



(10) **DE 10 2022 208 754 A1** 2024.02.29

(12)

Offenlegungsschrift

(21) Aktenzeichen: **10 2022 208 754.2**

(22) Anmeldetag: **24.08.2022**

(43) Offenlegungstag: **29.02.2024**

(51) Int Cl.: **H04L 9/08** (2006.01)

(71) Anmelder:

**Robert Bosch Gesellschaft mit beschränkter
Haftung, 70469 Stuttgart, DE**

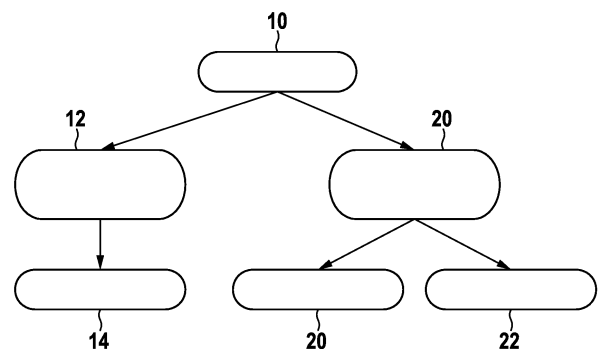
(72) Erfinder:

**Pai, Gautam, 70499 Stuttgart, DE; Pandit, Prajakta,
70499 Stuttgart, DE**

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **Authentifizierungsverfahren**

(57) Zusammenfassung: Verfahren zur Authentifizierung eines Clients gegenüber einem Server für einen Diagnosedienst, wobei der Client den Diagnosedienst anfragt durch Initiieren einer Verbindung mit einem Server durch Senden eines Zertifikats, das den Client identifiziert, der Server die Anfrage des Clients validiert und auf die Anfrage antwortet durch Senden von Informationen zum Einrichten eines sicheren Kanals an den Client, der Client die Antwort des Servers validiert, und den sicheren Kanal zu dem Server einrichtet, wobei ein nicht authentifiziertes Schlüsselvereinbarungsprotokoll verwendet wird, wobei das Verfahren den Schritt der bilateralen Authentifizierung umfasst, um nicht authentifizierte kurzlebige Schlüssel zu beheben, wobei eine eingerichtete Sitzung authentifiziert wird.



Beschreibung

[0001] Die Erfindung stellt ein Verfahren zur Authentifizierung eines Clients für einen Diagnosedienst und eine Anordnung zur Durchführung des Verfahrens bereit.

Stand der Technik

[0002] UDS (Unified Diagnostic Services) ist ein diagnostisches Kommunikationsprotokoll, das in elektronischen ECUs in der Automobilelektronik verwendet wird. Mit Hilfe der UDS-Diagnosetools können alle in einem Fahrzeug vorgesehenen ECUs kontaktiert werden.

[0003] Der UDS-Standard ISO 14229-1:2020 definiert einen neuen Diagnosedienst, den sogenannten Authentifizierungs- (29₁₆) Dienst. Der UDS-Standard besteht aus diagnostischen Diensten, wie Herunter-/Hochladen von Routinen, Lesen bestimmter Speicherorte von einem Server usw., bei denen eine Authentifizierung erforderlich ist, um die Sicherheit zu gewährleisten. Daher wird der Authentifizierungsdienst (29₁₆) als ein Mittel für den Client eingeführt, um seine Identität nachzuweisen, bevor ihm der Zugriff auf Daten und/oder Diagnosedienste, die eingeschränkten Zugriff haben, gestattet wird.

Offenbarung der Erfindung

[0004] Gemäß der Erfindung wird ein Authentifizierungsverfahren nach Anspruch 1 eingeführt.

[0005] Weiterhin wird eine zur Durchführung des Verfahrens geeignete Anordnung nach Anspruch 12 vorgestellt.

[0006] Das vorgeschlagene Verfahren dient der Authentifizierung eines Clients gegenüber einem Server für einen Diagnosedienst, wobei der Client den Diagnosedienst anfragt durch Initiieren einer Verbindung mit einem Server durch Senden eines den Client identifizierenden Zertifikats, der Server die Anfrage des Clients validiert und auf die Anforderung antwortet durch Senden von Informationen zum Einrichten eines sicheren Kanals an den Client, der Client die Antwort des Servers validiert und den sicheren Kanal zu dem Server einrichtet. Wobei ein nicht authentifiziertes Schlüsselvereinbarungsprotokoll, z. B. ein Diffie-Hellmann-Schlüsselvereinbarungsprotokoll, verwendet wird, das Verfahren den Schritt der bilateralen Authentifizierung umfasst, um nicht authentifizierte kurzlebige Schlüssel (engl: ephemeral keys) zu beheben, wodurch die eingerichtete Sitzung authentifiziert wird.

[0007] Die Sitzung kann unter Verwendung des Schritts der bilateralen Authentifizierung, der durch eine Option in dem Authentifizierungsdienst (29₁₆) bereitgestellt wird, vor der Einrichtung von Sitzungsschlüsseln eingerichtet werden, indem ein zusätzliches Schlüsseltransportprotokoll durch den Client verwendet wird, wobei die authentifizierten Schlüssel als die Sitzungsschlüssel eingerichtet werden, oder beispielsweise durch Verwendung eines Schlüsselvereinbarungsprotokolls, das es dem Client und dem Server ermöglicht, statische lange Schlüssel oder gemeinsame geheime Schlüssel zu verwenden, um Sitzungsschlüssel über einen unsicheren Kanal, Elliptic-Curve-Diffie-Hellmann- (ECDH) oder Kennwortauthentifizierte Schlüsselvereinbarung einzurichten.

[0008] Dementsprechend behebt das Verfahren eine Sicherheitsschwäche, nämlich nicht authentifizierte kurzlebige Schlüssel. Diese Sicherheitsschwäche wird hier, insbesondere im Zusammenhang mit den Zeichnungen, als Sicherheitsschwäche #2 bezeichnet.

[0009] In einer Ausführungsform wird das Verfahren zu Abschwächung eines Server-Spoof-Angriffs gegen den Client verwendet.

[0010] In einer anderen Ausführungsform wird beim Initiieren der Verbindung mit dem Server durch Senden eines Zertifikats, das den Client identifiziert, eine Präambel zu einem Handshake verwendet, um ein zuvor abgefangenes Client-Zertifikat zu beheben.

