



## (12)发明专利申请

(10)申请公布号 CN 107251520 A

(43)申请公布日 2017.10.13

(21)申请号 201680010918.1

(74)专利代理机构 北京龙双利达知识产权代理

(22)申请日 2016.04.12

有限公司 11329

(30)优先权数据

10201503071U 2015.04.20 SG

代理人 魏雪娇 毛威

(85)PCT国际申请进入国家阶段日

(51)Int.Cl.

H04L 29/06(2006.01)

2017.08.21

H04L 9/32(2006.01)

(86)PCT国际申请的申请数据

PCT/SG2016/050177 2016.04.12

(87)PCT国际申请的公布数据

W02016/171618 EN 2016.10.27

(71)申请人 华为国际有限公司

地址 新加坡新加坡市15A樟宜商务园中央  
1#03-03

(72)发明人 时杰 王贵林 吴双

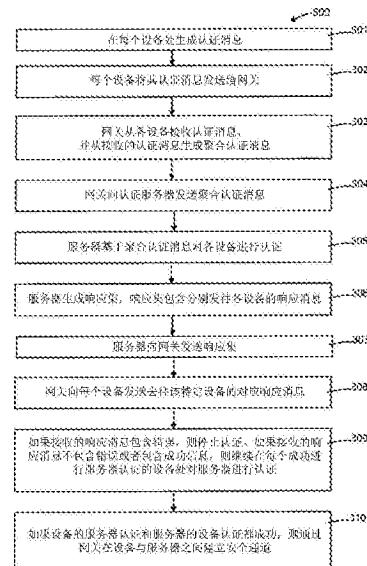
权利要求书8页 说明书9页 附图7页

(54)发明名称

用于M2M通信中的聚合认证协议的方法

(57)摘要

本发明实施例提供了一种用于M2M认证服务器和M2M设备的端到端认证协议，其中所述M2M认证服务器和M2M设备直接相互认证。无需假设M2M网关可信。避免了使用组认证和组标识符。此外，相互认证协议采用基于对称密钥的技术。



1. 一种用于通过服务器认证多个机对机(Machine-to-Machine, M2M)设备的方法, 其特征在于, 所述方法包括:

使用基于对称密钥的技术, 基于从网关接收的聚合认证消息对所述多个M2M设备进行服务器认证, 所述聚合认证消息基于从所述多个M2M设备分别接收的多个认证消息在所述网关处生成; 以及

生成多个认证响应, 所述多个认证响应通过所述网关分别发往所述多个M2M设备, 其中所述多个认证响应中的至少一些指定用于所述服务器的设备认证, 所述设备认证将使用基于对称密钥的技术在所述多个M2M设备中的成功进行服务器认证的M2M设备处执行。

2. 根据权利要求1所述的方法, 其特征在于, 所述聚合认证消息包括:

从包含在所述多个认证消息中的第一多个设备生成消息认证码(Message Authentication Code, MAC)计算出的聚合设备生成MAC,

对所述多个M2M设备进行服务器认证包括:

验证包含在所述聚合认证消息中的多个设备生成新参数的有效性, 以及

验证所述聚合设备生成MAC相对于第一聚合服务器生成MAC的有效性, 所述第一聚合服务器生成MAC是从所述多个M2M设备的第一多个服务器生成MAC计算出的, 以及

生成多个认证响应包括:

计算所述多个M2M设备的第二多个服务器生成MAC, 其中所述第二多个服务器生成MAC包含在所述多个认证响应中, 用于验证相对于所述多个M2M设备中成功进行服务器认证的M2M设备的多个第二设备生成MAC的有效性。

3. 根据权利要求2所述的方法, 其特征在于, 如果所述多个设备生成新参数中的任何一个无效, 或者如果所述聚合设备生成MAC相对于所述第一聚合服务器生成MAC无效, 则所述方法还包括:

将所述多个M2M设备添加到失败集中。

4. 根据权利要求2所述的方法, 其特征在于, 如果所述多个设备生成新参数中的任何一个无效, 则所述方法还包括:

将所述多个M2M设备中的任何具有无效的设备生成新参数的M2M设备添加到失败集中;

从所述网关获得所述多个认证消息的子集, 所述子集对应于所述多个M2M设备中的未包含在所述失败集中的剩余部分; 以及

在非聚合的基础上验证包含在所述多个认证消息的所述子集中的所述第一多个设备生成MAC的有效性;

如果包含在所述多个认证消息的所述子集中的所述第一多个设备生成MAC中的任何一个有效, 则将所述多个M2M设备中的具有有效的设备生成MAC的对应第一子集添加到成功集中; 以及

如果包含在所述多个认证消息的所述子集中的所述第一多个设备生成MAC中的任何一个无效, 则将所述多个M2M设备中的具有无效的设备生成MAC的对应第二子集添加到所述失败集中。

5. 根据权利要求2所述的方法, 其特征在于, 如果所述多个设备生成新参数中的任何一个无效, 则所述方法还包括:

将所述多个M2M设备中的任何具有无效的设备生成新参数的M2M设备添加到失败集中;

从所述网关接收布隆过滤器，所述布隆过滤器在所述网关处基于所述多个认证消息中的一个或多个而创建，所述多个认证消息中的一个或多个与所述多个M2M设备中的未包含在所述失败集中的剩余部分对应；

计算所述多个M2M设备的第三多个服务器生成MAC；

在所述布隆过滤器中查询所述第三多个服务器生成MAC；

如果所述第三多个设备生成MAC的第一子集存在于所述布隆过滤器中，则将与所述第三多个设备生成MAC的所述第一子集对应的所述多个M2M设备的第一子集添加到临时成功集(temporary success set, TSS)中；

如果所述第三多个设备生成MAC的第二子集不在所述布隆过滤器中，则将与所述第三多个设备生成MAC的所述第二子集对应的所述多个M2M设备的第二子集添加到临时失败集(temporary failure set, TFS)中；

验证临时聚合设备生成MAC相对于包含在所述临时成功集(temporary success set, TSS)中的所述多个M2M设备的所述第一子集的第四多个服务器生成MAC的有效性，其中所述临时聚合设备生成MAC是从所述网关接收的；

如果所述临时聚合设备生成MAC有效，则将所述多个M2M设备的所述第一子集添加到所述成功集中；以及

如果所述临时聚合设备生成MAC无效，则将所述多个M2M设备的所述第一子集添加到所述失败集中，并将所述多个M2M设备的所述第二子集添加到所述失败集中。

6. 根据权利要求2所述的方法，其特征在于，来自所述第一多个设备生成MAC的所述聚合设备生成MAC通过以下方式计算：

对所述第一多个设备生成MAC执行压缩函数。

7. 根据权利要求2所述的方法，其特征在于，所述多个新参数包括时间戳。

8. 根据权利要求1至7中的任一项所述的方法，其特征在于，还包括：

在从所述网关接收所述聚合认证消息之前，对所述服务器与所述网关进行相互认证，并在它们之间建立第一安全通道。

9. 根据权利要求1至7中的任一项所述的方法，其特征在于，还包括：

在从所述网关接收所述聚合认证消息之前，对所述服务器与所述网关进行相互认证，并在所述服务器与所述网关之间建立第一安全通道，其中所述网关已与所述多个M2M设备进行相互认证并且它们之间已建立多个第二安全通道。

10. 一种用于对多个机对机(Machine-to-Machine, M2M)设备进行认证的服务器，其特征在于，所述系统包括：处理器；以及存储设备，其存储供所述处理器执行的计算机可读指令，

其中所述处理器用于：使用基于对称密钥的技术，基于从网关接收的聚合认证消息对所述多个M2M设备进行认证，所述聚合认证消息基于从所述多个M2M设备分别接收的多个认证消息在所述网关处生成；生成多个认证响应，所述多个认证响应通过所述网关分别发往所述多个M2M设备，其中所述多个认证响应中的至少一些指定用于所述服务器的设备认证，所述设备认证将使用基于对称密钥的技术在所述多个M2M设备中的成功进行服务器认证的M2M设备处执行。

11. 根据权利要求10所述的服务器，其特征在于，所述聚合设备生成消息认证码

(Message Authentication Code, MAC) 是从包含在所述多个认证消息中的第一多个设备生成MAC计算出的,以及

所述处理器还用于:验证包含在所述聚合认证消息中的多个设备生成新参数的有效性;验证所述聚合设备生成MAC相对于第一聚合服务器生成MAC的有效性,所述第一聚合服务器生成MAC是从所述多个M2M设备的第一多个服务器生成MAC计算出的;计算所述多个M2M设备的第二多个服务器生成MAC,其中所述第二多个服务器生成MAC包含在所述多个认证响应中,用于验证相对于所述多个M2M设备中成功进行服务器认证的M2M设备的多个第二设备生成MAC的有效性。

12. 根据权利要求11所述的服务器,其特征在于,如果所述多个设备生成新参数中的任何一个无效,或者如果所述聚合设备生成MAC相对于所述第一聚合服务器生成MAC无效,则所述处理器还用于:将所述多个M2M设备添加到失败集中。

13. 根据权利要求11所述的服务器,其特征在于,如果所述多个设备生成新参数中的任何一个无效,则所述处理器还用于:将所述多个M2M设备中的任何具有无效的设备生成新参数的M2M设备添加到失败集中;从所述网关获得所述多个认证消息的子集,所述子集对应于所述多个M2M设备中的未包含在所述失败集中的剩余部分;以及在非聚合的基础上验证包含在所述多个认证消息的所述子集中的所述第一多个设备生成MAC的有效性,

如果包含在所述多个认证消息的所述子集中的所述第一多个设备生成MAC中的任何一个有效,则所述服务器还用于:将所述多个M2M设备中的具有有效的设备生成MAC的对应第一子集添加到成功集中,以及

如果包含在所述多个认证消息的所述子集中的所述第一多个设备生成MAC中的任何一个无效,则所述服务器还用于:将所述多个M2M设备中的具有无效的设备生成MAC的对应第二子集添加到所述失败集中。

14. 根据权利要求11所述的服务器,其特征在于,如果所述多个设备生成新参数中的任何一个无效,则所述处理器还用于:

将所述多个M2M设备中的任何具有无效的设备生成新参数的M2M设备添加到失败集中;

从所述网关接收布隆过滤器,所述布隆过滤器在所述网关处基于所述多个认证消息中的至少一些而创建,所述多个认证消息中的至少一些与所述多个M2M设备中的未包含在所述失败集中的剩余部分对应;

