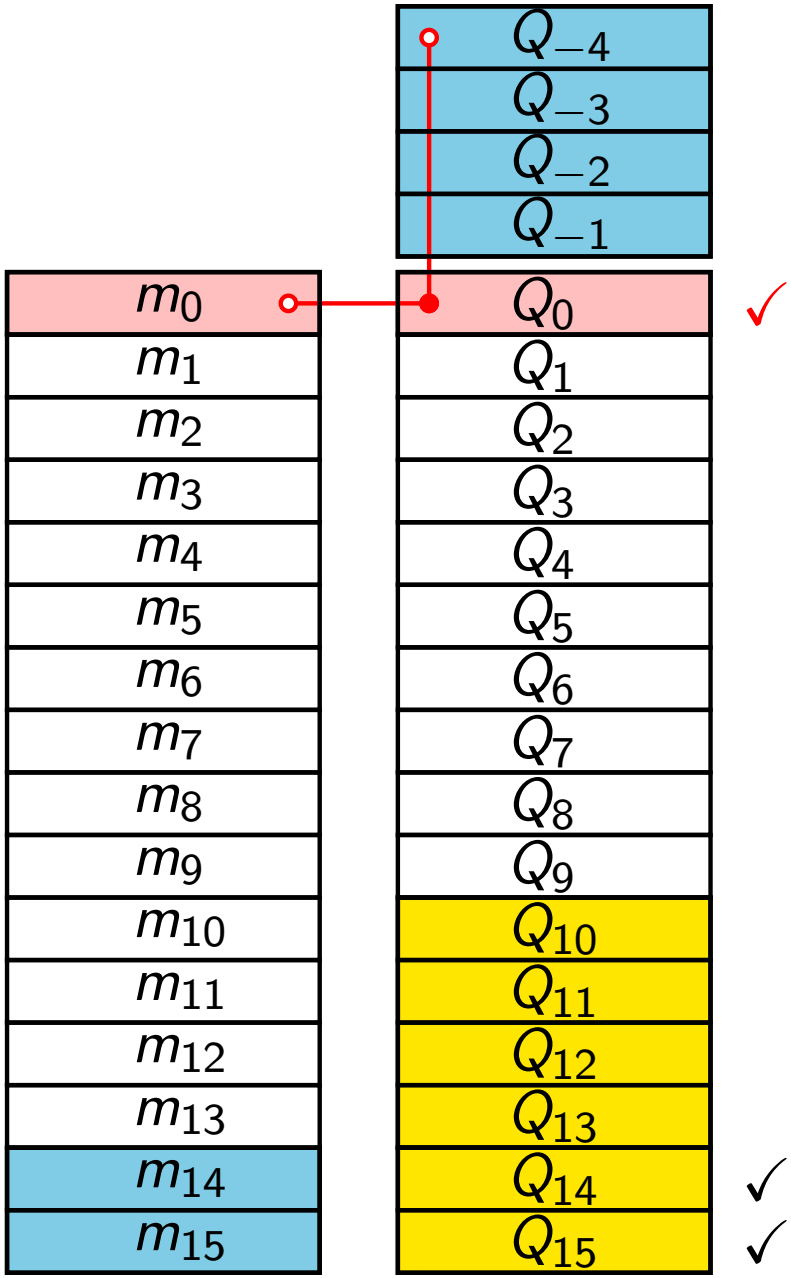
Warning

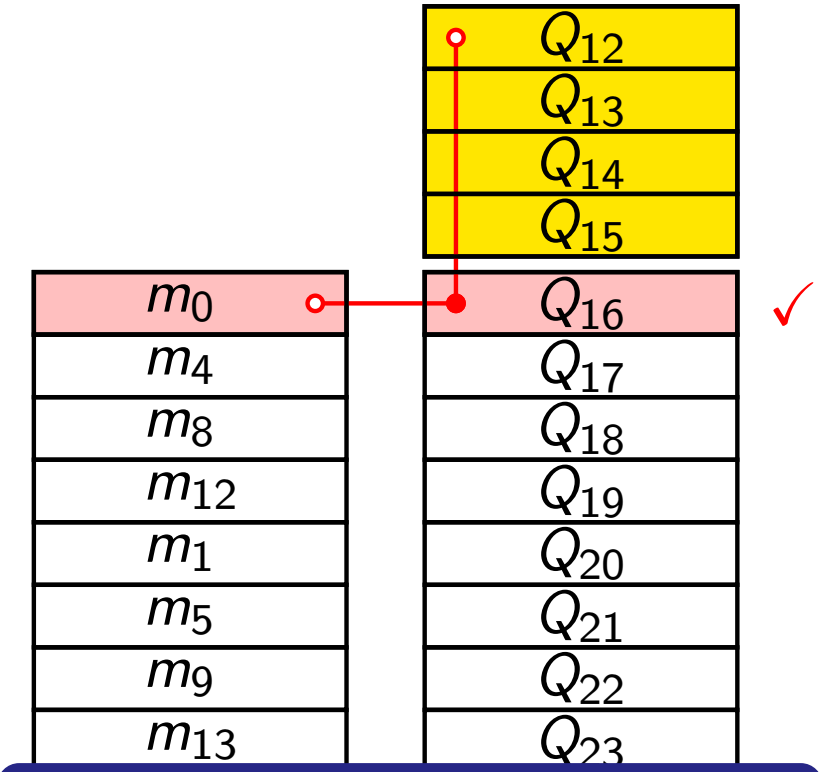
Restart from the beginning if Q_i does not satisfy the conditions.