[0011] In dieser Ausführungsform behebt das Verfahren eine Sicherheitsschwäche, nämlich die Wiedereinspielung des Zertifikat-Clients. Diese Sicherheitsschwäche wird hier, insbesondere im Zusammenhang mit den Zeichnungen, als Sicherheitsschwäche #1 bezeichnet.

[0012] Die hier beschriebene Anordnung ist zur Durchführung des vorgeschlagenen Verfahrens geeignet. Die Anordnung kann in Hardware und/oder Software implementiert werden. Darüber hinaus kann die Anordnung in einer elektronischen Steuereinheit (ECU) integriert sein oder, in einer Ausführungsform, eine ECU sein.

Kurze Beschreibung der Zeichnungen

Fig. 1 zeigt einen Überblick über den Authentifizierungsdienst.

Fig. 2 zeigt ein Flussdiagramm, das ein Verfahren darstellt, wie es durch den UDS-Standard für den Authentifizierungs- (29₁₆) Dienst (Dienst 29) eingeführt wurde.

Fig. 3 zeigt das Flussdiagramm von **Fig. 1**, das eine erste Sicherheitsschwäche #1 in dem Dienst 29 veranschaulicht.

Fig. 4 zeigt das Flussdiagramm von **Fig. 1**, das eine zweite Sicherheitsschwäche #2 in dem Dienst 29 veranschaulicht.

Fig. 5: zeigt ein überarbeitetes Flussdiagramm basierend auf dem Flussdiagramm von **Fig. 1**, das die Sicherungsabschwächung für den DoS darstellt.

[0013] Es versteht sich, dass die vorstehend genannten und die nachstehend noch zu erläuternden Merkmale nicht nur in der jeweils angegebenen Kombination, sondern auch in anderen Kombinationen oder in Alleinstellung verwendbar sind, ohne den Rahmen der Erfindung zu verlassen.

[0014] Die Erfindung ist anhand von Ausführungsbeispielen in den Zeichnungen schematisch dargestellt und wird nachfolgend unter Bezugnahme auf die Zeichnungen näher erläutert. Es versteht sich, dass die Beschreibung in keiner Weise den Umfang der vorliegenden Erfindung einschränkt und eine annähernde Veranschaulichung von Ausführungsformen der Erfindung ist.

Beschreibung von Ausführungsformen

[0015] **Fig. 1** zeigt einen Überblick über den Authentifizierungsdienst. Der Authentifizierungs- (29₁₆) Dienst 10 unterstützt zwei Sicherheitskonzepte:

Ein erstes Konzept #112 basiert auf PKI- (Öffentlicher-Schlüssel-Infrastruktur) Zertifikatsaustausch- (APCE) Prozeduren unter Verwendung von asymmetrischer Kryptografie 14.

Ein zweites Konzept #20 basiert auf einem Aufforderungs-Antwort-Verfahren (ACR) ohne PKI-Zertifikate, das entweder asymmetrische Kryptografie 22 oder symmetrische Kryptografie 24 verwendet.

[0016] Für das hier beschriebene Verfahren wird das erste Sicherheitskonzept #112 betrachtet. Um dieses Konzept zu realisieren, ist in **Fig. 2** ein detailliertes Flussdiagramm dargestellt, wie es vom UDS-Standard für den Authentifizierungs- (29₁₆) Dienst eingeführt wurde.

[0017] **Fig. 2** zeigt auf der linken Seite die von einem Client 100 ausgeführten Schritte und auf der rechten Seite die von einem Server 102 ausgeführten Schritte. Zwischen diesen beiden Parteien 100 und 102 zeigt **Fig. 2** Kommunikationen, die zwischen dem Client 100 und dem Server 102 ausgetauscht werden.

[0018] Der UDS-Standard in dem Authentifizierungs- (29₁₆) Dienst bietet mehrere Optionen innerhalb des ersten Konzepts #1. Eine solche Option ist die Auswahl zwischen unidirektionaler und bidirektionaler Authentifizierung. Bei der unidirektionalen Authentifizierung muss sich nur der Client 100 gegenüber dem Server 102 authentifizieren. Bei der bidirektionalen Authentifizierung müssen sowohl der Client 100 als auch der Server 102 sich gegenseitig authentifizieren. Der Mechanismus zum Verifizieren der Authentizität der kommunizierenden Entität in Konzept #1 verwendet typischerweise PKI-Zertifikate. Der Dienst unterstützt auch verschiedene Mechanismen zum Einrichten eines sicheren Kanals, um die Vertraulichkeit und Integrität der Nachrichten zu gewährleisten.

[0019] Der Authentifizierungsprozess wird mit Schritt 110 gestartet. Der Client 100 initiiert eine Verbindung mit dem Server 102 und stellt notwendige Sicherheitsinformationen bereit, wie etwa die Zertifikate, Optionen zum Konfigurieren des sicheren Kanals, falls erforderlich, und so weiter (Schritte 112 und 114).

Schritt 112 Erzeuge Aufforderung Client

Schritt 114 Sende Zertifikat Client, Aufforderung Client (Authentifizierungsdienst mit Unterfunktion verifiziereZertifikatUnidirektional oder verifiziereZertifikatBidirektional)

[0020] Er kann auch die Serverauthentifizierung anfragen. Der Server 102 validiert diese Client-Anfrage (Schritt 116) und fährt mit den notwendigen Schritten fort (Schritte 118, 120, 122, 124, 126, 128, 130).

Schritt 118 Erzeuge Aufforderung Server
 Schritt 120 Erzeuge kurzlebige privates/öffentliches-Schlüsselpaar Server
 Schritt 122 Berechnen einen Eigentumsnachweis Server = Signiere (kurzlebiger öffentlicher Schlüssel Server, Aufforderung Server, Aufforderung Client)
 Schritt 124 Sende Aufforderung Server, Zertifikat Server, Eigentumsnachweis Server
 Schritt 126 Verifiziere den Eigentumsnachweis Client
 Schritt 128 Erzeuge oder Leite den/die Sitzungsschlüssel ab, und Errichte Zugriffsrechte für Sitzungsschlüsselinformationen
 Schritt 130 Sende Antwort und Sitzungsschlüsselinformationen

[0021] Der Server 102 kann Sicherheitsmaterial für die Einrichtung eines sicheren Kanals mit dem Client 100 bereitstellen. Der Client 100 validiert die Antwort (Schritt 132) und fährt mit den Schritten fort, um denselben sicheren Kanal mit dem Server 102 einzurichten (Schritte 134, 136, 138, 140, 142, 144).