计算所述多个M2M设备的第三多个服务器生成MAC;

在所述布隆过滤器中查询所述第三多个服务器生成MAC;

如果所述第三多个设备生成MAC的第一子集存在于所述布隆过滤器中,则将与所述第三多个设备生成MAC的所述第一子集对应的所述多个M2M设备的第一子集添加到临时成功集(temporary success set, TSS)中;

如果所述第三多个设备生成MAC的第二子集不在所述布隆过滤器中,则将与所述第三多个设备生成MAC的所述第二子集对应的所述多个M2M设备的第二子集添加到临时失败集(temporary failure set, TFS)中;

验证临时聚合设备生成MAC相对于包含在所述临时成功集(temporary success set, TSS)中的所述多个M2M设备的所述第一子集的第四多个服务器生成MAC的有效性,其中所述临时聚合设备生成MAC是从所述网关接收的;

如果所述临时聚合设备生成MAC有效，则将所述多个M2M设备的所述第一子集添加到所述成功集中；

如果所述临时聚合设备生成MAC无效，则将所述多个M2M设备的所述第一子集添加到所述失败集中，并将所述多个M2M设备的所述第二子集添加到所述失败集中。

15. 根据权利要求11所述的服务器，其特征在于，所述聚合设备生成MAC通过以下方式计算：对所述第一多个设备生成MAC执行压缩函数。

16. 根据权利要求11所述的服务器，其特征在于，所述多个新参数包括时间戳。

17. 根据权利要求10至16中的任一项所述的服务器，其特征在于，所述处理器还用于：在从所述网关接收所述聚合一认证消息之前，与所述网关进行相互认证，并在它们之间建立第一安全通道。

18. 根据权利要求10至16中的任一项所述的服务器，其特征在于，所述处理器还用于：在从所述网关接收所述聚合一认证消息之前，与所述网关进行相互认证，并在所述服务器与所述网关之间建立第一安全通道，其中所述网关已与所述多个M2M设备进行相互认证并且它们之间已建立多个第二安全通道。

19. 一种用于对服务器进行认证的方法，其特征在于，所述方法包括：

在机对机(Machine to Machine, M2M)设备处，使用基于对称密钥的技术生成用于服务器认证的认证消息；

向网关发送所述认证消息以与来自多个其它M2M设备的多个其它认证消息聚合，从而生成用于服务器认证的聚合一认证消息；

从所述网关接收服务器生成的认证响应，所述认证响应是对所述聚合一认证消息进行服务器认证而得到的多个认证响应之一；以及

使用基于对称密钥的技术，基于所述服务器生成的认证响应对所述服务器进行设备认证。

20. 根据权利要求19所述的方法，其特征在于，基于所述服务器生成的认证响应对所述服务器进行设备认证包括：

验证包含在所述服务器生成的认证响应中的服务器生成的新参数的有效性；

验证服务器生成的消息认证码(Message Authentication Code, MAC)相对于设备生成的MAC的有效性；以及

如果所述服务器生成的新参数有效，并且所述服务器生成的MAC相对于所述设备生成的MAC有效，则通过所述网关在所述M2M设备与所述服务器之间建立安全通道。

21. 根据权利要求19或20中的任一项所述的方法，其特征在于，还包括：

在向网关发送所述认证消息之前，对所述M2M设备与所述网关进行相互认证，并在它们之间建立安全通道。

22. 一种机对机(Machine-to-Machine, M2M)设备，其特征在于，包括：

处理器；以及存储设备，其存储供所述处理器执行的计算机可读指令，其中所述处理器用于：使用基于对称密钥的技术生成用于服务器认证的认证消息；向网关发送所述认证消息以与来自多个其它M2M设备的多个其它认证消息聚合，从而生成用于服务器认证的聚合一认证消息；从所述网关接收服务器生成的认证响应，所述认证响应是对所述聚合一认证消息进行服务器认证而得到的多个认证响应之一；以及使用基于对称密钥的技术，基于所述服

务器生成的认证响应对所述服务器进行设备认证。

23. 根据权利要求22所述的M2M设备，其特征在于，所述处理器还用于：验证包含在所述服务器生成的认证响应中的服务器生成的新参数的有效性，验证服务器生成的消息认证码(Message Authentication Code, MAC)相对于设备生成的MAC的有效性，以及

如果所述服务器生成的新参数有效，并且所述服务器生成的MAC相对于所述设备生成的MAC有效，则通过所述网关在所述M2M设备与所述服务器之间建立安全通道。

24. 根据权利要求22和23中的任一项所述的M2M设备，其特征在于，所述处理器还用于：在向网关发送所述认证消息之前，对所述M2M设备与所述网关进行相互认证，并在它们之间建立安全通道。

25. 一种用于对多个机对机(Machine-to-Machine, M2M)设备与一个服务器进行相互认证的方法，其特征在于，所述方法包括：

使用基于对称密钥的技术，基于分别从所述多个M2M设备接收的多个认证消息，在网关处生成聚合认证消息；

使用基于对称密钥的技术，基于从所述网关接收的所述聚合认证消息对所述多个M2M设备进行服务器认证；

在所述服务器处生成分别发往所述多个M2M设备的多个认证响应；以及

通过所述网关分别向所述多个M2M设备发送所述多个认证响应，其中所述认证响应中的至少一些指定用于所述服务器的设备认证，所述设备认证将使用基于对称密钥的技术在所述多个M2M设备中的成功进行服务器认证的M2M设备处执行。

26. 根据权利要求25所述的方法，其特征在于，在网关处生成聚合认证消息包括：

从包含在所述多个认证消息中的第一多个设备生成消息认证码(Message Authentication Code, MAC)计算聚合设备生成MAC，

对所述多个设备进行服务器认证包括：

验证包含在所述聚合认证消息中的多个设备生成新参数的有效性，以及

验证所述聚合设备生成MAC相对于第一聚合服务器生成MAC的有效性，所述第一聚合服务器生成MAC是从所述多个M2M设备的第一多个服务器生成MAC计算出的，

在所述服务器处生成多个认证响应包括：

计算所述多个M2M设备的第二多个服务器生成MAC，其中所述第二多个服务器生成MAC包含在所述多个认证响应中，用于验证相对于所述多个M2M设备中成功进行服务器认证的M2M设备的多个第二设备生成MAC的有效性。

27. 根据权利要求26所述的方法，其特征在于，如果所述多个设备生成新参数中的任何一个无效，或者如果所述聚合设备生成MAC相对于所述第一聚合服务器生成MAC无效，则所述方法还包括：

将所述多个M2M设备添加到失败集中。

28. 根据权利要求26所述的方法，其特征在于，如果所述多个设备生成新参数中的任何一个无效，则所述方法还包括：

将所述多个M2M设备中的任何具有无效的设备生成新参数的M2M设备添加到失败集中；

从所述网关获得所述多个认证消息的子集，所述子集对应于所述多个M2M设备中的未包含在所述失败集中的剩余部分；以及

在非聚合的基础上验证包含在所述多个认证消息的所述子集中的所述第一多个设备生成MAC的有效性；

如果包含在所述多个认证消息的所述子集中的所述第一多个设备生成MAC中的任何一个有效，则将所述多个M2M设备中的具有有效的设备生成MAC的对应第一子集添加到成功集中；以及

如果包含在所述多个认证消息的所述子集中的所述第一多个设备生成MAC中的任何一个无效，则将所述多个M2M设备中的具有无效的设备生成MAC的对应第二子集添加到所述失败集中。

29. 根据权利要求26所述的方法，其特征在于，如果所述多个设备生成新参数中的任何一个无效，则所述方法还包括：

将所述多个M2M设备中的任何具有无效的设备生成新参数的M2M设备添加到失败集中；

基于所述多个认证消息中的一个或多个在所述网关处创建布隆过滤器(bf)，所述多个认证消息中的一个或多个与所述多个M2M设备中的未包含在所述失败集中的剩余部分对应；

向所述服务器发送所述布隆过滤器；

计算所述多个M2M设备的第三多个服务器生成MAC；

在所述布隆过滤器中查询所述第三多个服务器生成MAC；

如果所述第三多个设备生成MAC的第一子集存在于所述布隆过滤器中，则将与所述第三多个设备生成MAC的所述第一子集对应的所述多个M2M设备的第一子集添加到临时成功集(temporary success set, TSS)中；

如果所述第三多个设备生成MAC的第二子集不在所述布隆过滤器中，则将与所述第三多个设备生成MAC的所述第二子集对应的所述多个M2M设备的第二子集添加到临时失败集(temporary failure set, TFS)中；

在所述网关处计算包含在所述临时成功集(temporary success set, TSS)中的所述多个M2M设备的所述第一子集的临时聚合设备生成MAC；

在所述服务器处验证所述临时聚合设备生成MAC相对于包含在所述临时成功集(temporary success set, TSS)中的所述多个M2M设备的所述第一子集的第四多个服务器生成MAC的有效性；

如果所述临时聚合设备生成MAC有效，则将所述多个M2M设备的所述第一子集添加到所述成功集中；以及

如果所述临时聚合设备生成MAC无效，则将所述多个M2M设备的所述第一子集添加到所述失败集中，并将所述多个M2M设备的所述第二子集添加到所述失败集中。

30. 根据权利要求26所述的方法，其特征在于，从第一多个设备生成MAC计算聚合设备生成MAC包括：

对所述第一多个设备生成MAC执行压缩函数。

31. 根据权利要求26所述的方法，其特征在于，所述多个新参数包括时间戳。

32. 根据权利要求25至31中的任一项所述的方法，其特征在于，还包括：

在向所述服务器发送所述聚合并认证消息之前，对所述网关与所述服务器进行相互认证，并在它们之间建立第一安全通道。

33. 根据权利要求25至31中的任一项所述的方法,其特征在于,还包括:

在所述网关处接收所述多个认证消息之前,对所述网关与所述多个M2M设备中的每一个进行相互认证,并在它们之间建立第二安全通道。

34. 根据权利要求25至31中的任一项所述的方法,其特征在于,还包括:

在向所述服务器发送所述聚合认证消息之前,对所述网关与所述服务器进行相互认证,并在它们之间建立第一安全通道;以及

在所述网关处接收所述多个认证消息之前,对所述网关与所述多个设备中的每一个进行相互认证,并在它们之间建立第二安全通道。

35. 一种用于对多个机对机(Machine-to-Machine,M2M)设备进行相互认证的系统,其特征在于,所述系统包括:网关和服务器,其中所述网关用于:使用基于对称密钥的技术,基于分别从所述多个M2M设备接收的多个认证消息,在网关处生成聚合认证消息,