New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

How to choose part of the message

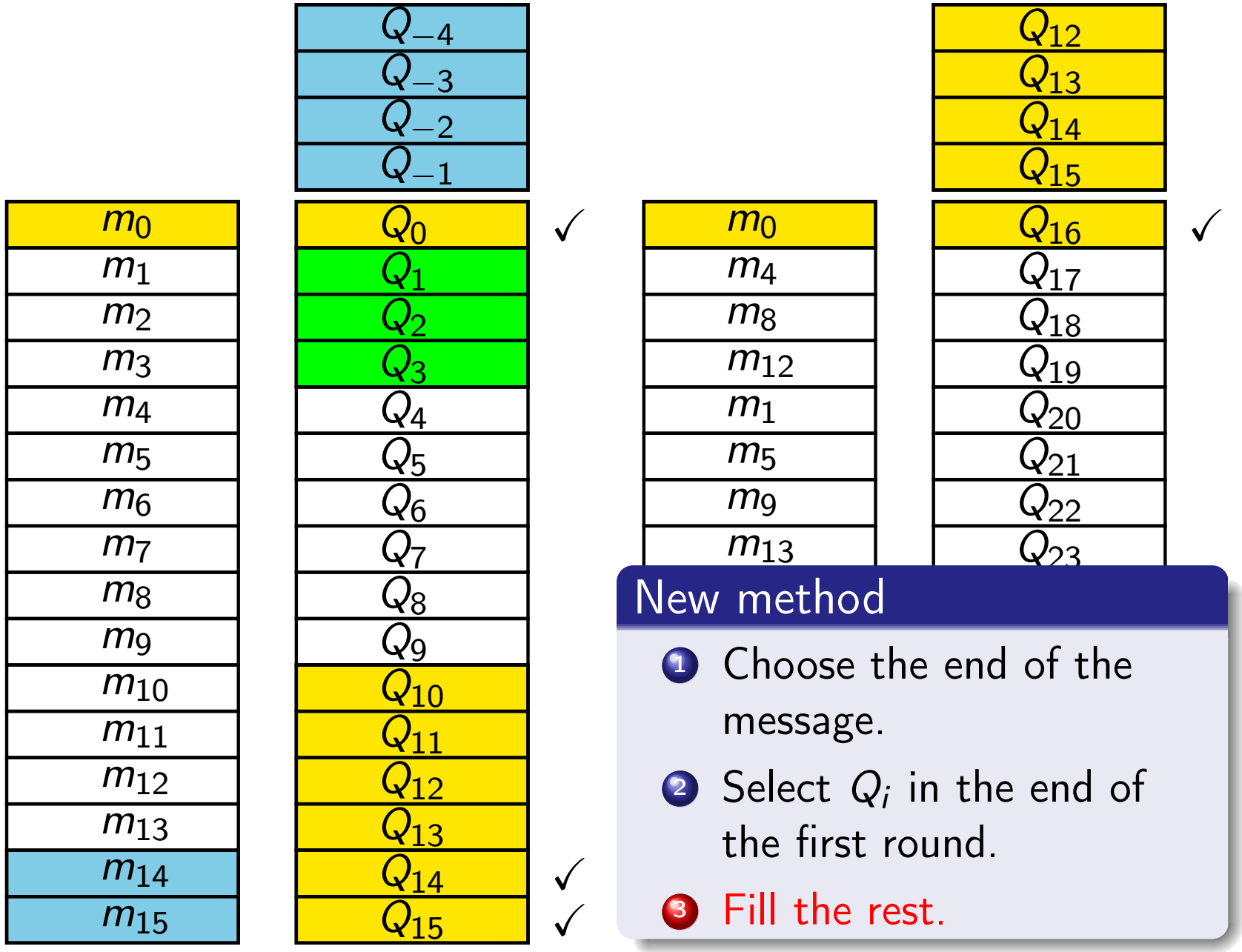




New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

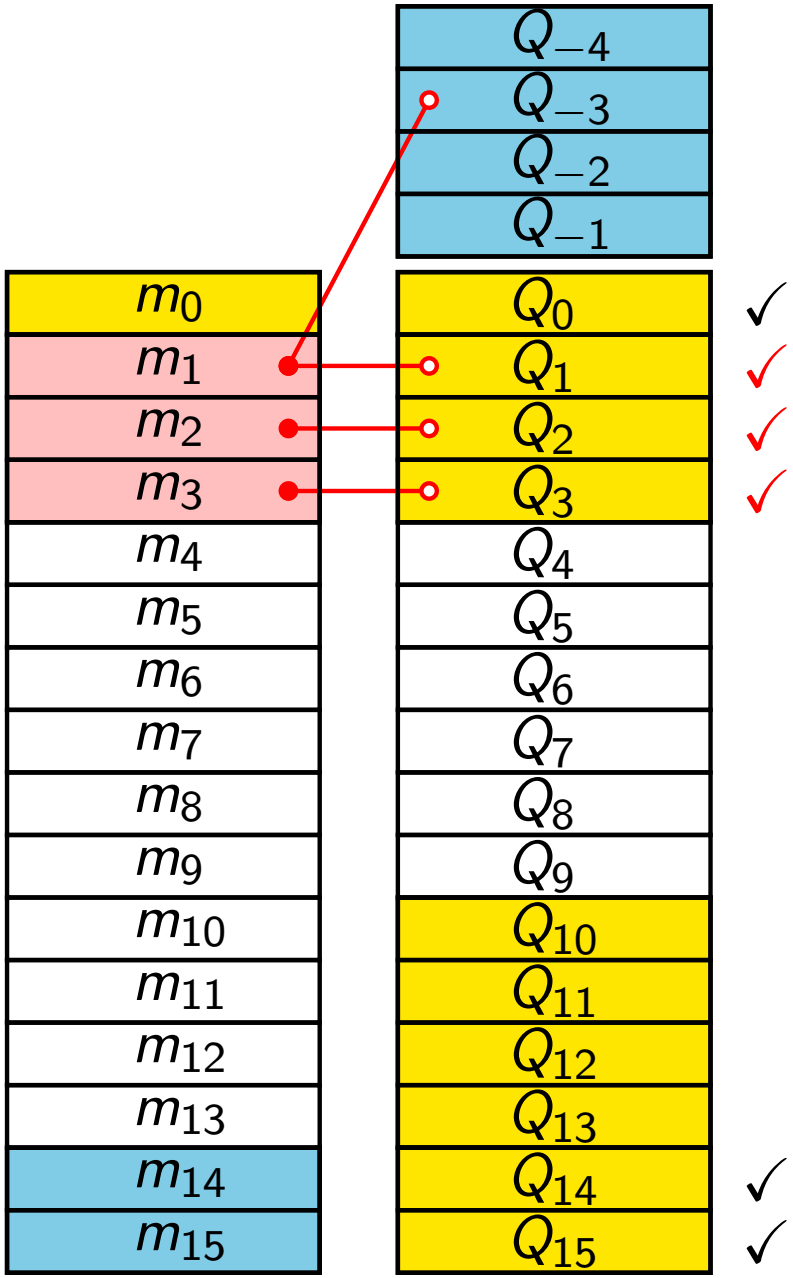
How to choose part of the message