Schritt 134 Erzeuge kurzlebige privates/öffentliches Schlüsselpaar Client
 Schritt 136 Berechne den Eigentumsnachweis Server = Signiere (kurzlebiger öffentlicher Schlüssel Server, Aufforderung Client, Aufforderung Server)
 Schritt 138 Send Eigentumsnachweis Client, kurzlebigen öffentlichen Schlüssel Client (Authentifizierungsdienst mit Unterfunktion Eigentumsnachweis)
 Schritt 140 Extrahiere oder Leite den/die Sitzungsschlüssel ab
 Schritt 142 Verifiziere Sitzungsschlüsselinformationen
 Schritt 144 Aktiviere Sitzungsschlüssel

[0022] Das Flussdiagramm umfasst ferner Kommunikationen, die zwischen dem Client 100 und dem Server 102 ausgetauscht werden.

Kommunikation 170 Anforderungsnachricht
 < Kommunikationskonfiguration>
 <Zertifikat Client>
 <Aufforderung Client>
 Kommunikation 172 Antwortnachricht
 < Rückgabewert>
 Kommunikation 174 Antwortnachricht
 < Rückgabewert>
 < Herausforderung Server>
 <Zertifikat Server>
 < Eigentumsnachweis Server>
 <kurzlebiger öffentlicher Schlüssel Server>
 Kommunikation 176 Anforderungsnachricht
 < Eigentumsnachweis Client>
 <kurzlebiger öffentlicher Schlüssel Client>
 Kommunikation 178 Antwortnachricht

< Rückgabewert>
 Kommunikation 180 Antwortnachricht
 < Rückgabewert>
 < Sitzungsschlüsselinformationen>

[0023] Ferner zeigt das Flussdiagramm Schritte, die von dem Client 100 und dem Server 102 durchgeführt werden, die innerhalb des beschriebenen Ablaufs durchgeführt werden, wobei

Schritt 133 Zertifikats Server Eigentumsnachweis ok?

Wenn nicht, ist die Authentifizierung fehlgeschlagen 150. Wenn ja, fährt das Verfahren mit Schritt 134 fort.

Schritt 143 Sitzungsschlüsselinformationen ok?

Wenn nicht, ist die Authentifizierung fehlgeschlagen 150. Wenn ja, fährt das Verfahren mit Schritt 144 fort, und dann war die Authentifizierung erfolgreich 152.

Schritt 117 Zertifikat Client gültig?

Wenn nicht, wird die Mitteilung 172 gesendet. Wenn ja, fährt das Verfahren mit Schritt 118 fort.

Schritt 127 Eigentumsnachweis Client ok?

Wenn nicht, wird die Mitteilung 178 gesendet. Wenn ja, fährt das Verfahren mit Schritt 128 fort.

[0024] Bei der Untersuchung des in **Fig. 2** gezeigten Verfahrens wurden Sicherheitsschwächen in diesem Authentifizierungsdienst identifiziert, und die Sicherheitsangriffe, die als Auswirkung dieser Sicherheitsschwächen auftreten können, werden aufgelistet. Die Sicherheitslücken und Angriffe sind in den **Fig. 3** und **4** dargestellt.

[0025] **Fig. 3** zeigt die Sicherheitsschwäche #1 in dem Dienst 29.

[0026] Es gibt zwei Sicherheitslücken, die unten aufgeführt sind:

i. Sicherheitslücke #1: Wiedereinspielung des „Zertifikat Client“ (Bezugszeichen 200)

[0027] In dieser Figur beinhaltet der Schritt 116 nur die Verifizierung der Vertrauenskette. Der Client (Tester) 100 wird jedoch nicht als Teil dieses Schrittes authentifiziert. Ein Angreifer kann diese Schwachstelle ausnutzen, und jedes zuvor abgefangene Client-Zertifikat wiedereinzuspielen. Da der Server (ECU) 102 derzeit keine anderen Informationen hat, um die Identität des Testers in Schritt 116 zu verifizieren, würde dies dem Angreifer einen unbefugten Zugriff auf die ECU-Ressourcen verschaffen, um die folgenden Angriffe durchzuführen:

a. Sicherheitsangriff #1: Denial-of-Service (DoS) auf den Server 102 für „Aufforderung Server“ (Bezugszeichen 202)

Nachdem der Angreifer eine nicht autorisierte authentifizierte Sitzung mit der ECU erhalten hat, kann er bei der ECU anfragen, eine „Aufforderung“ zu berechnen. Dies ist in Schritt 118 von **Fig. 3** angegeben. Wenn die Anforderungen an die Berechnung der Aufforderung streng sind, um Sicherheit zu gewährleisten, wenn beispielsweise die Aufforderung kryptografisch berechnet werden muss, die Aufforderung ein Mindestlängenkriterium erfüllen muss, die Aufforderung zufällig sein muss, und die Erzeugung nicht vorhersehbar sein sollte, und so weiter, dann führt dies zur Erschöpfung der ECU-Ressourcen. Somit kann ein Angreifer mit abgefangenen Client-Zertifikatsinformationen DoS-Angriffe auf die ECU durchführen. Da das Client-Zertifikat nicht vertraulich behandelt wird, kann ein Angreifer diese Informationen leicht erhalten, was weiterhin den Replay-Angriff ermöglicht.

b. Sicherheitsangriff #2: DoS auf Server für „Eigentumsnachweis Server“ (Bezugszeichen 204)

Wie in dem obigen Sicherheitsangriff erläutert, kann ein Angreifer, sobald er erfolgreich eine nicht autorisierte authentifizierte Sitzung mit der ECU erhalten hat, DoS-Angriffe problemlos durchführen. Eine weitere Angriffsfläche, die durch das Protokolldesign bereitgestellt wird, befindet sich in Schritt 122. Dieser Schritt 122 beinhaltet eine asymmetrische kryptografische Berechnung. Die Rechenzyklen und der Rechenaufwand sind für eine asymmetrische Kryptografie (digitale Signaturen) aufgrund der Schlüssellänge des verwendeten Algorithmus sehr hoch. Dies wird auch vor jeder Tester-Authentifizierung durch-

geführt. Somit bietet dies einem Angreifer eine weitere Gelegenheit, die ECU-Ressourcen zu missbrauchen und einen DoS durchzuführen.

