(19) Deutsches Patent- und Markenamt

(10) **DE 11 2019 003 096 B4** 2023.08.17

(12) # Patentschrift

(21) Deutsches Aktenzeichen: **11 2019 003 096.5**
(86) PCT-Aktenzeichen: **PCT/US2019/036100**
(87) PCT-Veröffentlichungs-Nr.: **WO 2019/245760**
(86) PCT-Anmeldetag: **07.06.2019**
(87) PCT-Veröffentlichungstag: **26.12.2019**
(43) Veröffentlichungstag der PCT Anmeldung in deutscher Übersetzung: **02.06.2021**
(45) Veröffentlichungstag der Patenterteilung: **17.08.2023**

(51) Int Cl.: ***H04L 9/00*** (2022.01)
H04L 9/32 (2006.01)
H04L 9/08 (2006.01)
G06F 12/14 (2006.01)
G06F 21/60 (2013.01)
G06F 13/42 (2006.01)
G06F 21/78 (2013.01)
H04L 9/40 (2022.01)

(30) Unionspriorität:

62/687,146	**19.06.2018**	**US**
16/431,548	**04.06.2019**	**US**

(73) Patentinhaber:
Infineon Technologies LLC, San Jose, CA, US

(74) Vertreter:
Murgitroyd Germany Patentanwaltsgesellschaft mbH, 80636 München, DE

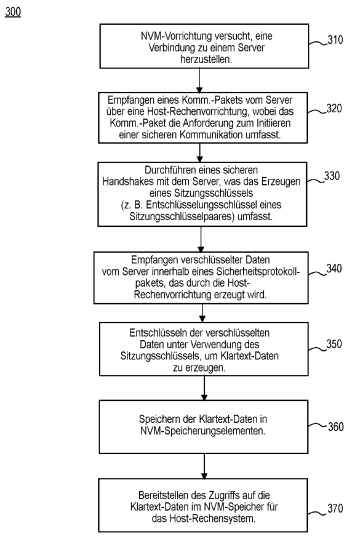
(72) Erfinder:
Ostrikov, Sergey, Redwood City, CA, US; Rosner, Stephan, Campbell, Calif., US; Zitlaw, Cliff, San Jose, CA, US

(56) Ermittelter Stand der Technik:

US	**2005 / 0 108 571**	**A1**
US	**2014 / 0 215 111**	**A1**
US	**2015 / 0 127 930**	**A1**
US	**2017 / 0 310 652**	**A1**

(54) Bezeichnung: **ABGESICHERTE KOMMUNIKATION AUS EINER NICHTFLÜCHTIGEN SPEICHERVORRICHTUNG HERAUS**

(57) Hauptanspruch: Ein Gerät, das Folgendes beinhaltet:
einen Speicher-Controller;
einen statischen Direktzugriffsspeicher (SRAM), der an den Speicher-Controller gekoppelt ist, wobei der SRAM für ein Host-Rechensystem, das kommunikativ an einen Server gekoppelt ist, unzugänglich ist; und
eine an das Host-Rechensystem gekoppelte nichtflüchtige Speichervorrichtung (NVM-Vorrichtung), wobei die NVM-Vorrichtung eine Verarbeitungsvorrichtung umfasst zum:
Empfangen eines Kommunikationspakets von dem Server über das Host-Rechensystem, wobei das Kommunikationspaket eine Anforderung zum Einleiten sicherer Kommunikationen beinhaltet;
Durchführen eines sicheren Handshakes mit dem Server, über eine Kommunikation durch das Host-Rechensystem, unter Verwendung eines Sicherheitsprotokolls;
Empfangen von Daten, über das Host-Rechensystem, von dem Server innerhalb eines Sicherheitsprotokollpakets;
Speichern, als Reaktion auf das Detektieren eines Krypto-Schreib-Befehls in dem Sicherheitsprotokollpaket, des Sicherheitsprotokollpakets in einen Krypto-Puffer des SRAM;
Parsen des in dem Krypto-Puffer gespeicherten Sicherheitsprotokollpakets, um die Daten abzurufen;
Abrufen einer Sicherheitsprotokolloperations-Kennung und von Sicherheitsprotokoll-Metadaten aus einem Header des Sicherheitsprotokollpakets;
Transferieren von Abschnitten des Sicherheitsprotokollpakets von dem Krypto-Puffer zu dem SRAM; und
Verarbeiten der Abschnitte des Sicherheitsprotokollpakets aus dem SRAM heraus gemäß dem Sicherheitsprotokoll, um die ...