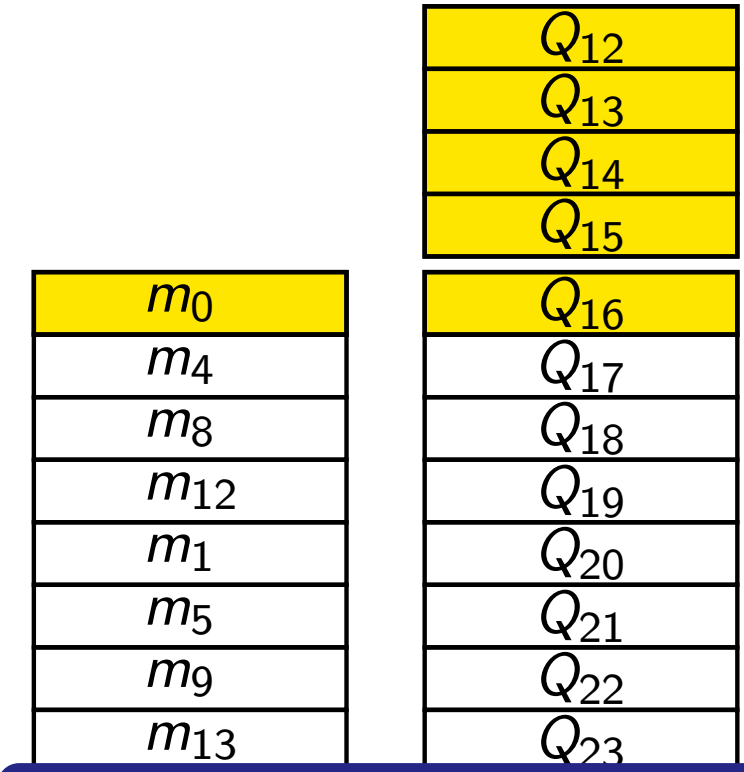


New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

How to choose part of the message

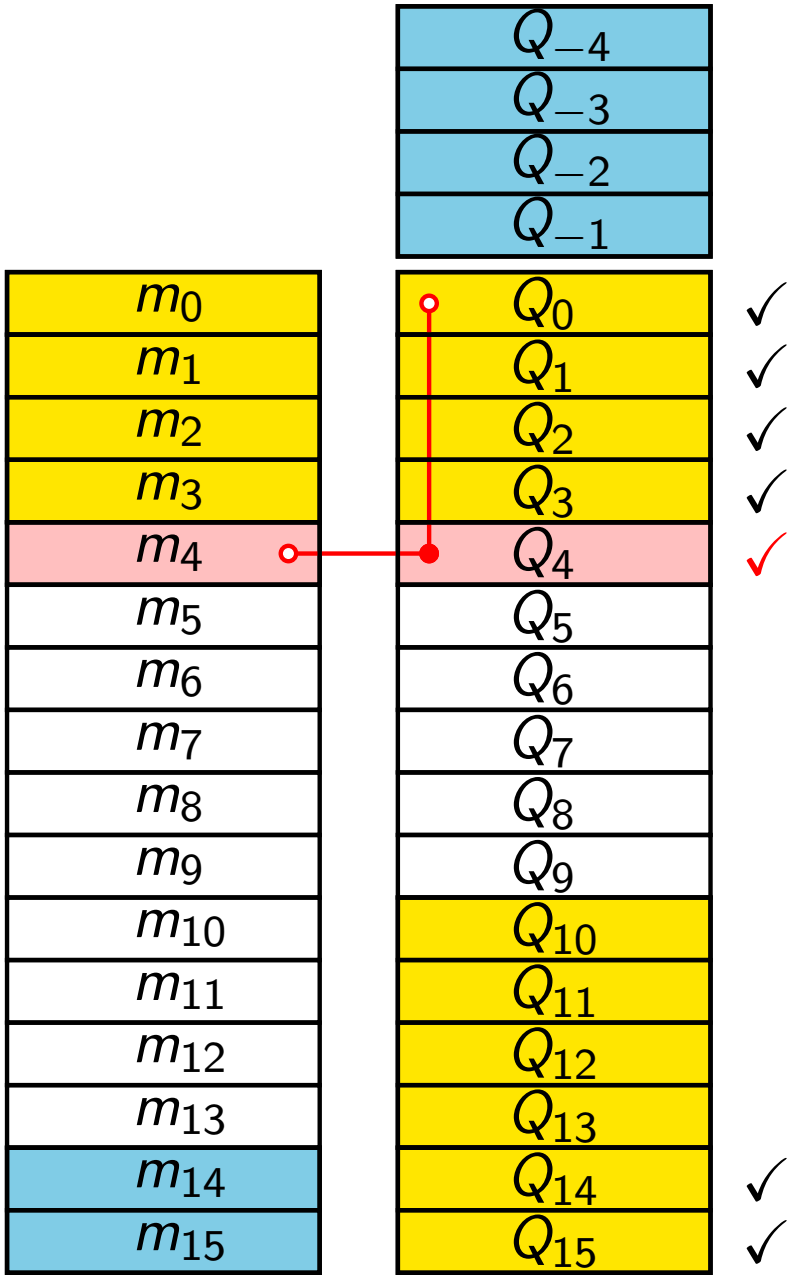


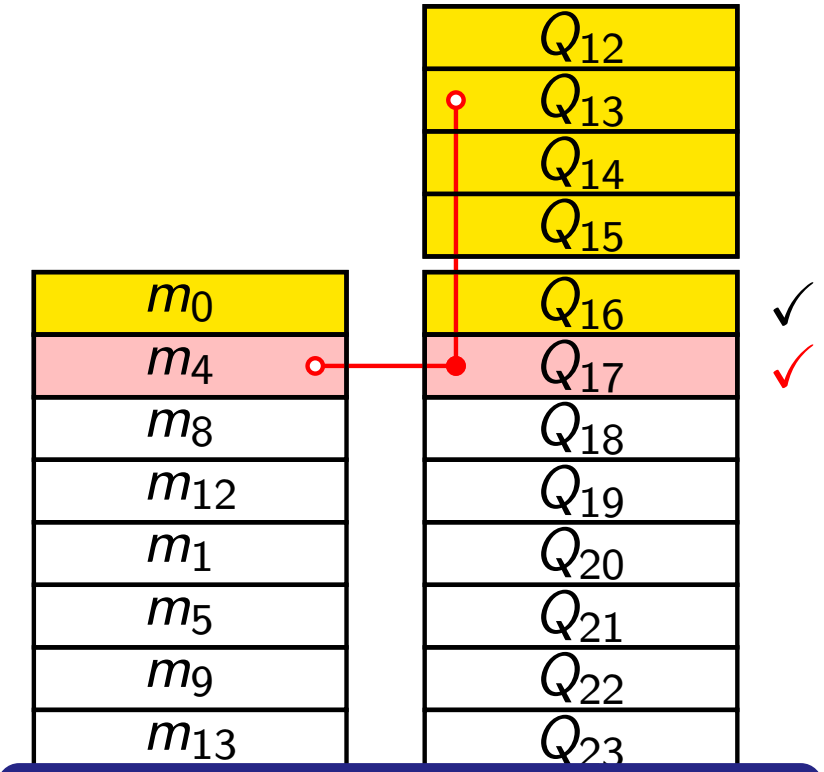


New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