[0028] Ein Denial-of-Service- (DoS) Angriff ist ein Cyberangriff, bei dem der Angreifer versucht, eine Maschine oder eine Netzwerkressource für die vorgesehenen Benutzer unzugänglich zu machen, indem er die Dienste eines über ein Netzwerk verbundenen Hosts vorübergehend oder auf unbestimmte Zeit unterbricht. Ein DoS wird typischerweise erreicht, indem die Maschine oder das Netzwerk mit überflüssigen Anfragen überflutet wird, um die Maschine oder das Netzwerk zu überlasten und zu verhindern, dass einige oder alle legitimen Anfragen erfüllt werden.

[0029] Fig. 4 zeigt die Sicherheitsschwäche #2 in Dienst 29.

ii. Sicherheitsschwäche #2: nicht authentifizierte kurzlebige Schlüssel (Bezugszeichen 300)

[0030] Der UDS-Standard bietet die Möglichkeit, zwischen unidirektionaler und bidirektionaler Authentifizierung zu wählen. Der Hauptunterschied besteht darin, dass sich bei der bidirektionalen Authentifizierung sowohl der Client 100 als auch der Server 102 gegenseitig authentifizieren müssen, wohingegen sich bei der unidirektionalen Authentifizierung nur der Client 100 gegenüber dem Server 102 authentifizieren muss. Wenn dann eine sichere Diagnosekommunikation gewünscht wird, bietet der UDS-Standard auch eine Option zur Verwendung eines Schlüsselvereinbarungsprotokolls zum Ableiten der Sitzungsschlüssel.

[0031] Diese Schwachstelle wird in den UDS-Dienst eingeführt, wenn eine unidirektionale Authentifizierung zusammen mit kurzlebiger Diffie-Hellman als Schlüsselvereinbarungsprotokoll gewählt wird. Das Diffie-Hellman-Schlüsselvereinbarungsprotokoll ist ein Verfahren zum sicheren Austausch kryptografischer Schlüssel über einen öffentlichen Kanal. Die Diffie-Hellman-Schlüsselvereinbarung selbst ist ein nicht authentifiziertes Schlüsselvereinbarungsprotokoll. Es ermöglicht einem Angreifer, den folgenden Angriff durchzuführen:

a. Sicherheitsangriff #3: Server-Spoof-Angriff auf den Client

Die Diffie-Hellman-Schlüsselvereinbarung stützt sich auf die Kryptografie mit öffentlichen Schlüsseln, um die Authentizität sicherzustellen. In dem Fall einer unidirektionalen Authentifizierung wird nur der Client (Tester) 100 gegenüber dem Server (ECU) 102 authentifiziert, und daher bleiben die von der ECU (in Schritt 120) erzeugten Schlüsselinformationen unauthentifiziert. Dadurch kann sich ein Angreifer als ECU ausgeben und böswillige Antworten an den Tester zurücksenden. Dieser Angriff wird in den Szenarien kritischer, in denen der Tester nicht in der Lage ist, die ECU, mit der er verbunden ist, physisch zu authentifizieren oder zu verifizieren, beispielsweise in Funktionen, wie Firmware-über-die-Luft- (FOTA) Aktualisierungen, Ferndiagnose usw.

[0032] Bei dem Verfahren nach Fig. 4 sind die Schritte 112, 122, 132 und 133 optionale Schritte.

[0033] Es muss berücksichtigt werden, dass das Automobilunternehmen in das Internet der Dinge (IoT) integriert wird. Damit ist ein Automobil kein isoliertes einfaches System mehr, sondern wird Teil der vernetzten Welt. Dies hat zur Einführung mehrerer Funktionen geführt, die den Zugriff auf verschiedene drahtlose Technologien erfordern. Einige dieser Funktionen, die in der Automobilindustrie weit verbreitet sind, sind Software-Aktualisierungen über die Luft, Ferndiagnose, vernetzte Dienste, usw. Daher wird es für die Automobilindustrie entscheidend, die Sicherheitskonzepte und Schutzmechanismen zu aktualisieren.

[0034] Die ISO- (Internationale Organisation für Standardisierung) Gemeinschaft hat aus ähnlichen Gründen Sicherheitserweiterungen für den bestehenden UDS-Standard eingeführt. Der Hauptzweck bestand darin, die Einschränkungen der in dem bestehenden Standard bereitgestellten Sicherheitsmechanismen zu überwinden und die Sicherheit zu erhöhen. Mit der Zunahme von Anwendungen, wie Ferndiagnose und OTA (über die Luft), wächst die Angriffsfläche und damit das Sicherheitsrisiko. Daher muss sichergestellt werden, dass der als Sicherheitsmechanismus eingeführte Dienst keine weiteren Sicherheitsrisiken einführt. Das vorgeschlagene Verfahren zielt darauf ab, einige dieser Sicherheitsschwächen zu identifizieren und zu mindern.

[0035] Das vorgeschlagene Verfahren zielt darauf ab, die nachstehenden Sicherheitsschwächen zu beheben, die aufgrund von Designfehlern im Authentifizierungsdienst im neuesten UDS-Standard eingeführt wurden.

i. Sicherheitsschwäche #1: Wiedereinspielung von „Zertifikat Client“ (Bezugszeichen 200 in Fig. 3)

[0036] Um diese Schwachstelle zu beseitigen, wird im Folgenden eine Präambel zum bestehenden Handshake vorgeschlagen. Die folgenden Angriffe werden daher abgeschwächt.

- a. Sicherheitsangriff #1: DoS auf Server für „Aufforderung-Server“ (Bezugszeichen 202)
- b. Sicherheitsangriff #2: DoS auf Server für „Eigentumsnachweis“ (Bezugszeichen 204)

[0037] Auch dieser Vorschlag zielt darauf ab, die Änderungen des Core-Handshakes gegenüber dem UDS-Standard möglichst gering zu halten.

- ii. Sicherheitsschwäche #2: Nicht authentifizierte kurzlebige Schlüssel (Bezugszeichen 300 in Figur 4)

[0038] Um diese Schwachstelle zu beseitigen, wird im Folgenden eine ausführliche Einschränkung des Sicherheitsdesigns vorgeschlagen. Mit dieser zusätzlichen Einschränkung werden die folgenden Angriffe abgeschwächt.

- a. Sicherheitsangriff #3: Server-Spoof-Angriff auf den Client

[0039] Der Aufbau und die Funktion des vorgeschlagenen Verfahrens werden nachstehend zusammen mit alternativen Ausführungsformen beschrieben.