Beschreibung

VERWANDTE ANMELDUNGEN

[0001] Diese Anmeldung ist eine internationale Anmeldung, die die Prioritäten der US-amerikanischen Patentanmeldung Nr. 16/431,548, angemeldet am 4. Juni 2019, und der vorläufigen US Patentanmeldung Nr. 62/687,146, angemeldet am 19. Juni 2018, beansprucht.

TECHNISCHES GEBIET

[0002] Die Offenbarung bezieht sich auf das Gebiet der Speichervorrichtungen und insbesondere auf das Absichern von Kommunikation aus einer nichtflüchtigen Speichervorrichtung heraus.

HINTERGRUND

[0003] Vorrichtungen des Internets der Dinge (Internet-of-Things, IoT) umfassen ein Host-Rechensystem oder eine Host-Rechenvorrichtung, das oder die an eine externe nichtflüchtige Speicher(Non-Volatile Memory, NVM)-Vorrichtung, wie etwa eine Flash-Speichervorrichtung, gekoppelt ist, obwohl auch andere Vorrichtungen wie ferroelektrischer RAM (FRAM), magnetoresistiver RAM (MRAM) und dergleichen verwendet werden können. Anwendungen, die auf dem Host-Rechensystem laufen, laden Anwendungen oder Boot-Code in die NVM-Vorrichtung herunter, wodurch Sicherheitsrisiken entstehen, ohne dass die Möglichkeit besteht, die Quelle des Codes zu authentifizieren oder sicherzustellen, dass der Code auf dem Weg dahin, in den NVM geschrieben zu werden, nicht modifiziert wurde.

[0004] In einigen Ausführungsformen erfolgt die Sicherstellung der Authentifizierung und der sicheren Herkunft des Codes über eine Sicherheitsprotokoll-Transaktion mit dem Host-Rechensystem. Die Speicher-Controller-Einheiten vieler kleiner dimensionierter IoT-Vorrichtungen können jedoch keine Schlüssel schützen und somit die Sicherheitsprotokoll-Transaktion gefährden. Darüber hinaus sind Klartext-Daten, die von dem Host-Rechensystem in die externe NVM-Vorrichtung geschrieben werden, ungeschützt, wodurch sie durch einen Angreifer ausgelesen und/oder modifiziert werden können. Selbst wenn stattdessen verschlüsselte Daten in den NVM der NMV-Vorrichtung geschrieben werden, müssen die Daten in dem Host-Rechensystem entschlüsselt werden, wodurch die IoT-Vorrichtung für Replay-Angriffe geöffnet und die Angriffsfläche, z. B. sowohl auf dem Host-Rechensystem als auch auf der NVM-Vorrichtung, vergrößert wird.

[0005] Die Offenbarungen der US 2015/0127930 A1, US 2005/0108571 A1, US 2017/0310652 A1 und US 2014/0215111 A1 können hilfreich für das Verständnis der vorliegenden Erfindung sein.

[0006] Die US 2015/0127930 A1 betrifft eine Vorrichtung und Verfahren zur Durchführung einer Authentifizierungsverarbeitung während der Initialisierung eines Gerätes. In Übereinstimmung mit einigen Ausführungsformen hat ein Datenspeichergerät einen Hauptspeicher, der Benutzerdaten von einem Host speichert, und einen Controller mit einer Initialisierungsprogrammierung, die in einem Boot-Speicher gespeichert ist. Die Initialisierungsprogrammierung wird von der Steuereinheit ausgeführt, um das Datenspeichergerät von einem inaktiven Zustand in einen normalen Betriebsmodus zu überführen. Während eines Bootstrap-Modus erzeugt der Controller ein erstes Authentifizierungstoken, empfängt ein zweites Authentifizierungstoken als Reaktion auf das erste Authentifizierungstoken und autorisiert die Verwendung einer neuen Systemprogrammierung als Reaktion auf das zweite Authentifizierungstoken. Die neue Systemprogrammierung wird in einem lokalen Speicher des Datenspeichergeräts gespeichert und von der Steuereinheit während des normalen Betriebsmodus ausgeführt.