How to choose part of the message

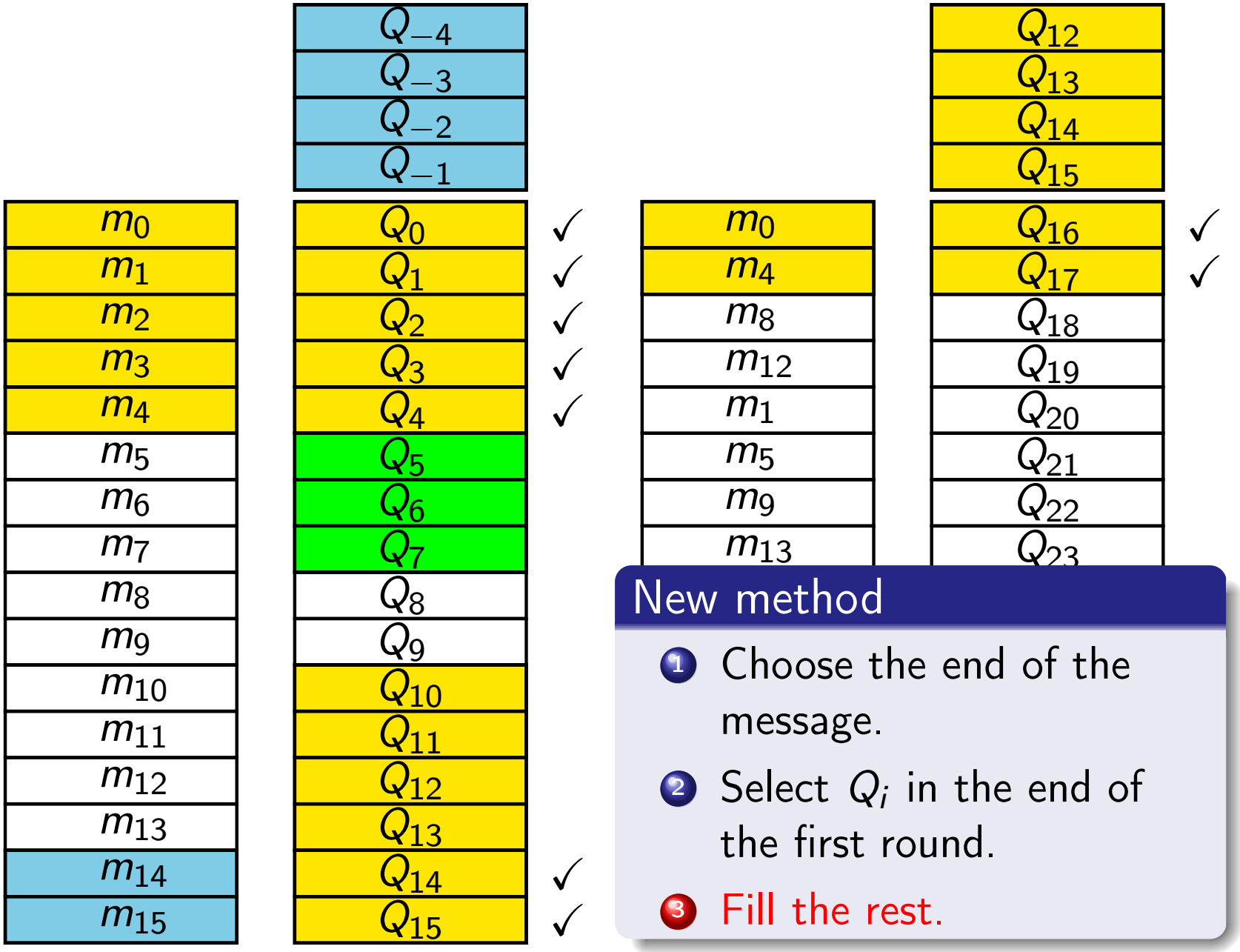




New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

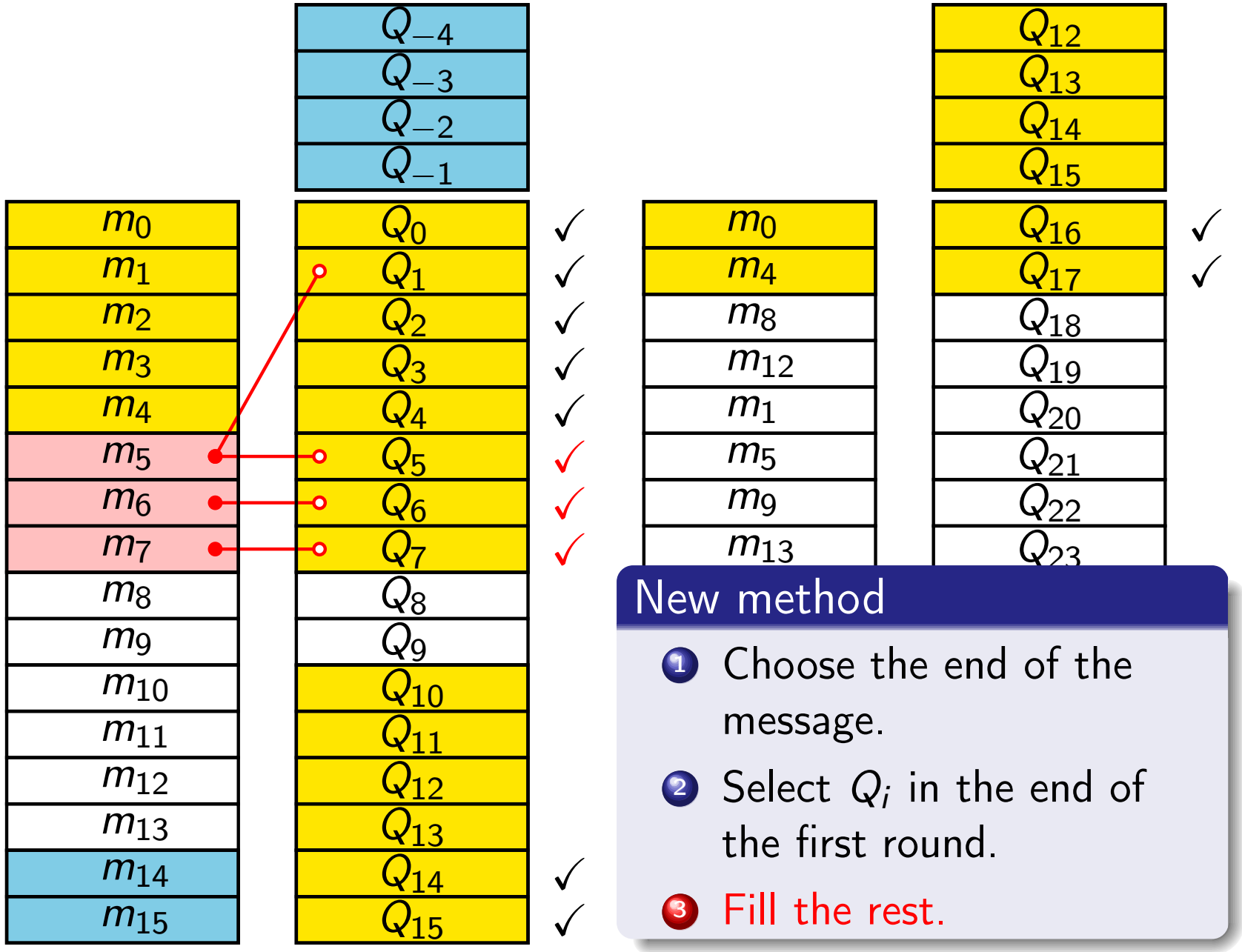
How to choose part of the message



New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