其中所述服务器用于:使用基于对称密钥的技术,基于从所述网关接收的所述聚合认证消息对所述多个M2M设备进行认证;生成分别发往所述多个M2M设备的多个认证响应,

其中所述网关还用于:分别向所述多个M2M设备发送所述多个认证响应,其中所述认证响应中的至少一些指定用于所述服务器的设备认证,所述设备认证将使用基于对称密钥的技术在所述多个M2M设备中的成功进行服务器认证的M2M设备处执行。

36. 根据权利要求35所述的系统,其特征在于,所述网关还用于:从包含在所述多个认证消息中的第一多个设备生成消息认证码(Message Authentication Code,MAC)计算聚合设备生成MAC,以及

所述服务器还用于:验证包含在所述聚合认证消息中的多个设备生成新参数的有效性;验证所述聚合设备生成MAC相对于第一聚合服务器生成MAC的有效性,所述第一聚合服务器生成MAC是从所述多个M2M设备的第一多个服务器生成MAC计算出的;计算所述多个M2M设备的第二多个服务器生成MAC,其中所述第二多个服务器生成MAC包含在所述多个认证响应中,用于验证相对于所述多个M2M设备中成功进行服务器认证的M2M设备的多个第二设备生成MAC的有效性。

37. 根据权利要求36所述的系统,其特征在于,如果所述多个设备生成新参数中的任何一个无效,或者如果所述聚合设备生成MAC相对于所述第一聚合服务器生成MAC无效,则所述服务器还用于:将所述多个M2M设备添加到失败集中。

38. 根据权利要求36所述的系统,其特征在于,如果所述多个设备生成新参数中的任何一个无效,则所述服务器还用于:将所述多个M2M设备中的任何具有无效的设备生成新参数的M2M设备添加到失败集中;从所述网关获得所述多个认证消息的子集,所述子集对应于所述多个M2M设备中的未包含在所述失败集中的剩余部分;以及在非聚合的基础上验证包含在所述多个认证消息的所述子集中的所述第一多个设备生成MAC的有效性,

如果包含在所述多个认证消息的所述子集中的所述第一多个设备生成MAC中的任何一个有效,则所述服务器还用于:将所述多个M2M设备中的具有有效的设备生成MAC的对应第一子集添加到成功集中,以及

如果包含在所述多个认证消息的所述子集中的所述第一多个设备生成MAC中的任何一个无效,则所述服务器还用于:将所述多个M2M设备中的具有无效的设备生成MAC的对应第二子集添加到所述失败集中。

39. 根据权利要求36所述的系统，其特征在于，如果所述多个设备生成新参数中的任何一个无效，则所述服务器还用于：将所述多个M2M设备中的任何具有无效的设备生成新参数的M2M设备添加到失败集中；

所述网关还用于：基于所述多个认证消息中的一个或多个创建布隆过滤器(bf)，所述多个认证消息中的一个或多个与所述多个M2M设备中的未包含在所述失败集中的剩余部分对应；以及向所述服务器发送所述布隆过滤器，

所述服务器还用于：计算所述多个M2M设备的第三多个服务器生成MAC；在所述布隆过滤器中查询所述第三多个服务器生成MAC；如果所述第三多个设备生成MAC的第一子集存在于所述布隆过滤器中，则将与所述第三多个设备生成MAC的所述第一子集对应的所述多个M2M设备的第一子集添加到临时成功集(temporary success set,TSS)中；如果所述第三多个设备生成MAC的第二子集不在所述布隆过滤器中，则将与所述第三多个设备生成MAC的所述第二子集对应的所述多个M2M设备的第二子集添加到临时失败集(temporary failure set,TFS)中，

所述网关还用于：计算包含在所述临时成功集(temporary success set,TSS)中的所述多个M2M设备的所述第一子集的临时聚合设备生成MAC；验证所述临时聚合设备生成MAC相对于包含在所述临时成功集(temporary success set,TSS)中的所述多个M2M设备的所述第一子集的第四多个服务器生成MAC的有效性，

所述服务器还用于：如果所述临时聚合设备生成MAC有效，则将所述多个M2M设备的所述第一子集添加到所述成功集中；如果所述临时聚合设备生成MAC无效，则将所述多个M2M设备的所述第一子集添加到所述失败集中，并将所述多个M2M设备的所述第二子集添加到所述失败集中。

40. 根据权利要求36所述的系统，其特征在于，所述网关还用于对所述第一多个设备生成MAC执行压缩函数。

41. 根据权利要求36所述的系统，其特征在于，所述多个新参数包括时间戳。

42. 根据权利要求35至41中的任一项所述的系统，其特征在于，所述网关还用于：在向所述服务器发送所述聚合认证消息之前，与所述服务器进行相互认证，并在它们之间建立第一安全通道。

43. 根据权利要求35至41中的任一项所述的系统，其特征在于，所述网关还用于：在接收所述多个认证消息之前，与所述多个M2M设备中的每一个进行相互认证，并在它们之间建立第二安全通道。

44. 根据权利要求35至41中的任一项所述的系统，其特征在于，所述网关还用于：在向所述服务器发送所述聚合认证消息之前，对所述网关与所述服务器进行相互认证，并在它们之间建立第一安全通道；以及在接收所述多个认证消息之前，对所述网关与所述多个设备中的每一个进行相互认证，并在它们之间建立第二安全通道。

## 用于M2M通信中的聚合认证协议的方法

### 技术领域

[0001] 本发明涉及一种机对机(Machine-to-Machine, M2M)通信系统,尤其涉及用于多个M2M设备与一个服务器之间的相互认证的方法。

### 背景技术

[0002] 机对机(Machine-to-Machine, M2M)通信在机器之间发生,带有计算和通信能力。已有很多应用具有M2M通信能力,例如用于个人健康监测、智能跟踪和供应链跟踪、智能电网、自动售货机远程控制等的设备。启用M2M的设备或终端的数量呈指数增长,预计到2014年会从2008年的5千万增加到超过2亿,到2020年会多达500亿。

[0003] 图1示出了典型的M2M系统架构,包括M2M设备、M2M网关和M2M认证服务器。在M2M系统中,M2M设备代表末端节点,例如传感器,M2M网关代表数据聚合节点,M2M认证服务器代表对M2M设备进行认证的M2M服务器。M2M网关提供网桥来将M2M设备可通信地耦合到M2M服务器,反之亦然。具体来说,M2M设备与M2M网关直接通信,M2M网关与M2M服务器直接通信。

[0004] 由于M2M通信的广泛使用,通信安全非常重要。一个安全问题是通过M2M认证服务器来认证M2M设备。随着4G(第四代移动通信)的使用和5G(第五代移动网络)的出现,连接设备的数量大幅倍增。作为4G和5G的代表性场景,M2M设备的数量会很庞大,当所有M2M设备在短时间内单独执行认证时,M2M认证服务器的工作量非常巨大。这可能导致认证服务器超负荷而崩溃。因此,急需一种有效且可扩展的认证协议。

[0005] 现有认证协议支持M2M认证服务器同时对一组M2M设备进行认证。

[0006] 一种现有组认证协议在“安全TS的组认证机制(Group Authentication Mechanism for Security TS)”OneM2M, SEC#11, SEC-2014-0314R02中描述并在图2中示出。图2的组认证包括下述三个阶段:

[0007] 1. 内组认证:内组中的所有M2M设备与M2M网关相互认证并建立安全通道以供进一步通信。

[0008] 2. 外组认证:M2M网关与M2M认证服务器相互认证并建立安全通道以供进一步通信。

[0009] 3. 建立端到端安全通道:M2M认证服务器通过已建立的安全通道,即M2M认证服务器与M2M网关之间的安全通道、M2M网关与各种M2M设备之间的安全通道,向所有M2M设备发送安全资料。基于安全资料以及M2M认证服务器与M2M设备之间的共享密钥,它们推导出与彼此通信的会话密钥,即,建立了端到端安全通道。

[0010] 在上述协议中,基于OneM2M通信架构提出了组认证,其中M2M设备称为应用服务节点或应用专用节点,M2M网关称为中间节点,M2M认证服务器称为基础设施节点。

[0011] 中国专利申请案CN 102223231 A中描述了另一种组认证协议。在该协议中,外组认证在第一阶段处理,内组认证在第二阶段处理。此外,该协议基于LTE架构,其中M2M设备称为用户设备(user equipment, UE),M2M认证服务器称为LTE核心网。

[0012] 在上述和现有协议中,存在一种重要的假设,即,M2M网关是可信的。例如,在图2的

组认证协议中，M2M认证服务器并不直接对M2M设备进行认证。M2M设备仅由M2M网关进行认证，M2M网关将认证成功的M2M设备通知给M2M认证服务器。然后，M2M认证服务器将认为这些M2M设备已认证，因为M2M网关是可信的。因此，M2M认证服务器无法直接对M2M设备进行认证；换言之，组认证协议不支持M2M认证服务器与M2M设备之间的端到端认证。

[0013] 该假设引发了一个安全问题。M2M网关可能并不是很难破坏，因为M2M网关通常部署在户外。如果攻击者破坏了M2M网关，则攻击者可以欺骗M2M认证服务器认为一些M2M设备成功认证，即使这些M2M设备并未成功认证。

## 发明内容

[0014] 本发明实施例提供了一种用于M2M认证服务器和M2M设备的端到端认证协议，其中所述M2M认证服务器和M2M设备直接相互认证。无需假设M2M网关可信。避免了使用组认证和组标识符。此外，相互认证协议采用基于对称密钥的技术。

[0015] 提供了一种用于通过服务器认证多个M2M设备的方法。所述方法包括：

[0016] 使用基于对称密钥的技术，基于从网关接收的聚合认证消息对所述多个M2M设备进行服务器认证，所述聚合认证消息基于从所述多个M2M设备分别接收的多个认证消息在所述网关处生成；以及

[0017] 生成多个认证响应，所述多个认证响应通过所述网关分别发往所述多个M2M设备，其中所述多个认证响应中的至少一些指定用于所述服务器的设备认证，所述设备认证将使用基于对称密钥的技术在所述多个M2M设备中的成功进行服务器认证的M2M设备处执行。