[0040] Die Minderung der oben genannten Sicherheitsschwächen wird im Folgenden beschrieben:

- i. Sicherungsabschwächung für die Sicherheitslücke #1 (Bezugszeichen 200)

[0041] Die Hauptursache für die Einführung dieser Schwachstelle ist die Tatsache, dass das Protokolldesign einen hohen Rechenaufwand seitens des Servers (ECU) vor der Authentifizierung des Clients (Tester) berücksichtigte. Somit ist die ECU nicht in der Lage, zwischen legitimen (authentifizierten) und unrechtmäßigen Anfragen zu unterscheiden.

[0042] Dieser Vorschlag führt einen kleinen Handshake zwischen dem Client (Tester) und dem Server (ECU) ein, bevor die ECU-Ressourcen für umfangreiche kryptografische Berechnungen verbraucht werden. Dieser Handshake würde der ECU helfen, zwischen einem legitimen Tester und einem Angreifer zu unterscheiden. Der Vorschlag zielt darauf ab, eine einzige Lösung für beide oben beschriebenen Sicherheitsangriffe einzuführen, ohne dass drastische Anpassungen des Standards erforderlich sind. Dieser Vorschlag kann auch als Vorbedingungsprüfung vor dem Schritt 112 in **Fig. 3** betrachtet werden. Dieser Handshake-Vorschlag beinhaltet den Austausch von Informationen, die die Wiedereinspielung der sensiblen Informationen (Zertifikat Client) in Schritt 114 an die ECU verhindern würden.

[0043] Der Vorschlag wird in **Fig. 5** ausführlich beschrieben.

[0044] Weitere Schritte und Kommunikationen, die in **Fig. 5** gezeigt werden, im Vergleich zu den **Fig. 2** bis 5, sind:

Schritt 400	Hallo-Anfrage
Schritt 404	Cookie = Hash (ECU-Geheimnis, ZeitstempelServer)
Schritt 406	Sende ZeitstempelServer, Cookie
Schritt 408	Cookie gültig?

[0045] Falls nicht, wird die Mitteilung 424 gesendet. Falls ja, wird das Verfahren mit Schritt 116 fortgesetzt.

Schritt 410	Weiterverarbeitung arbeitung gemäß dem Authentifizierungsdienstfluss im UDS-Standard
Kommunikation 420	Antwortnachricht < ZeitstempelServer> <Cookie>
Kommunikation 422	Anforderungsnachricht < Kommunikationskonfiguration >

	< Zertifikat-Client>
	< Herausforderung Client>
	< ZeitstempelServer>
	<Cookie>
Kommunikation 424	Antwortnachricht
	< Rückgabewert>
Kommunikation 426	Antwortnachricht
	< Rückgabewert>
Kommunikation 428	Antwortnachricht
	< Rückgabewert>
	< Herausforderung Server>
	<Zertifikat Server>
	< Eigentumsnachweis Server>
	<kurzlebiger öffentlicher Schlüsselservers>

[0046] In einer Ausführungsform umfasst das Verfahren den Schritt der Berechnung des „Cookie“ auf Seiten der ECU, bevor die ECU sich selbst in die umfangreichen kryptografischen Berechnungen einbezieht. Ein Hash-Algorithmus ist ein möglicher Algorithmus, da dieser im Vergleich zu anderen kryptografischen Algorithmen keine hohen ECU-Ressourcen verbrauchen würde. Das „Cookie“ wird hauptsächlich als Hash über zwei Eingaben berechnet:

- a. ECU-Geheimnis: Dieser Wert kann anhand der ECU-Hardwareeigenschaften abgeleitet werden und ist nur der ECU bekannt. Dieses Geheimnis sollte weder ausgelesen noch geteilt werden. Ohne die Information des „ECU-Geheimnisses“ wäre es rechnerisch nicht machbar, das Cookie zu erzeugen. Da das „ECU-Geheimnis“ unter Verwendung der ECU-Eigenschaften abgeleitet wird, wird das erzeugte Cookie außerdem ECU-spezifisch, und die Wiedereinspielung der Cookie-Informationen von einer kompromittierten ECU an andere ECUs wird unpraktisch.
- b. Zeitstempel: Diese Informationen sind enthalten, um Angreifer daran zu hindern, zuvor abgefangene Cookies an eine Ziel-ECU zu senden. Für die Gültigkeit des Zeitstempels kann ein Zeitfenster definiert werden.

[0047] Die primäre Absicht bei der Auswahl dieser Eingabewerte für die „Cookie“-Berechnung besteht darin, die ECU zustandslos zu halten. Wenn also ein Angreifer versucht, einen DoS durchzuführen, wäre die ECU nicht an der kryptografischen Berechnung beteiligt. Gemäß dem UDS-Standard würde die ECU mit der Verifizierung des Client-Zertifikats (Schritt 116) nur dann beginnen, wenn ein gültiges Cookie vom Tester in Schritt 114 von **Fig. 5** bereitgestellt wird.

[0048] Gemäß dem Vorschlag in **Fig. 5** zeigt ein Tester der ECU an, dass er den Authentifizierungsdienst initiieren möchte (Schritt 400). Die ECU soll dem Tester mit einem „Cookie“ antworten (Schritt 400). Es wird dann erwartet, dass der Tester dieses „Cookie“ in Schritt 114 zusammen mit den Zertifikatsinformationen anhängt. Die Einbeziehung des Cookies in diesen Schritt gibt der ECU die Gewissheit einer legitimen Anfrage. Somit befreit die Einführung des Cookie-Konzepts in dem Authentifizierungsdienst als eine Vorbedingung die ECU von der Durchführung der schwierigen kryptografischen Berechnungen in Schritt 118 und Schritt 122 . Somit werden beide Angriffe, Sicherheitsangriff #1 202 und Sicherheitsangriff #2 204, abgeschwächt.

[0049] Zusätzlich kann das vorgeschlagene Verfahren Sicherheit für die neu eingeführten Parameter im Handshake bieten. Diese sind unten aufgeführt:

- a. Verhinderung der Wiedereinspielung von Cookies: Ein Angreifer, der es erfolgreich geschafft hat, die in Schritt 114 von **Fig. 5** (Cookie + Client-Zertifikat) ausgetauschten Informationen abzufangen, kann nicht länger einen Replay-Angriff durchführen und eine nicht autorisierte authentifizierte Sitzung mit der ECU starten, da die ECU das Cookie aufgrund des Ablaufs des Zeitstempels ungültig machen würde.

b. Verhinderung der Manipulation des Zeitstempels: Ein Angreifer wird auch nicht in der Lage sein, den Zeitstempel zu manipulieren und das Cookie wiedereinzuspielen. Dies liegt daran, dass die Cookie-Berechnung durch die ECU aufgrund des falschen Zeitstempelwerts fehlschlagen würde.

ii. Sicherungsabschwächung für die Sicherheitslücke #2 (Bezugszeichen 300)

[0050] Wie oben beschrieben, hat der UDS-Standard einen neuen Authentifizierungsdienst eingeführt, um die Sicherheit der Diagnosekommunikation zu gewährleisten. Der Standard erleichtert es auch dem OEM (Hersteller des Originalprodukts) und/oder Lieferanten, diesen Service basierend auf seinen Sicherheitsanforderungen zu konfigurieren. Die Beispielkonfigurationen sind unilaterale/bilaterale Authentifizierung, PKI-basierte/Aufforderung-Antwort-basierte Authentifizierung, Aufbau eines sicheren Kanals, Sicherheitsalgorithmen, und so weiter.