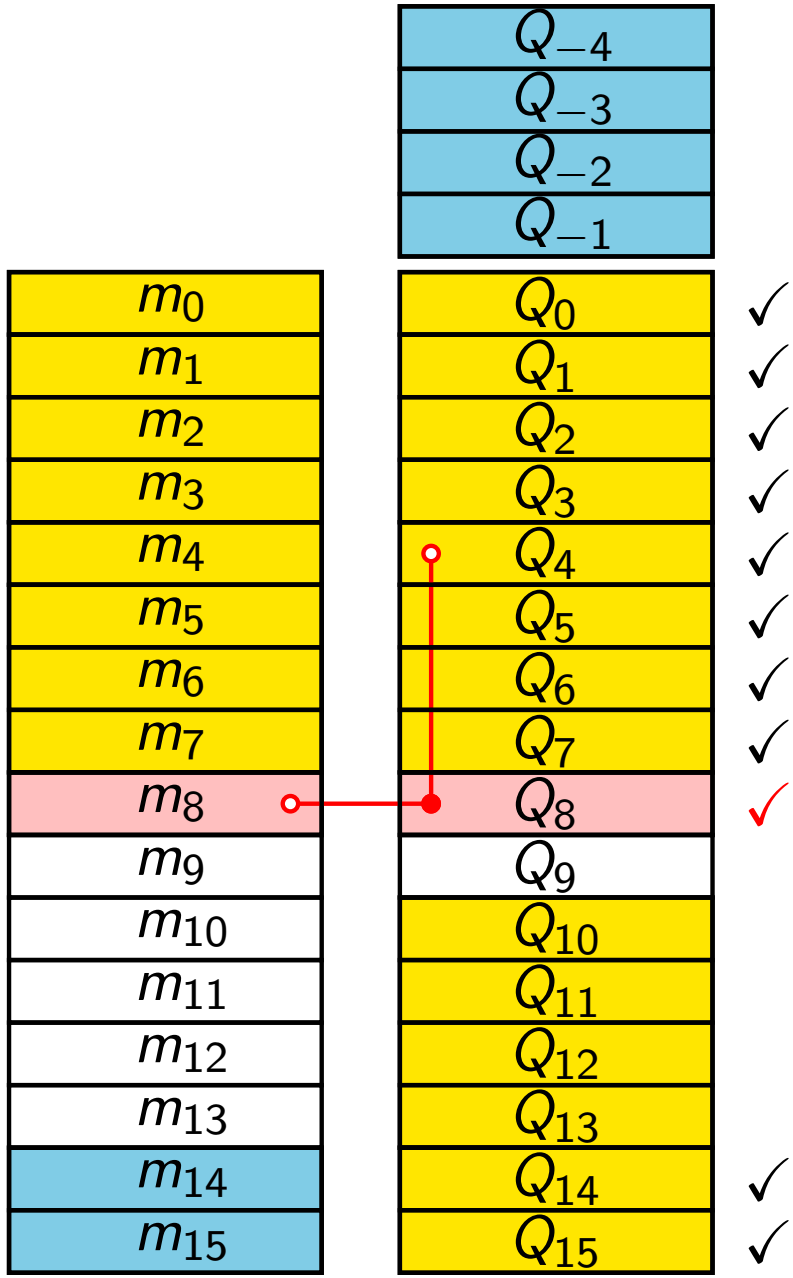
How to choose part of the message

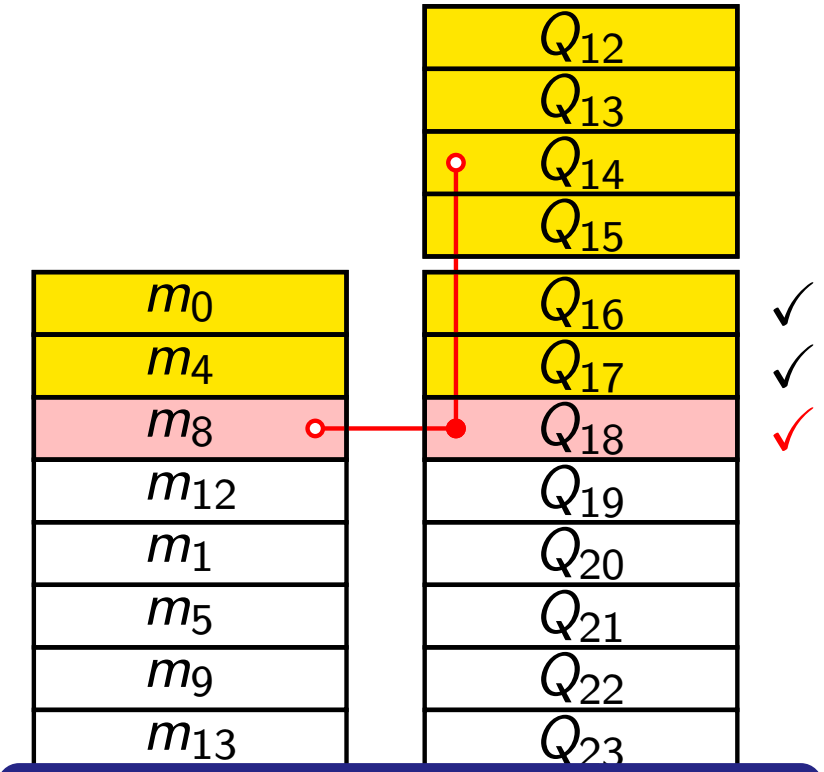


New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

How to choose part of the message

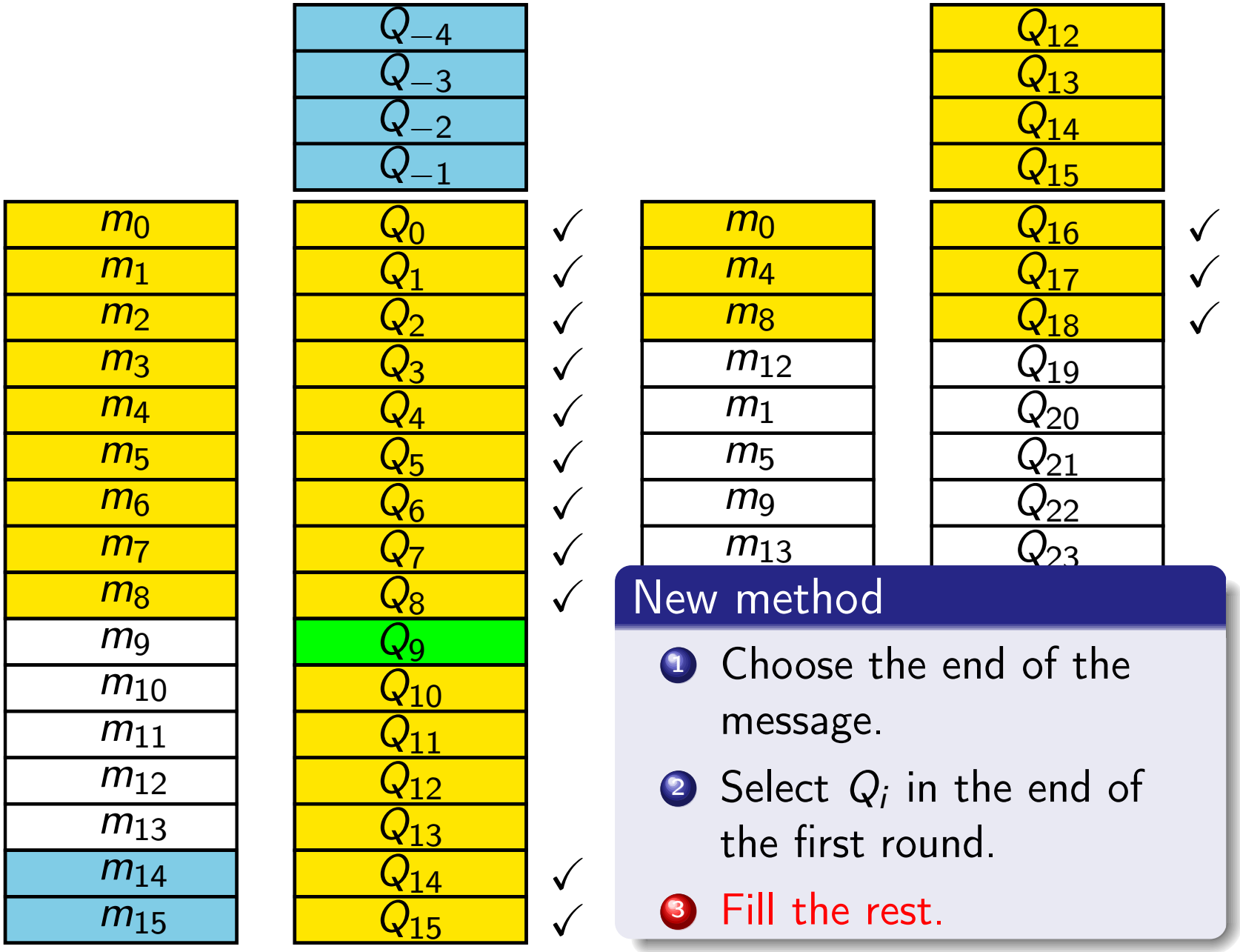




New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