[0018] 提供了一种用于对多个M2M设备进行认证的服务器。所述服务器包括：处理器；以及存储设备，其存储供所述处理器执行的计算机可读指令，

[0019] 其中所述处理器用于：使用基于对称密钥的技术，基于从网关接收的聚合认证消息对所述多个M2M设备进行认证，所述聚合认证消息基于从所述多个M2M设备分别接收的多个认证消息在所述网关处生成；生成多个认证响应，所述多个认证响应通过所述网关分别发往所述多个M2M设备，其中所述多个认证响应中的至少一些指定用于所述服务器的设备认证，所述设备认证将使用基于对称密钥的技术在所述多个M2M设备中的成功进行服务器认证的M2M设备处执行。

[0020] 提供了一种用于对服务器进行认证的方法。所述方法包括：

[0021] 在所述M2M设备处，使用基于对称密钥的技术生成用于服务器认证的认证消息；

[0022] 向网关发送所述认证消息以与来自多个其它M2M设备的多个其它认证消息聚合，从而生成用于服务器认证的聚合认证消息；

[0023] 从所述网关接收服务器生成的认证响应，所述认证响应是对所述聚合认证消息进行服务器认证而得到的多个认证响应之一；以及

[0024] 使用基于对称密钥的技术，基于所述服务器生成的认证响应对所述服务器进行设备认证。

[0025] 提供了一种M2M设备，包括：

[0026] 处理器；以及存储设备，其存储供所述处理器执行的计算机可读指令。所述处理器用于：使用基于对称密钥的技术生成用于服务器认证的认证消息；向网关发送所述认证消息以与来自多个其它M2M设备的多个其它认证消息聚合，从而生成用于服务器认证的聚合

认证消息；从所述网关接收服务器生成的认证响应，所述认证响应是对所述聚合认证消息进行服务器认证而得到的多个认证响应之一；以及使用基于对称密钥的技术，基于所述服务器生成的认证响应对所述服务器进行设备认证。

[0027] 提供了一种用于对多个机对机 (Machine-to-Machine, M2M) 设备与一个服务器进行相互认证的方法。所述方法包括：

[0028] 使用基于对称密钥的技术，基于分别从所述多个M2M设备接收的多个认证消息，在网关处生成聚合认证消息；

[0029] 使用基于对称密钥的技术，基于从所述网关接收的所述聚合认证消息对所述多个M2M设备进行服务器认证；

[0030] 在所述服务器处生成分别发往所述多个M2M设备的多个认证响应；以及

[0031] 通过所述网关分别向所述多个M2M设备发送所述多个认证响应，其中所述认证响应中的至少一些指定用于所述服务器的设备认证，所述设备认证将使用基于对称密钥的技术在所述多个M2M设备中的成功进行服务器认证的M2M设备处执行。

[0032] 提供了一种用于对多个机对机 (Machine-to-Machine, M2M) 设备进行相互认证的系统。所述系统包括：网关和服务器，其中所述网关用于：使用基于对称密钥的技术，基于分别从所述多个M2M设备接收的多个认证消息，在网关处生成聚合认证消息，

[0033] 其中所述服务器用于：使用基于对称密钥的技术，基于从所述网关接收的所述聚合认证消息对所述多个M2M设备进行认证；生成分别发往所述多个M2M设备的多个认证响应，

[0034] 其中所述网关还用于：分别向所述多个M2M设备发送所述多个认证响应，其中所述认证响应中的至少一些指定用于所述服务器的设备认证，所述设备认证将使用基于对称密钥的技术在所述多个M2M设备中的成功进行服务器认证的M2M设备处执行。

## 附图说明

[0035] 下文参考图示公开本发明的实施例，在图示中：

[0036] 图1示出了典型的M2M系统架构；

[0037] 图2示出了现有组认证协议；

[0038] 图3为根据本发明的示出聚合认证方法的流程图；

[0039] 图4示出了根据本发明第一实施例的M2M设备与M2M服务器之间的聚合认证流程；

[0040] 图5示出了根据本发明第二实施例的M2M设备与M2M服务器之间的聚合认证流程；

[0041] 图6示出了根据本发明第三实施例的M2M设备与M2M服务器之间的聚合认证流程；

[0042] 图7示出了根据本发明第四实施例的M2M设备与M2M服务器之间的聚合认证流程；

[0043] 图8为示出用于认证设备并找到成功认证的设备的方法的流程图。

## 具体实施方式

[0044] 下文描述中陈述许多具体细节，以对本发明各实施例进行通彻理解。然而，本领域熟练技术人员将理解，可以在不具有这些具体细节中的一些或全部的情况下实践本发明的实施例。在其它情况下，为了不多余地混淆所描述的实施例的相关方面，并未详细地描述熟知的过程操作。在图式中，所有的若干张图以相同参考标号指代相同或相似的功能或特征。

[0045] 说明书和权利要求书中使用序数词“第一”、“第二”、“第三”等来描述公共元素，仅表明提及相似元素的不同实例，并非旨在暗示所描述的元素必须按照给定的时间、空间、排名顺序或以任何其它方式排列的顺序，除非另有说明。

[0046] 说明书和权利要求书中使用的本领域技术人员已知的若干概念在下文描述：

[0047] • 伪随机函数 (Pseudorandom Function, PRF) 是一种映射两个不同集合 (域和范围) 的有效确定性函数。一个PRF仅有一个输入d (域) 和一个隐藏随机种子 (范围)。当伪随机函数使用相同的输入和种子多次运行时，其始终输出相同的值。尽管如此，对于任意输入，由于随机种子，输出看起来是随机的。

[0048] • 密钥导出函数 (key derivation function, KDF)：密钥导出函数使用伪随机函数从主密钥等秘值或者密码或通行码等其它已知信息推导出一个或多个私密密钥。

[0049] • 消息认证码 (Message Authentication Code, MAC)：密码学中的消息认证码是用于对消息进行认证并提供消息的完整性和真实性保证的一小段信息。完整性保证检测偶然的和有意的消息变化，而真实性保证确认消息的来源。

[0050] • 布隆过滤器 (Bloom Filter, bf)：布隆过滤器是一种节省空间的数据结构，用于测试一个元素是否是集合成员。假正匹配是可能的，但是假负是不可能的。布隆过滤器是一个m比特的阵列bf，用于表示具有n个元素的集合 $S = \{x_1, x_2, \dots, x_n\}$ ，bf中的所有比特最初都设为零。存在k个不同的哈希函数 $h_i()$ ，每个哈希函数将某个集合元素映射到均匀随机分布的m个阵列位置中的一个。存在两个基本操作：添加和查询。

[0051] 要将元素x添加到过滤器bf中，仅需将比特 $bf[h_i(x)]$ 设为1，其中 $1 \leq i \leq k$ 。

[0052] 要查询元素y (测试y是否在集合中)，仅需测试是否所有比特 $bf[h_i(y)]$ 都为1。如果所有比特都为1，则认为元素y在集合中；如果任何一个比特为0，则该元素肯定不在集合中。

[0053] 本发明实施例提供了一种用于M2M认证服务器和M2M设备的端到端认证协议，其中M2M认证服务器和M2M设备直接相互认证。无需假设M2M网关可信。

[0054] 图3为根据本发明的示出聚合认证方法的流程图300。

[0055] 在步骤301中，在待认证的每个M2M设备 (下文称为“设备”) 处生成认证消息。

[0056] 在步骤302中，每个设备将其认证消息发送给M2M网关 (下文称为“网关”)。

[0057] 在步骤303中，网关从各设备接收认证消息。网关使用基于对称密钥的技术从接收的认证消息生成一个或多个聚合认证消息。该聚合可在以下情况之后发生：预配置的时间间隔已过去，或者从网络中的所有设备收到认证消息，或者收到一定数量的认证消息，或者满足其它标准，或者以上几种情况的组合。

[0058] 在步骤304中，网关向M2M认证服务器 (下文称为“服务器”) 发送聚合认证消息。

[0059] 在步骤305中，服务器使用基于对称密钥的技术，基于聚合认证消息对各设备进行认证。

[0060] 在步骤306中，服务器基于步骤305的认证结果生成响应集，响应集包含服务器生成的分别发往各设备的认证响应。

[0061] 在步骤307中，服务器向网关发送响应集。

[0062] 在步骤308中，网关向每个设备发送服务器生成的发往该特定设备的对应认证响应。

[0063] 在步骤309中,在任一设备中,如果接收的服务器生成的对应认证响应包含错误信息,则表明该特定设备在步骤305中的服务器认证已失败。因此,该特定设备的认证流程停止或中断。在任一设备中,如果接收的服务器生成的对应认证响应不包含错误信息,或者该认证响应包含成功指示,则表明该特定设备在步骤305中的服务器认证已成功,因此流程顺序继续使用基于对称密钥的技术在每个成功认证的设备处对服务器进行设备认证。

[0064] 在步骤310中,对于任意设备,如果设备(在步骤305中)的服务器认证和服务器(在步骤309中)的设备认证都成功,则通过网关在该设备与服务器之间建立安全通道。

[0065] 从上述内容显而易见,步骤301至304涉及各设备基于聚合认证消息的服务器认证;随后的步骤305至309涉及服务器的设备认证,服务器的设备认证在已由服务器在步骤305中成功认证的每个设备处分别进行。

[0066] 图4示出了根据本发明第一实施例的M2M设备与M2M服务器之间的聚合认证流程。该流程基于的是基于对称密钥的技术。

[0067] 本实施例的先决条件包括:M2M认证服务器和M2M设备*i*彼此共享密钥K<sub>i</sub>和设备标识ID<sub>i</sub>;存在n个待认证的M2M设备。

[0068] 在步骤401中,在待认证的每个设备*i*处生成认证消息。具体来说,具有标识ID<sub>i</sub>的设备*i*选择新参数P<sub>i</sub>,例如时间戳,并计算临时密钥K<sub>i1</sub>=KDF(K<sub>i</sub>,P<sub>i</sub>)和设备生成的MAC T<sub>i</sub>=FunMAC(K<sub>i1</sub>,ID<sub>i</sub>),其中FunMAC是MAC算法,例如,FunMAC是AES-CMAC算法。认证消息包括ID<sub>i</sub>、P<sub>i</sub>和T<sub>i</sub>。