[0051] Bei solchen Konfigurationsparametern ist es zwingend erforderlich, einen sicheren Designprozess zu implementieren, um sicherzustellen, dass keine zusätzlichen Schwachstellen oder Schlupflöcher eingeführt werden. Eine solche Konfiguration, die eine Sicherheitslücke einführt, ist die Verwendung des Diffie-Hellmann-Schlüsselvereinbarungsprotokolls mit unidirektionaler Authentifizierung. Wenn Diffie-Hellmann als Schlüsselvereinbarungsprotokoll geplant ist, wird empfohlen, die bilaterale Authentifizierung als Mindestanforderung zu verwenden, um die Sicherheit des Dienstes 29 zu gewährleisten.

[0052] Der UDS-Standard ist bei allen Produkten aus verschiedenen Fahrzeugbereichen weit verbreitet. Produkte, wie Motor-ECUs, zentrale Gateways, Lenkungs-ECUs sowie Testgeräte, basieren auf dem UDS-Standard. Das vorgeschlagene Verfahren kann in jedem der oben erwähnten Produkte durchgeführt werden.

Patentansprüche

1. Verfahren zur Authentifizierung eines Clients (100) gegenüber einem Server (102) für einen Diagnosedienst, wobei

- der Client (100) den Diagnosedienst anfragt, indem er eine Verbindung mit einem Server (102) initiiert, indem er ein Zertifikat sendet, das den Client (100) identifiziert,
- der Server (102) die Anfrage des Clients (100) validiert und auf die Anfrage antwortet, indem er Informationen zum Einrichten eines sicheren Kanals an den Client (100) sendet,
- der Client (100) die Antwort des Servers (102) validiert und den sicheren Kanal zu dem Server (102) einrichtet,
- wobei ein nicht authentifiziertes Schlüsselvereinbarungsprotokoll verwendet wird, wobei das Verfahren den Schritt einer bilateralen Authentifizierung umfasst, um nicht authentifizierte kurzlebige Schlüssel zu beheben, wobei eine eingerichtete Sitzung authentifiziert wird.

2. Verfahren nach Anspruch 1, wobei die Sitzung eingerichtet wird, indem der Schritt der bilateralen Authentifizierung verwendet wird, die durch eine Option in dem Authentifizierungs- (29₁₆) Dienst (10) vor dem Einrichten von Sitzungsschlüsseln bereitgestellt wird.

3. Verfahren nach Anspruch 1, wobei die Sitzung unter Verwendung eines zusätzlichen Schlüsseltransportprotokolls durch den Client (100) eingerichtet wird, wobei die authentifizierten Schlüssel als die Sitzungsschlüssel eingerichtet werden.

4. Verfahren nach Anspruch 1, wobei die Sitzung eingerichtet wird, indem ein Schlüsselvereinbarungsprotokoll verwendet wird, das es dem Client (100) und dem Server (102) ermöglicht, statische lange Schlüssel oder gemeinsame geheime Schlüssel zu verwenden, um Sitzungsschlüssel über einen unsicheren Kanal, Elliptic-Curve-Diffie-Hellmann- (ECDH) oder Kennwortauthentifizierte Schlüsselvereinbarung einzurichten.

5. Verfahren nach einem der Ansprüche 1 bis 4, wobei das Verfahren zum Abschwächen eines Server-Spoof-Angriffs gegen den Client (100) verwendet wird.

6. Verfahren nach einem der Ansprüche 1 bis 5, wobei beim Initiieren der Verbindung mit dem Server (102) durch Senden eines den Client (100) identifizierenden Zertifikats eine Präambel zu einem Handshake verwendet wird, um ein zuvor abgefangenes Client-Zertifikat zu beheben.

7. Verfahren nach Anspruch 6, wobei das Verfahren dazu dient, einen Denial-of-Service-Angriff auf den Server (102) zum Erzeugen einer Aufforderung abzuschwächen.
8. Verfahren nach Anspruch 6 oder 7, wobei das Verfahren dazu verwendet wird, einen Denial-of-Service-Angriff auf den Server (102) zum Eigentumsnachweis abzuschwächen.
9. Verfahren nach einem der Ansprüche 6 bis 8, wobei zum Bestimmen der Präambel ein Cookie berechnet wird.
10. Verfahren nach Anspruch 9, wobei das Cookie hauptsächlich als ein Hash über zwei Eingaben berechnet wird.
11. Verfahren nach Anspruch 10, wobei die zwei Eingaben zum Berechnen eines Hashs ein ECU-Geheimnis und ein Zeitstempel sind.
12. Anordnung zur Authentifizierung, die dazu geeignet ist, ein Verfahren nach einem der Ansprüche 1 bis 10 durchzuführen.