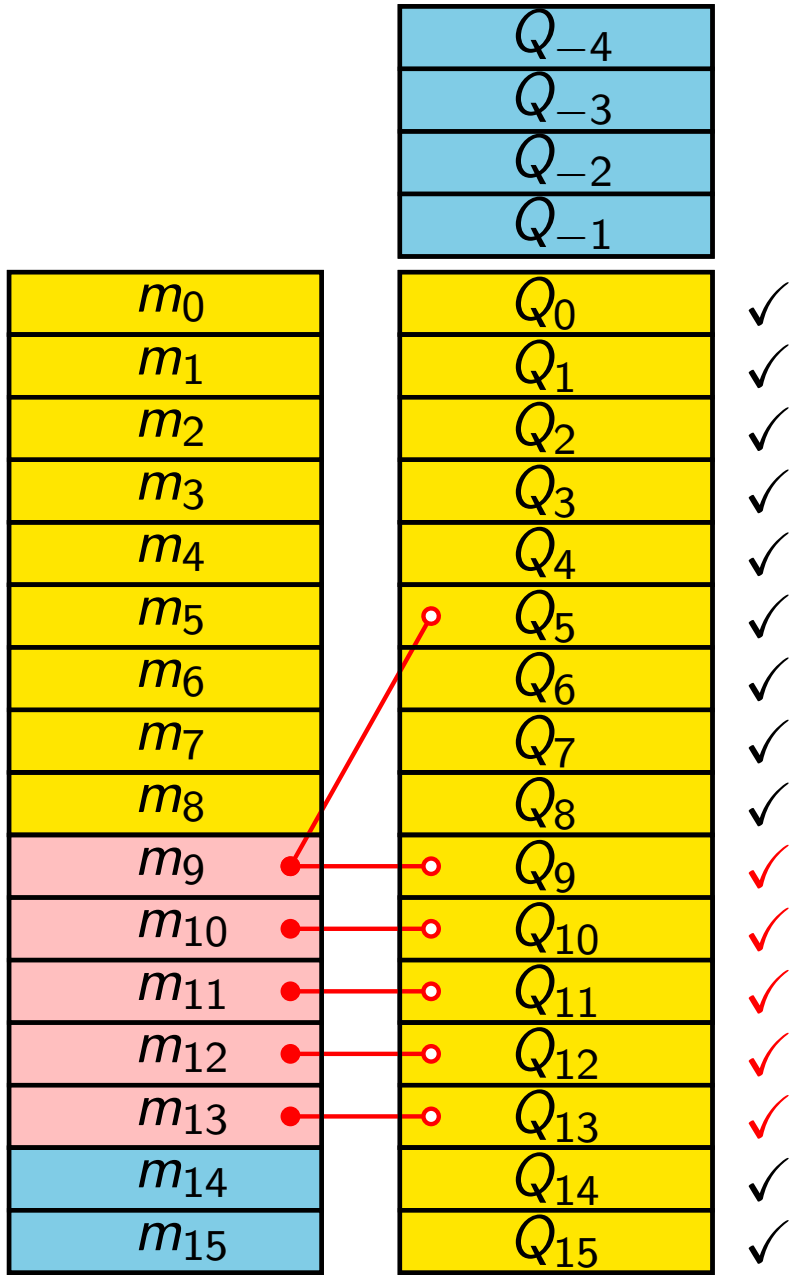
How to choose part of the message

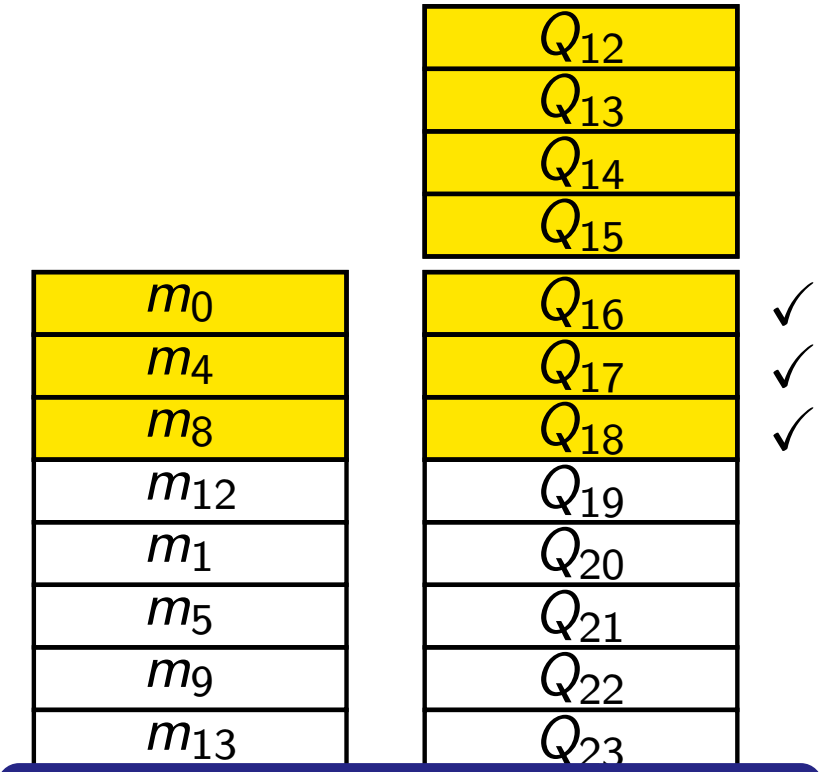


New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

How to choose part of the message

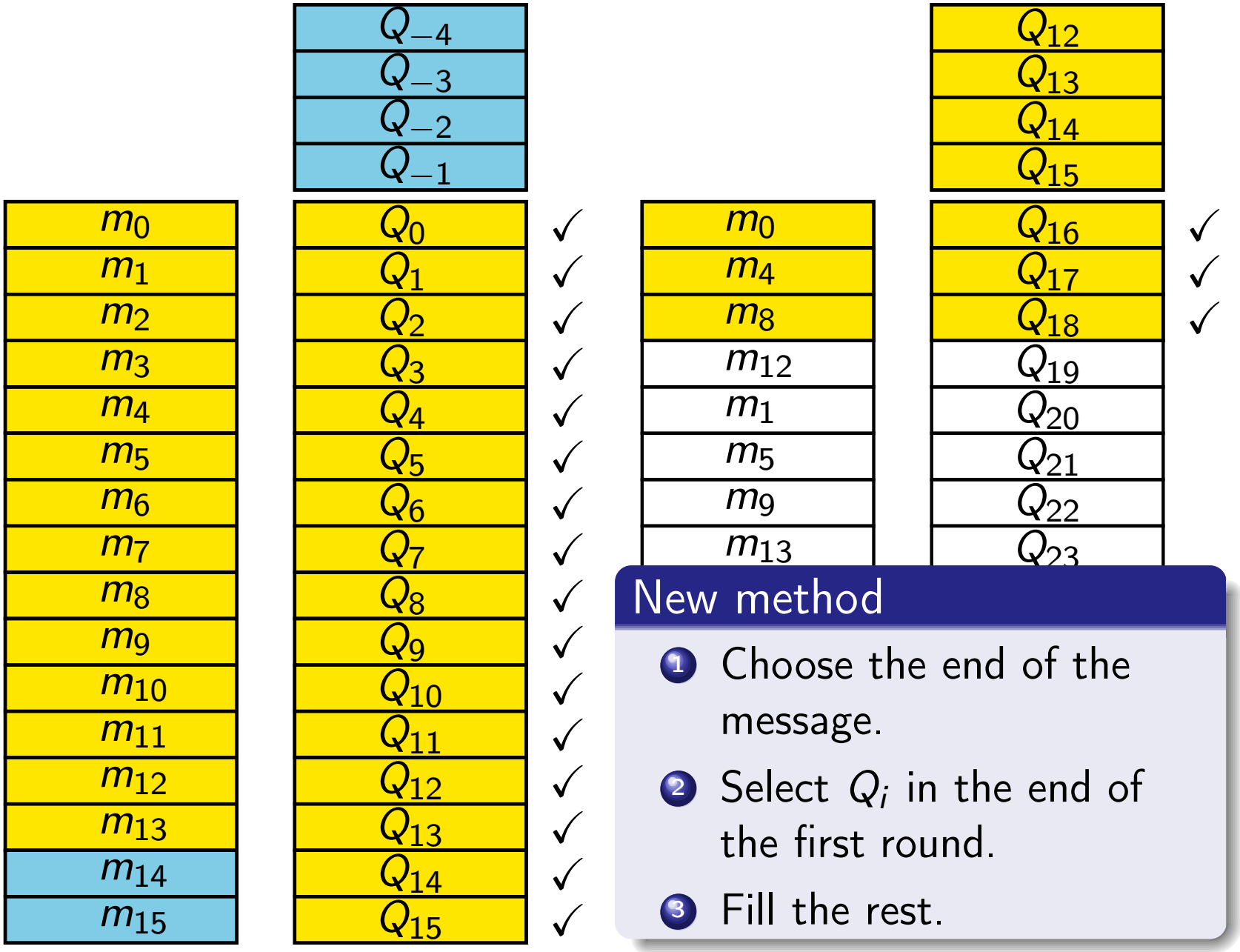




New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

How to choose part of the message



New method

1. Choose the end of the message.
2. Select Q_i in the end of the first round.
3. Fill the rest.

Message Freedom
Results

MD4 Collision Freedom

- With Wang's path (Eurocrypt 2005):