[0069] 在步骤402中,每个设备*i*向网关发送各自的认证消息(ID<sub>i</sub>,P<sub>i</sub>,T<sub>i</sub>)。

[0070] 在步骤403中,在从所有设备接收认证消息{(ID<sub>1</sub>,P<sub>1</sub>,T<sub>1</sub>),…,(ID<sub>n</sub>,P<sub>n</sub>,T<sub>n</sub>)}后,网关将接收的MAC{T<sub>1</sub>,…,T<sub>n</sub>}聚合为新的或聚合MAC T=CFun(T<sub>1</sub>,…,T<sub>n</sub>),其中CFun是压缩函数,例如T=T<sub>1</sub>⊕…⊕T<sub>n</sub>,或者T=hash(T<sub>1</sub>||…||T<sub>n</sub>),其中hash是密码哈希函数。输入是多个字符串的压缩函数输出长度较短的单个字符串。这里,为了保证聚合认证协议的安全,可采用T=T<sub>1</sub>⊕…⊕T<sub>n</sub>或T=hash(T<sub>1</sub>||…||T<sub>n</sub>)。该聚合使用基于对称密钥的技术来执行。

[0071] 在步骤404中,网关向服务器发送聚合认证消息({(ID<sub>i</sub>,P<sub>i</sub>)},T)。该聚合认证消息包括n个设备的ID<sub>i</sub>和P<sub>i</sub>以及聚合MAC T。

[0072] 在步骤405中,服务器基于聚合认证消息,使用基于对称密钥的技术来执行n个设备的认证。具体来说,服务器找到成功认证的设备,将这些成功设备的标识ID添加到成功集SS中。对于服务器认证失败的那些设备,将标识ID添加到失败集FS中。在说明书和权利要求书中将认识到,提到将设备添加到成功集SS或失败集FS中就等同于将设备的标识ID添加到成功集SS或失败集FS中。稍后将结合图8描述一种用于认证设备并找到成功认证的设备的方法。

[0073] 在步骤406中,服务器基于步骤405的认证结果生成响应集RI=(SI,FI),其中RI包括服务器生成的多个认证响应,SI包括针对成功集SS中的所有设备的响应信息,FI包括针对失败集FS中的所有设备的响应信息:

[0074] • SI={ (ID<sub>i</sub>,P' <sub>i</sub>,T' <sub>i</sub>) },其中ID<sub>i</sub>在SS中,P' <sub>i</sub>是新参数,K' <sub>i1</sub>=KDF(K<sub>i</sub>,P' <sub>i</sub>),服务器生成的MAC T' <sub>i</sub>=FunMAC(K' <sub>i1</sub>,ID<sub>i</sub>)。针对设备ID<sub>i</sub>计算会话密钥K' <sub>i</sub>=KDF(K<sub>i</sub>,P<sub>i</sub>⊕P' <sub>i</sub>),其中KDF是密钥导出函数。

[0075] • FI={ (ID<sub>j</sub>,AEL<sub>j</sub>,P' <sub>j</sub>,T' <sub>j</sub>) },其中ID<sub>j</sub>在FS中,AEL<sub>j</sub>是设备ID<sub>j</sub>的认证错误信息,P' <sub>j</sub>

是新参数,  $K'_{j1} = KDF(K_j, P'_{j1})$ , 服务器生成的MAC  $T'_{j1} = \text{FunMAC}(K'_{j1}, ID_j || AEL_j)$ 。

[0076] 在步骤407中, 服务器向网关发送响应集  $RI = (SI, FI)$ 。

[0077] 在步骤408中, 网关向每个设备  $i$  发送响应集中的服务器生成的对应认证响应  $(ID_i, P'_i, T'_i)$  或  $(ID_i, AEL_i, P'_i, T'_i)$ 。

[0078] 步骤409适用于失败集中包含的设备, 即, 步骤405中服务器认证失败的设备。在步骤409中, 如果在一个设备处接收的服务器生成的认证响应是  $(ID_i, AEL_i, P'_i, T'_i)$ , 其中错误信息是  $AEL_i$ , 则该设备的认证失败, 方法停止。

[0079] 步骤410适用于成功集中包含的设备, 即, 步骤405中认证成功的设备。在步骤410中, 使用基于对称密钥的技术在每个成功进行服务器认证的设备处执行服务器的认证。

[0080] 具体来说, 在每个成功认证的设备处, 设备确定服务器生成的认证响应  $(ID_i, P'_i, T'_i)$  中包含的  $P'_i$  和  $T'_i$  是否都有效。如果  $P'_i$  和  $T'_i$  中的任意一个无效, 则认证失败。如果  $P'_i$  和  $T'_i$  都有效, 则设备计算会话密钥  $K'_i = KDF(K_i, P_i \oplus P'_i)$ 。

[0081] 用于检查新参数  $P_i$  的有效性的方法取决于在认证中使用的新参数的类型。例如, 如果新参数是时间戳, 则当  $P_i$  不在预先确定的范围内时,  $P_i$  是无效的。

[0082] 用于确定MAC  $T'_i$  的有效性的方法描述如下:

[0083] (i) 计算  $K^a_{i1} = KDF(K_i, P'_i)$  和设备生成的MAC  $T^a_{i1} = \text{FunMAC}(K^a_{i1}, ID_i)$ 。

[0084] (ii) 设备确定服务器生成的MAC  $T'_i$  与设备生成的MAC  $T^a_{i1}$  是否相同。如果服务器生成的MAC  $T'_i$  与设备生成的MAC  $T^a_{i1}$  相同, 则确定服务器生成的MAC  $T'_i$  有效。如果服务器生成的MAC  $T'_i$  与设备生成的MAC  $T^a_{i1}$  不同, 则确定服务器生成的MAC  $T'_i$  无效。

[0085] 图5示出了根据本发明第二实施例的M2M设备与M2M服务器之间的聚合认证流程。第二实施例与第一实施例相似。第二实施例还包括使网关与服务器共享私密密钥  $K'$ 。网关在向服务器发送聚合认证消息之前的任意时间, 即, 在步骤404之前的任意时间, 执行与认证服务器的相互认证, 并使用共享密钥  $K'$  在它们之间建立安全通道。在相互认证之后, 网关与服务器之间的所有通信, 即步骤404、407等, 都将通过该安全通道进行。

[0086] 图6示出了根据本发明第三实施例的M2M设备与M2M服务器之间的聚合认证流程。第三实施例还包括使每个设备与网关共享私密密钥  $K''_i$ 。每个设备在向网关发送认证消息之前的任意时间, 即, 在步骤402之前的任意时间, 执行与网关的相互认证, 并使用共享密钥  $K''_i$  在它们之间建立安全通道。在相互认证之后, 设备与网关之间的所有通信, 即步骤402、408等, 都将通过对应安全通道进行。

[0087] 图7示出了根据本发明第四实施例的M2M设备与M2M服务器之间的聚合认证流程。第四实施例可视为第二实施例与第三实施例的组合。

[0088] 在第四实施例中, 网关与服务器共享私密密钥  $K'$ 。网关在向服务器发送聚合认证消息之前的任意时间, 即, 在步骤404之前的任意时间, 执行与认证服务器的相互认证, 并使用共享密钥  $K'$  在它们之间建立安全通道。在相互认证之后, 网关与服务器之间的所有通信, 即步骤404、407等, 都将通过该安全通道进行。

[0089] 在第四实施例中, 每个设备与网关共享私密密钥  $K''_i$ 。每个设备在向网关发送认证消息之前的任意时间, 即, 在步骤402之前的任意时间, 执行与网关的相互认证, 并使用共享密钥  $K''_i$  在它们之间建立安全通道。在相互认证之后, 设备与网关之间的所有通信, 即步骤402、408等, 都将通过对应安全通道进行。

[0090] 图8为示出用于认证设备并找到成功认证的设备的方法的流程图,该方法可在上述步骤405中执行。

[0091] 图8的方法的先决条件包括:服务器已收到聚合认证消息( $\{(ID_i, P_i)\}, T$ )。

[0092] 在步骤801中,服务器创建两个空集,即成功集SS和失败集FS,它们将分别存储成功进行服务器认证的设备的标识ID和未成功进行服务器认证的设备的标识ID。在说明书和权利要求书中,提到将成功进行服务器认证的设备存储在成功集SS中就等同于将成功进行服务器认证的设备的标识ID存储在成功集SS中;提到将未成功进行服务器认证的设备存储在失败集FS中就等同于将未成功进行服务器认证的设备的标识ID存储在失败集FS中。

[0093] 在步骤802中,服务器基于聚合认证消息( $\{(ID_i, P_i)\}, T$ )确定 $P_i$ 是否对所有设备 $i$ 都有效。如果确定 $P_i$ 无效,则将该设备的标识 $ID_i$ 添加到FS。

[0094] 在步骤803中,服务器确定失败集FS是否为空。如果失败集FS不为空,则流程顺序前进到步骤804。如果失败集FS为空,则流程前进到步骤806。

[0095] 在步骤804中,服务器可选择一种用于无效新参数的错误处理方法。服务器可根据其它条件选择前进到用于处理参数错误的第一方法(步骤804)还是用于处理参数错误的第二方法(步骤806)。

[0096] 在步骤805中,用于处理参数错误的第一方法包括将所有设备的标识ID添加到失败集FS中,然后流程顺序前进到步骤811,在步骤811中,图8的流程顺序结束。

[0097] 在步骤806中,服务器确定聚合设备生成MAC T是否有效。确定聚合设备生成MAC T的有效性的流程描述如下:

[0098] (i) 针对所有设备 $i$ 计算 $K^b_{i1}=KDF(K_i, P_i)$  和 $T^b_i=FunMAC(K^b_{i1}, ID_i)$ ,并将服务器生成的所有MAC  $T^b_i$ 聚合为聚合MAC  $T^b$ 。

[0099] (ii) 服务器确定聚合设备生成MAC T与聚合服务器生成MAC  $T^b$ 是否相同。如果聚合设备生成MAC T与聚合服务器生成MAC  $T^b$ 相同,则确定聚合设备生成MAC T有效。如果聚合设备生成MAC T与聚合服务器生成MAC  $T^b$ 不同,则确定聚合服务器生成MAC T无效。

[0100] 如果确定聚合设备生成MAC T有效,则流程顺序前进到步骤811,在步骤811中,图8的流程顺序结束。如果聚合设备生成MAC T无效,则流程顺序前进到步骤807。

[0101] 在步骤807中,服务器可选择一种用于无效MAC的错误处理方法。服务器可根据其它条件选择前进到用于处理MAC错误的第一方法(步骤808)还是用于处理MAC错误的第二方法(步骤809)还是用于处理MAC错误的第三方法(步骤810)。

[0102] 在步骤808中,用于处理MAC错误的第一方法包括将所有设备的标识ID添加到失败集FS中,然后流程顺序前进到步骤811,在步骤811中,图8的流程顺序结束。