Es folgen 5 Seiten Zeichnungen

Anhängende Zeichnungen

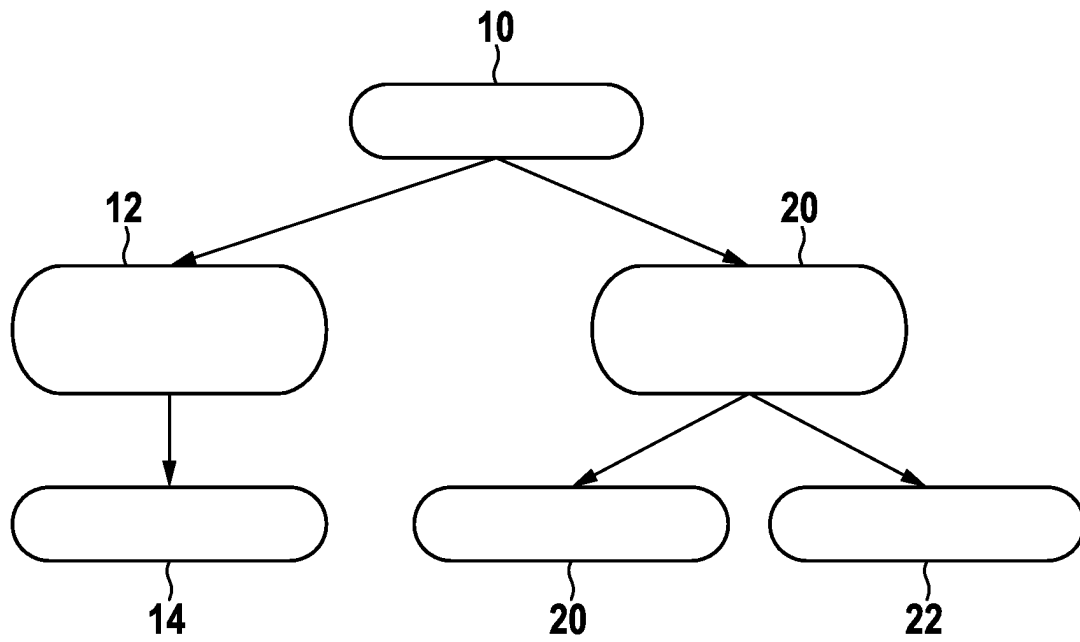


Fig. 1

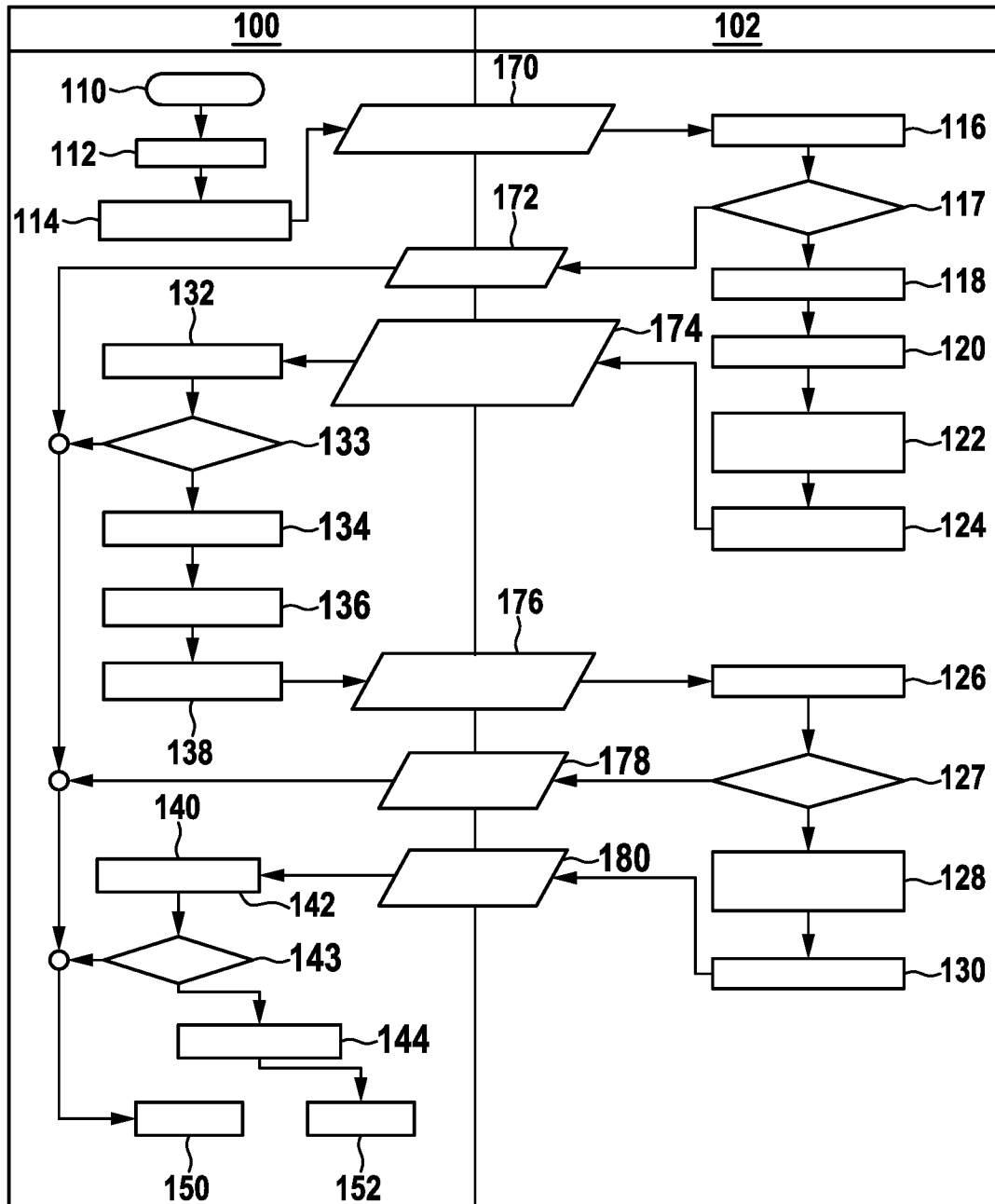