complexity about 2^5 MD4

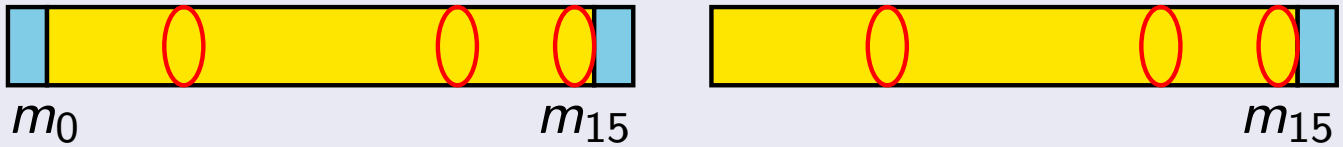
complexity about 2^{21} MD4

- With Yu's path (CANS 2005):

complexity about 2^{31} MD4

MD5 Collision Freedom

We can choose 3 words of a two block collision:



Outline

APOP challenge

RFC description

The challenge has to be a message-id:

- Begins with '<' and ends with '>'.
- Two parts separated by an '@'.
- Restricted set of allowed characters.

⇒ Attack fails (ASCII collisions not possible today).

In practice

With most mail user agents:

- Very few restrictions on the set of allowed characters

⇒ Attack works

APOP challenge

RFC description

The challenge has to be a message-id:

- Begins with '<' and ends with '>'.
- Two parts separated by an '@'.
- Restricted set of allowed characters.

$\Rightarrow$ Attack fails (ASCII collisions not possible today).

In practice

With most mail user agents:

- Very few restrictions on the set of allowed characters

$\Rightarrow$ Attack works

APOP challenge in practice

APOP challenge in practice

- Begins with '<' and ends with '>'.
- Must contain at least one '@'.
- Inside the msg-id, all characters are accepted, excepted:
  - `0x00 Null`
  - `0x3e Greater-Than Sign` ('>')
  - `0x0a Line-Feed`
  - `0x0d Carriage-Return`

Collision search

- First block computed only once: include '@'.
- For the second block:
  - avoid 4 characters.
  - ends with '>' followed by some password characters.

Using message freedom, we recover 3 password characters.

The APOP attack in practice

Mail User Agents

- Microsoft Outlook, Apple Mail: No APOP support
- KMail: Attack does not work
- Thunderbird, Evolution, Mutt, Fetchmail: Attack works

Attack complexity

- A few seconds of computations per collision.
- Need about 100 collisions (200 authentications): if the client checks mails every minute, 3 hours.
- Remaining characters can be brute-forced: 5 characters $\longrightarrow$ 2 hours.

Recommendations

- Mail User Agent: check challenge conformity to RFC.
- Users: avoid APOP if possible (eg. use CRAM-MD5).

The APOP attack in practice

Mail User Agents

- Microsoft Outlook, Apple Mail: No APOP support
- KMail: Attack does not work
- Thunderbird, Evolution, Mutt, Fetchmail: Attack works

Attack complexity

- A few seconds of computations per collision.
- Need about 100 collisions (200 authentications): if the client checks mails every minute, 3 hours.
- Remaining characters can be brute-forced: 5 characters $\longrightarrow$ 2 hours.

Recommendations

- Mail User Agent: check challenge conformity to RFC.
- Users: avoid APOP if possible (eg. use CRAM-MD5).

The APOP attack in practice

Mail User Agents

- Microsoft Outlook, Apple Mail: No APOP support
- KMail: Attack does not work
- Thunderbird, Evolution, Mutt, Fetchmail: Attack works

Attack complexity

- A few seconds of computations per collision.
- Need about 100 collisions (200 authentications): if the client checks mails every minute, 3 hours.
- Remaining characters can be brute-forced: 5 characters $\longrightarrow$ 2 hours.

Recommendations

- Mail User Agent: check challenge conformity to RFC.
- Users: avoid APOP if possible (eg. use CRAM-MD5).

Conclusion

Summary

- Wang's attack allows some message freedom.
- Collision attacks can be used to attack real protocols.

Outlook

- Study collision impact against other authentication protocols.
- New differential paths could improve the attack:
  - Better message difference (ASCII collisions)
  - More freedom

Table: APOP MD5 collision. These two msg-id's collides if padded with "bar"

Message M	
<xxxÑÕç\ØºÄ4<0	3c 78 78 78 d1 d5 e7 5c 88 d8 ba e4
mXÀn]Ø\4UµiQ1	6d 58 c0 9f 6e 5d 17 d8 5c 34 55 08
Åì"xô!¿ïFsc'áÙ	c5 ec 22 06 02 78 f4 21 bf ef 46 73
Ýìvsôgcû49¾VJxxx@	dd ec 8a f4 47 1b fb 34 39 be 56 89
ÚcCòßPºQÙÖRéý	da 63 43 f2 df 50 ba 05 d9 d6 09 83
ÄC4$½MH-:y6ÇÊ©Üs	c4 94 34 24 bd 4d 48 2d 3a 79 36 c7
-0:Þñw|enuë5{cþ¾	2d 30 3a de f1 95 7c 65 6e 8c eb 35
ïL´ïªØ#»ì])ü>	ef 4c b4 ef aa d8 23 bb ec 5d 2
Message M'	
<xxxÑÕç\ØºÄ4<0	3c 78 78 78 d1 d5 e7 5c 88 d8 ba e4
mXÀun]Ø\4UµiQ1	6d 58 c0 1f 6e 5d 17 d8 5c 34 55 08
Åì"xô!¿ïFsc§áÙ	c5 ec 22 06 02 78 f4 21 bf ef 46 73
Ýìvsôgcû49¾VTxxx@	dd ec 8a f4 47 1b fb 34 39 be 56 09
ÚcCòßPºQÙÖRéý	da 63 43 f2 df 50 ba 05 d9 d6 09 83
ÄC4O½MH-:y6CÊ©Üs	c4 94 34 a4 bd 4d 48 2d 3a 79 36 c7