[0103] 在步骤809中,用于处理MAC错误的第二方法需要在服务器与网关之间通信。第二MAC错误处理流程描述如下:

[0104] (i) 服务器向网关发送消息以请求未包含在失败集FS中的所有设备的认证信息。

[0105] (ii) 网关向认证服务器发送详细的认证信息 $\{(ID_i, P_i, T_i)\}$ 。

[0106] (iii) 服务器分别或在非聚合的基础上确定设备生成的MAC  $T_i$ 的有效性。如果 $T_i$ 无效,则将对应设备的标识 $ID_i$ 添加到失败集FS中;如果 $T_i$ 有效,则将设备标识 $ID_i$ 添加到成功集SS中。

[0107] 为了确定 $T_i$ 的有效性,服务器首先计算 $K'_{i1}=KDF(K_i, P_i)$  和 $T'_{i1}=MAC(K'_{i1}, ID_i)$ ,然

后确定设备生成的MAC  $T_i$ 与服务器生成的MAC  $T'_i$ 是否相同。如果设备生成的MAC  $T_i$ 与服务器生成的MAC  $T'_i$ 相同，则确定 $T_i$ 有效。如果设备生成的MAC  $T_i$ 与服务器生成的MAC  $T'_i$ 不同，则确定服务器生成的MAC  $T'_i$ 无效。

[0108] 当步骤809完成时，流程顺序前进到步骤811，在步骤811中，图8的流程顺序结束。

[0109] 在步骤810中，用于处理MAC错误的第三方法需要在服务器与网关之间通信。该MAC错误处理流程描述如下：

[0110] (i) 对于未包含在失败集FS中的设备，服务器向网关发送对基于布隆过滤器的对应认证信息的请求。

[0111] (ii) 网关基于认证信息 $\{(ID_i, P_i, T_i)\}$ 创建布隆过滤器bf，其中 $ID_i$ 未包含在FS中，即，将所有 $T_i$ 添加到布隆过滤器bf中，其中的所有比特最初都为0。

[0112] (iii) 网关向服务器发送创建的布隆过滤器bf。

[0113] (iv) 服务器创建两个临时集合：临时成功集TSS和临时失败集TFS。针对所有设备 $i$ 计算 $K_{i1}^c = KDF(K_i, P_i)$ 和 $T_i^c = \text{FunMAC}(K_{i1}^c, ID_i)$ ，并在bf中查询 $T_i^c$ 。如果服务器生成的MAC  $T_i^c$ 存在于bf中，则将对应的设备标识 $ID_i$ 添加到TSS中。如果服务器生成的MAC  $T_i^c$ 不在bf中，则将对应的设备标识 $ID_i$ 添加到TFS中。

[0114] (v) 服务器向网关发送TSS。

[0115] (vi) 对于TSS中的所有设备标识 $ID_i$ ，网关使用与图4的步骤403中相同的方法基于设备生成的MAC  $T_i$ 来计算聚合 $T'$ 。

[0116] (vii) 网关向服务器发送聚合MAC  $T'$ 。

[0117] (viii) 服务器检查聚合MAC  $T'$ 是否有效。如果聚合MAC  $T'$ 有效，则将TSS中的所有设备标识和其它信息都添加到SS中，将TFS中的所有设备标识和其它信息都添加到FS中。如果聚合MAC  $T'$ 无效，则将TSS和TFS中的所有设备标识和其它信息都添加到FS中。

[0118] 确定 $T'$ 是否有效的方法与步骤806相似，但是仅针对TSS中的设备标识 $ID_i$ 。当步骤810完成时，流程顺序前进到步骤811，在步骤811中，图8的流程顺序结束。

[0119] 步骤811引出图4的步骤406，其中聚合认证如图4的流程顺序中所描述的那样继续。

[0120] 虽然图8包括各种处理新参数错误和处理MAC错误的方法，但是可以认识到，在本发明的某些实施方式中，可在结合或不结合其它可能的错误处理方法的情况下使用上述部分或全部错误处理方法。

[0121] 通过本发明，认证服务器与一组设备进行相互认证，并为每个设备建立独有的会话密钥。因此，本发明提供的优势包括但不限于以下：

[0122] (i) 在本发明中，服务器仅基于聚合认证消息对多个设备进行认证。因此，服务器的认证工作量，包括通信和计算开销，得到明显降低。

[0123] (ii) 此外，服务器接收的认证消息是聚合来自各设备的认证消息而得到的。因此，网关不可能伪造聚合认证消息来欺骗服务器，所以不要求网关节点是可信的。

[0124] 本发明的实施例可应用到任何包括设备(或终端)、网关和认证服务器的通信系统。通信系统可以是固定网络或移动网络。例如，如果本发明应用于电信网络，则运营商网络可以通过网关对多个用户设备(user equipment, UE)进行认证。

[0125] 本领域熟练技术人员根据对本说明书的考量和对本发明的实践将清楚其它实施

例。此外,出于描述明确性的目的使用了某些术语且这些术语不会限制本发明的所揭示实施例。上文描述的实施例和特征应被视为示例性的。

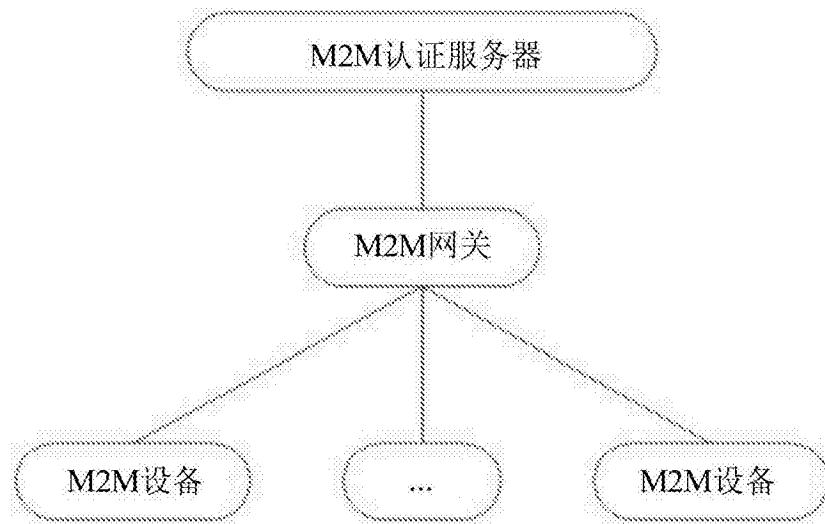


图1

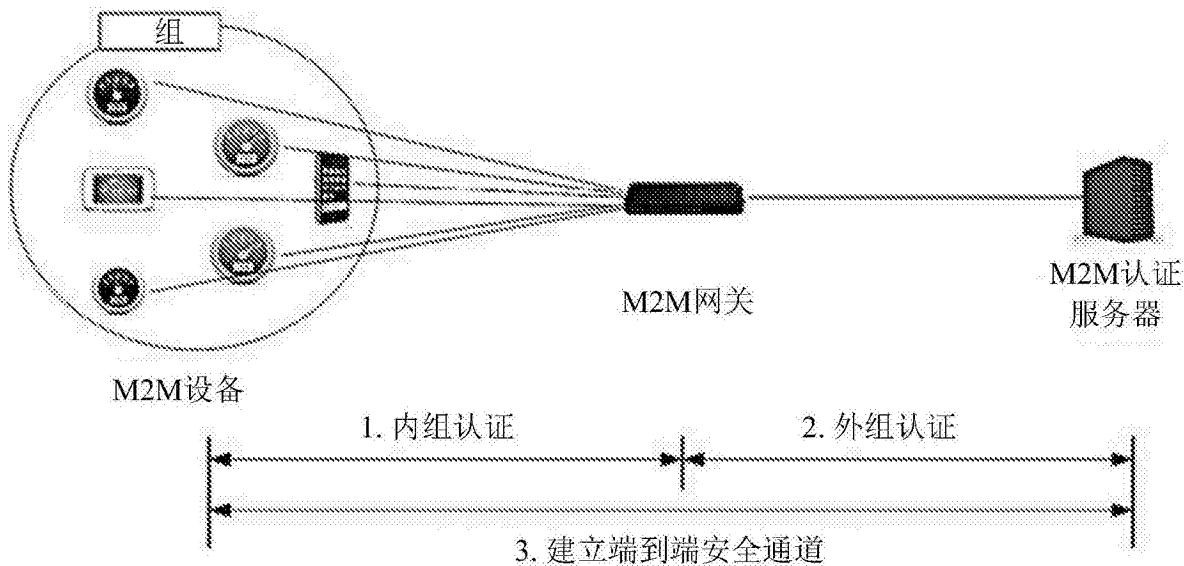


图2 (现有技术)