Fig. 2

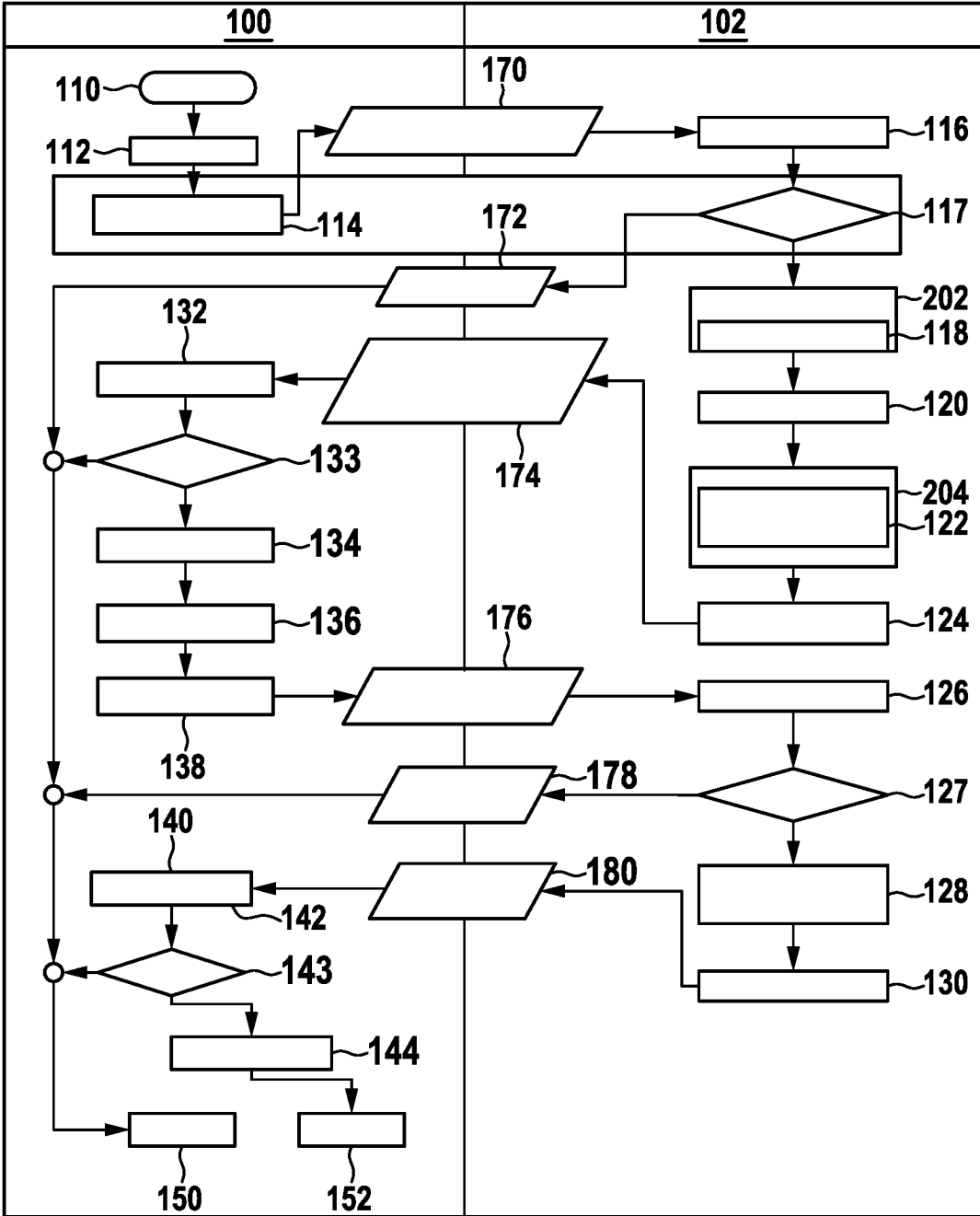
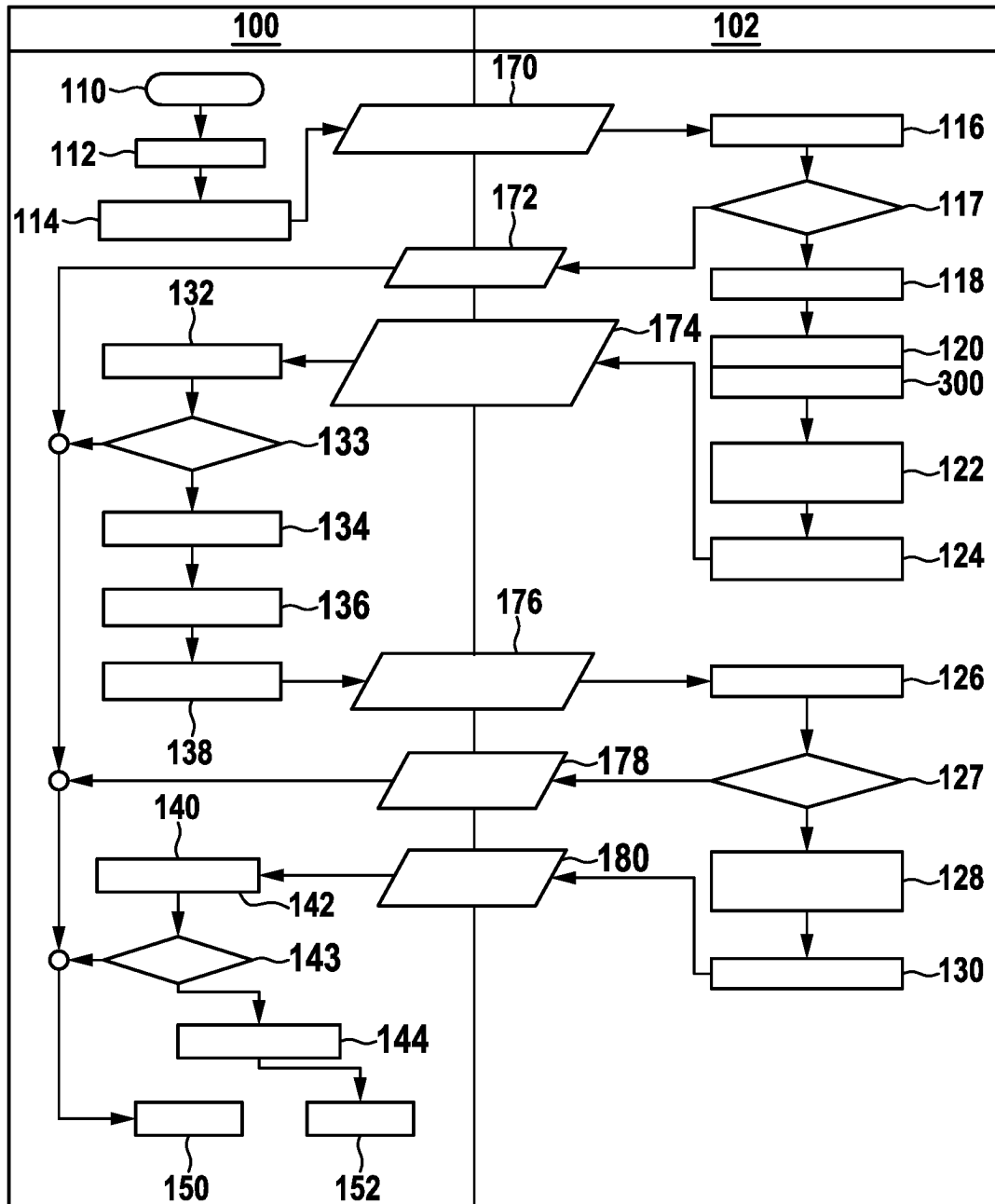


Fig. 3

**Fig. 4**