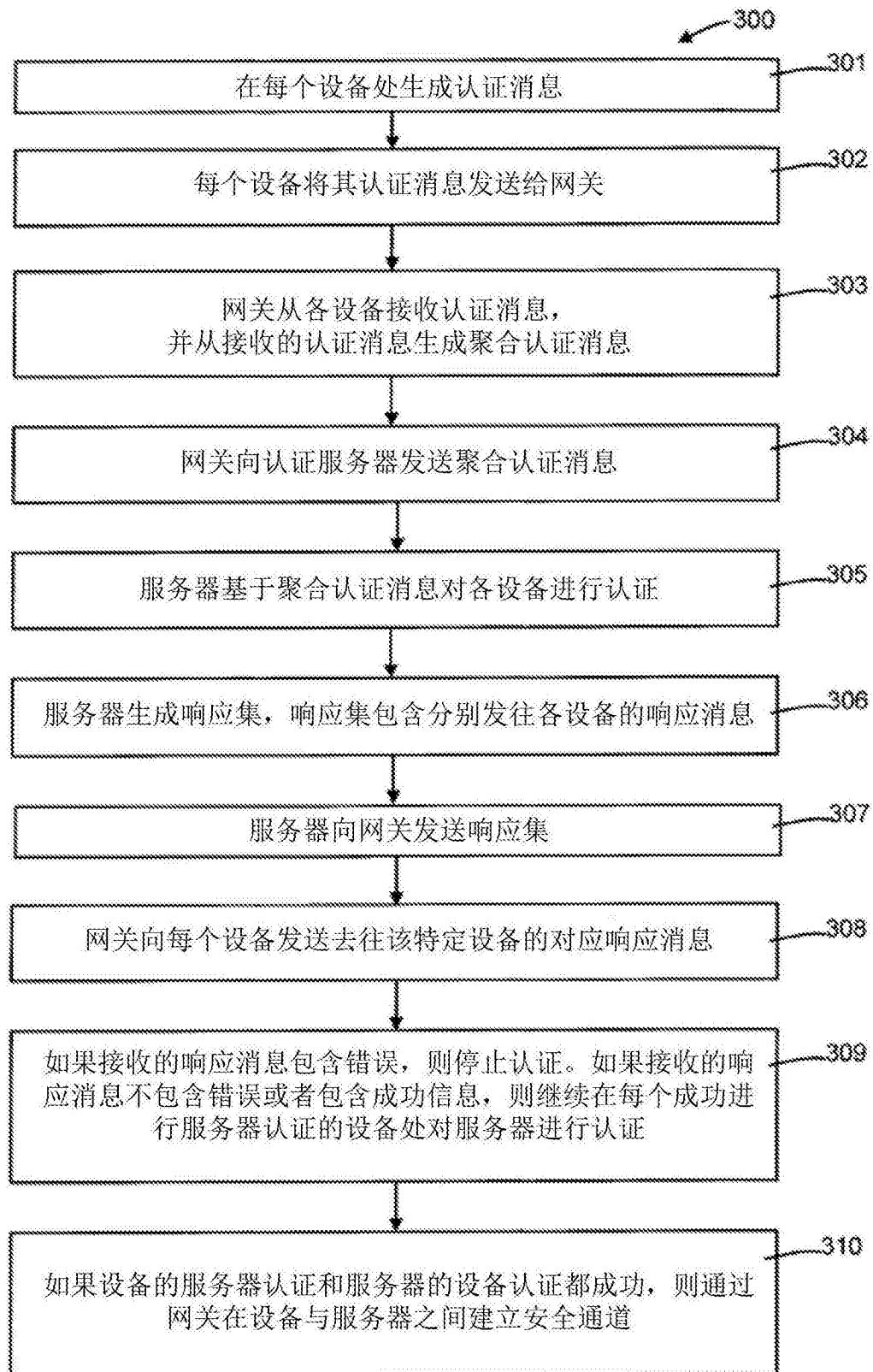


图3

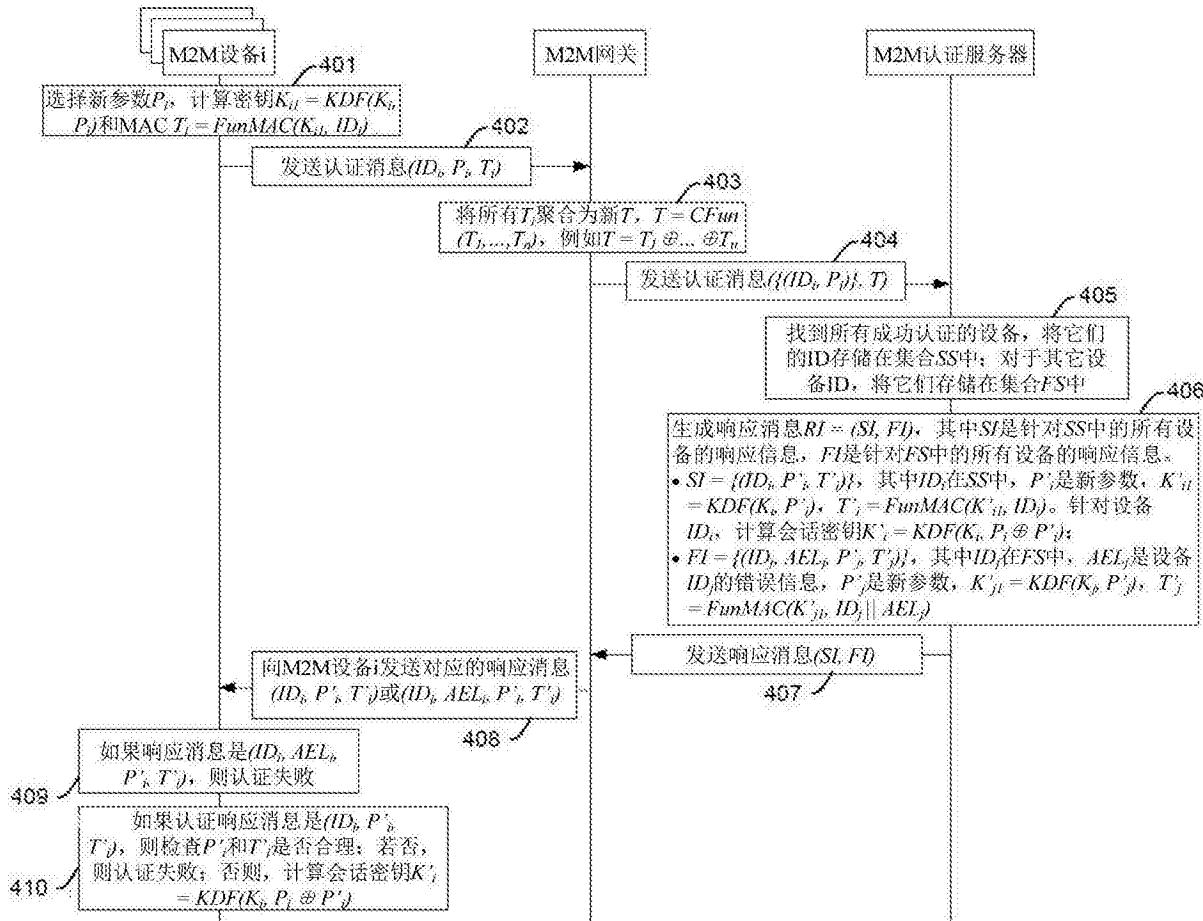


图4

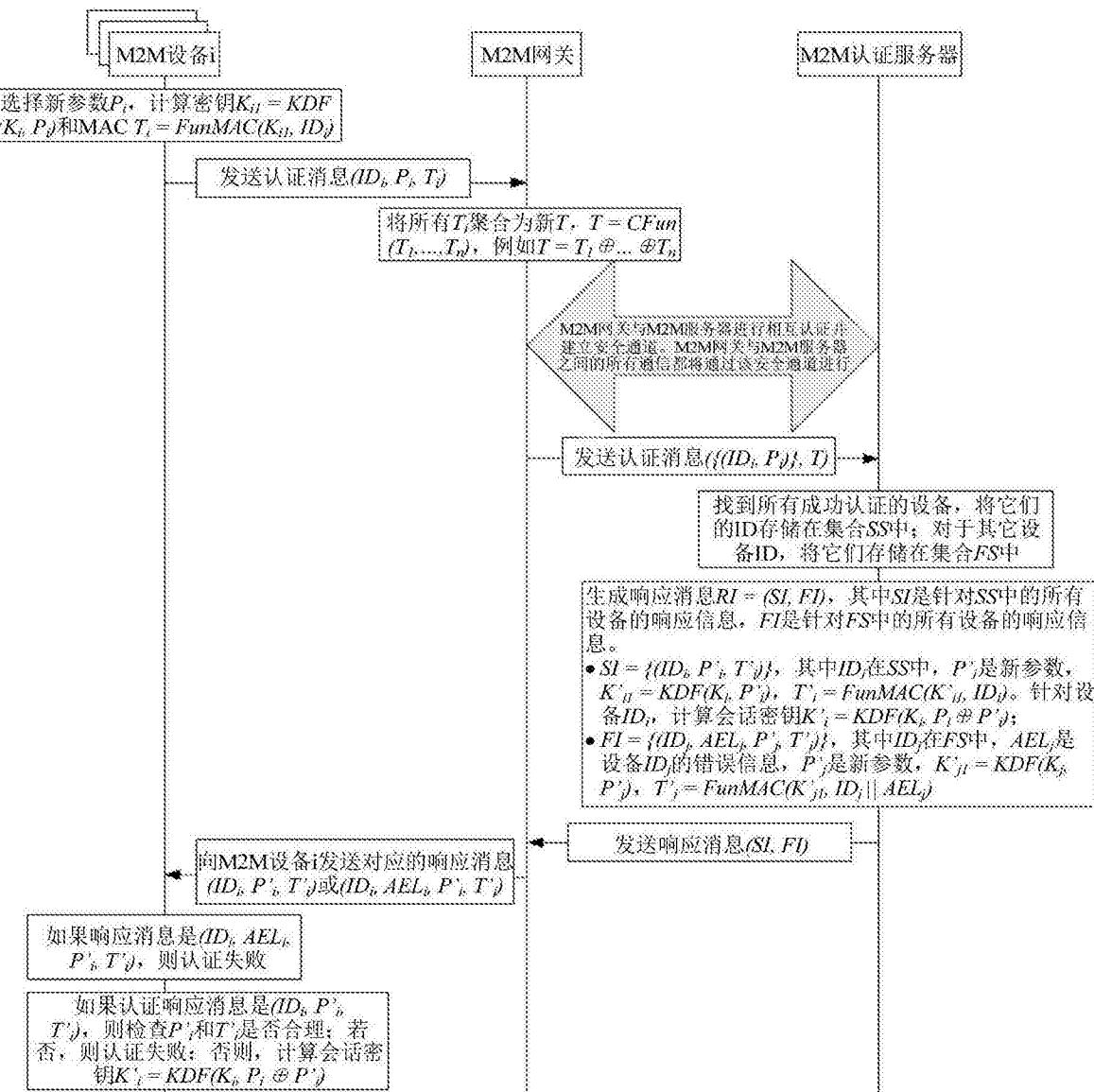


图5

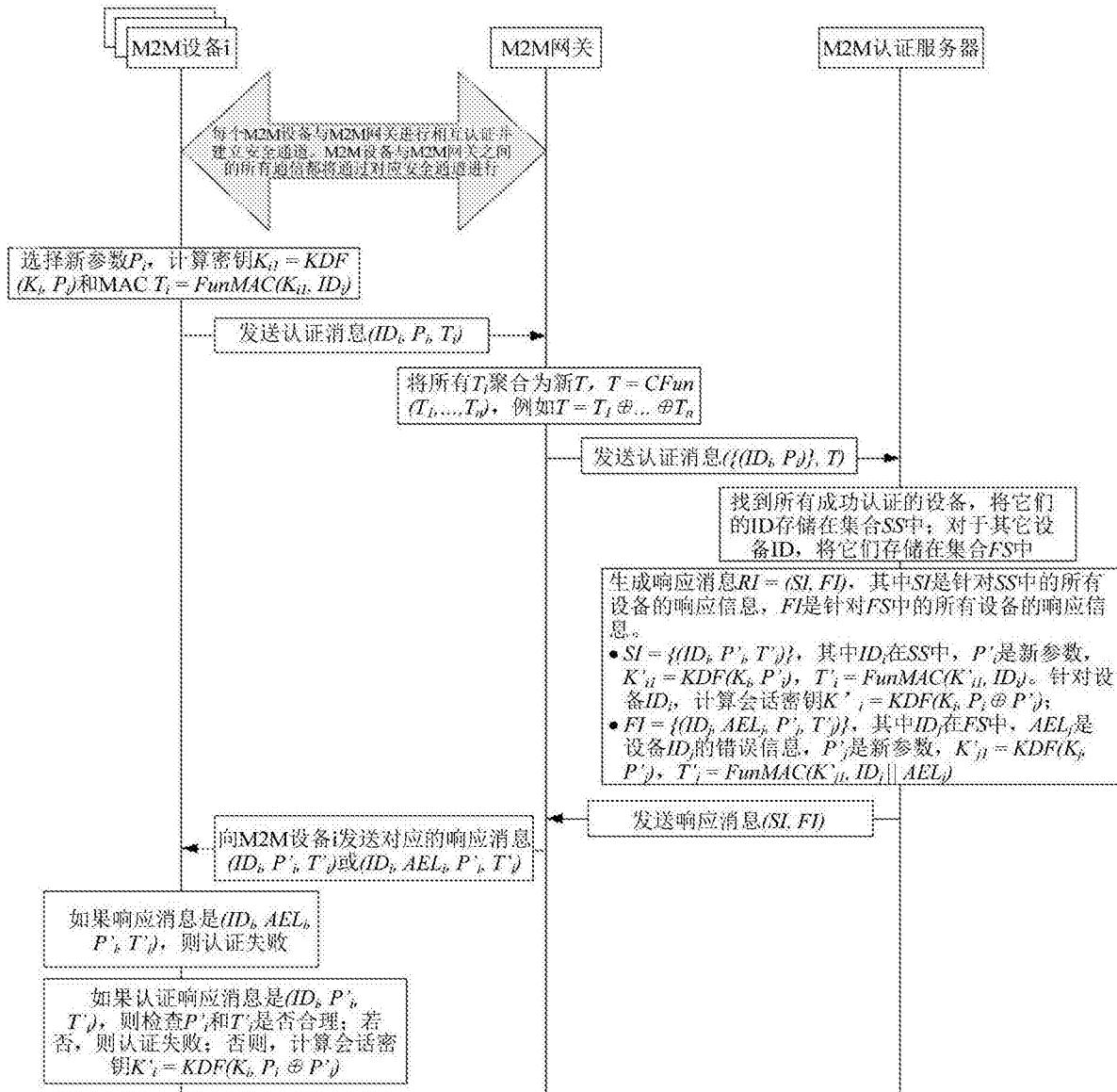


图6

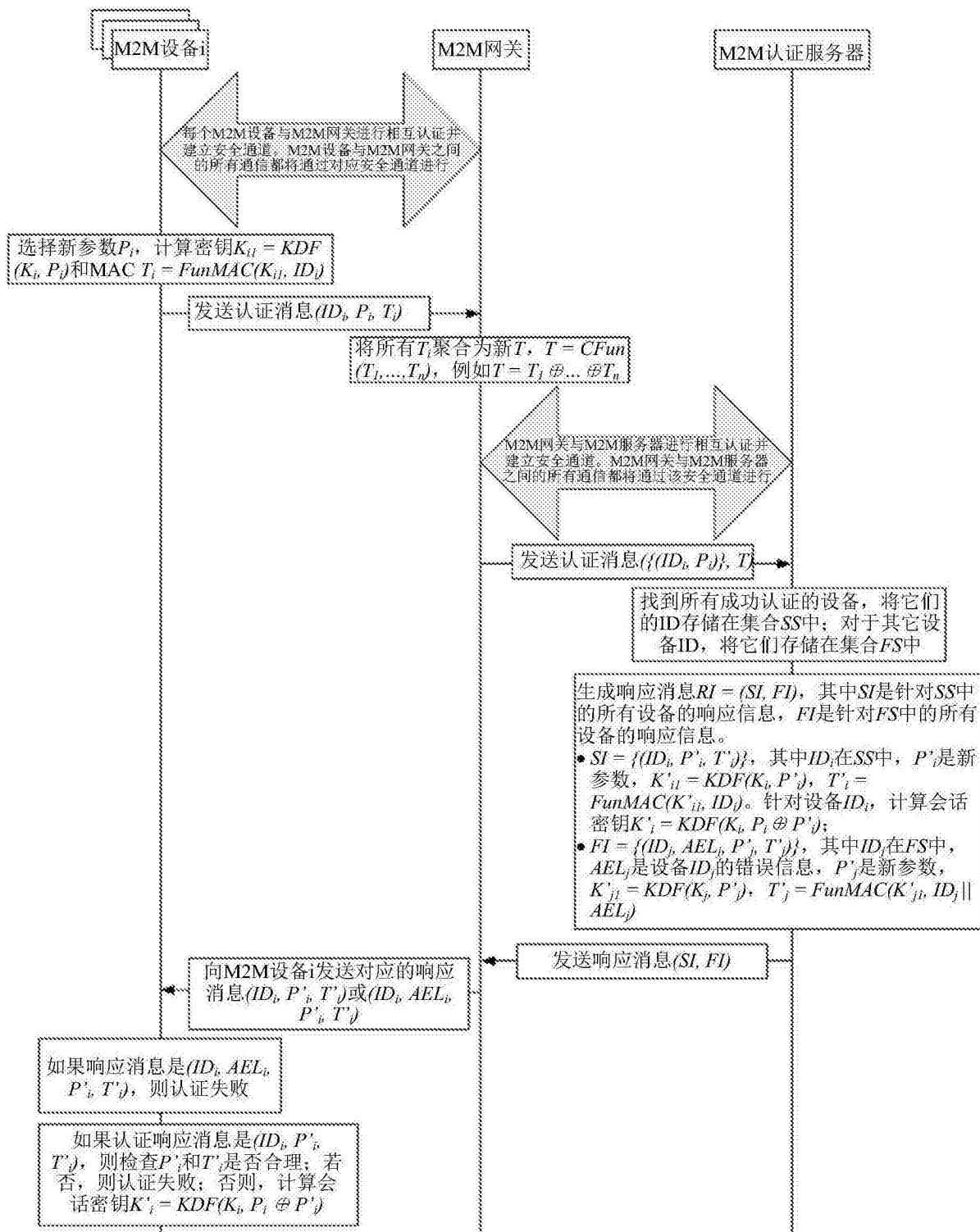


图7

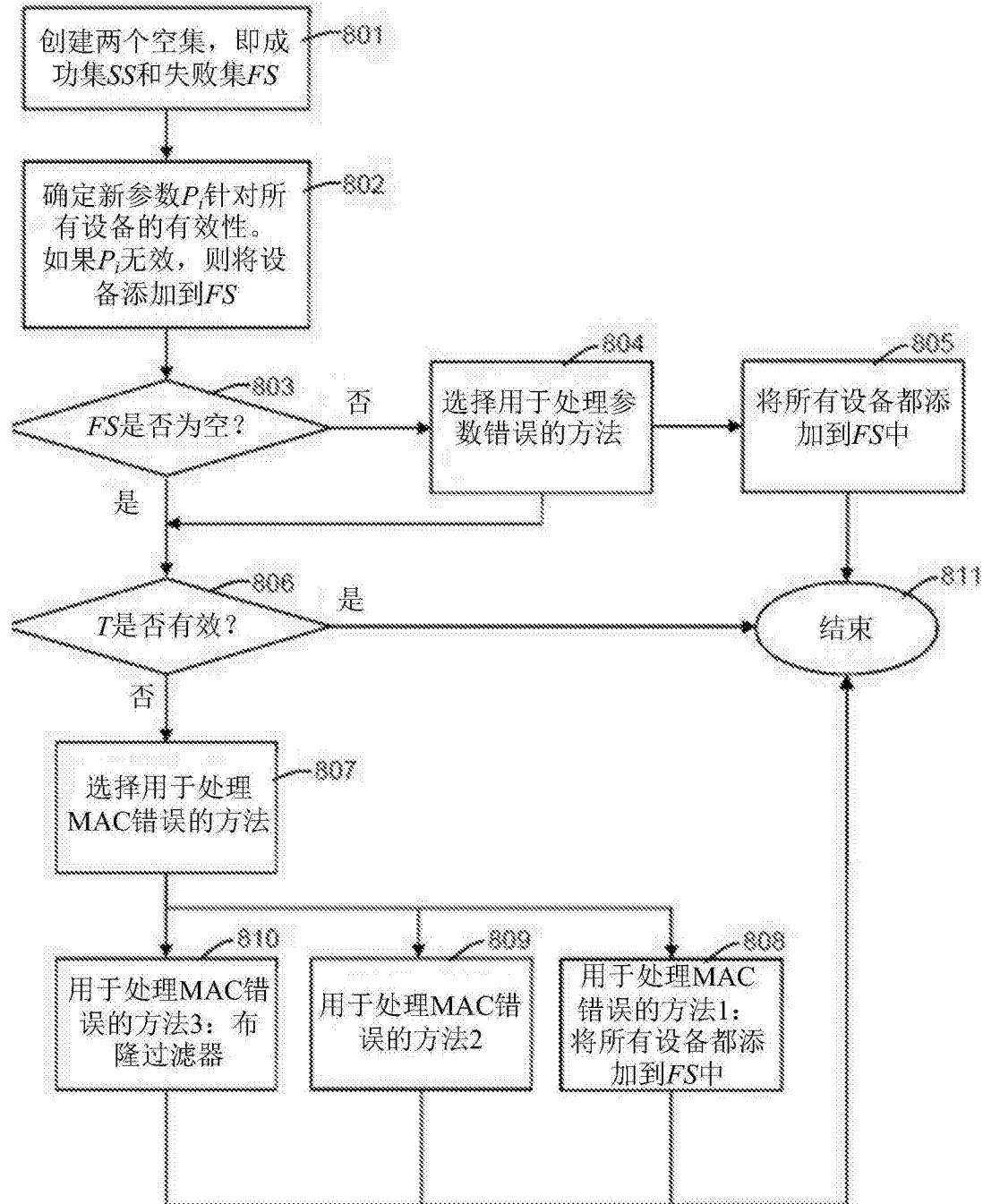


图8