[0007] Die US 2005/0108571 A1 beschreibt eine sichere Kommunikation zwischen einem ressourcenbeschränkten Gerät und entfernten Netzwerkknoten über ein Netzwerk, wobei das ressourcenbeschränkte Gerät als Netzwerkknoten fungiert. Die entfernten Netzknoten kommunizieren mit dem Gerät unter Verwendung nicht modifizierter Netzwerk-Clients und -Server. Auf dem ressourcenbeschränkten Gerät implementiert ein Kommunikationsmodul ein oder mehrere Verbindungsschicht-Kommunikationsprotokolle, die mit einem Host-Computer sowie mit entfernten Netzwerkknoten kommunizieren und Netzwerksicherheitsprotokolle implementieren können, wodurch eine Sicherheitsgrenze innerhalb des ressourcenbeschränkten Geräts festgelegt wird.

[0008] Die US 2017/0310652 A1 beschreibt ein System, das Daten an eine erste Entität senden kann, um eine Verbindung zwischen der ersten Entität und einem öffentlichen Schlüssel anzuzeigen. Der öffentliche

Schlüssel soll dazu verwendet, eine kryptographisch geschützte Kommunikationssitzung zwischen der ersten Entität und einer zweiten Entität aufzubauen, Daten als Reaktion auf eine Anfrage zur Verifizierung der Verbindung zu empfangen und eine Mitteilung an die zweite Entität zu senden, dass die Daten gültig sind. Bei dem System kann es sich um einen Kryptographiedienst handeln, der teilweise von der ersten und der zweiten Entität betrieben wird.

[0009] In der US 2014/0215111 A1 werden Systeme und/oder Verfahren bereitgestellt, die den Einsatz einer variablen Leselatenz auf einem seriellen Speicherbus erleichtern. In einem Aspekt kann ein Speicher eine unbestimmte Zeitspanne nutzen, um Daten von einem Speicherarray zu erhalten und die Daten für die Übertragung auf dem seriellen Speicherbus vorzubereiten. Der serielle Speicherbus kann in einen bestimmten Zustand gebracht werden. Wenn die Daten für die Übertragung bereit sind, kann der Speicher ein Startbit auf dem seriellen Speicherbus aktivieren, um einen Host zu benachrichtigen, bevor er die Datenübertragung einleitet.

Figurenliste

[0010] Die Offenbarung wird beispielhaft, und nicht einschränkend, in den Figuren der begleitenden Zeichnungen veranschaulicht.

Fig. 1A ist ein Blockdiagramm eines Systems eines Internet-der-Dinge(IoT)-Knotens, der eine externe nichtflüchtige Speichervorrichtung (NVM-Vorrichtung) gemäß verschiedenen Ausführungsformen umfasst.

Fig. 1B ist ein Blockdiagramm des Systems der **Fig. 1A**, das zusätzliche Teilkomponenten gemäß einer Ausführungsform veranschaulicht.

Fig. 2A ist ein Blockdiagramm des Systems der **Fig. 1A**, das Fernverbindungen und einen SPI(Serial Peripheral Interface)-Bus innerhalb des IoT-Knotens gemäß einer Ausführungsform darstellt.

Fig. 2B ist ein vereinfachtes Datenflussdiagramm, das der in **Fig. 2A** veranschaulichten Hardware gemäß einer Ausführungsform entspricht.

Fig. 3 ist ein Flussdiagramm eines Verfahrens zum Initiieren einer sicheren Kommunikationssitzung und Austauschen verschlüsselter Daten zwischen einem Server und einer sicheren NVM-Vorrichtung gemäß einer Ausführungsform.

Fig. 4A ist ein Blockdiagramm der Software und Firmware der Hauptkomponenten des Systems der **Fig. 1A-1B** gemäß einer Ausführungsform.

Fig. 4B ist ein Blockdiagramm der Hardware dieser Hauptsystemkomponenten der **Fig. 1A-1B**, das ein Verfahren zur Sicherheitsprotokoll-Kommunikation zwischen dem Server und der NVM-Vorrichtung gemäß einer Ausführungsform veranschaulicht.

Fig. 5 ist ein Blockdiagramm, das die Schritte veranschaulicht, die von dem Host-Rechensystem und der NVM-Vorrichtung einer IoT-Vorrichtung unternommen werden, um einen abgesicherten Schreibbefehl auf der NVM-Vorrichtung auszuführen, der durch den Server initiiert wurde, gemäß einer Ausführungsform.

Fig. 6 ist ein Flussdiagramm eines Verfahrens zum Herstellen einer Sicherheitsprotokoll-Kommunikationssitzung zum Zwecke des Herstellens einer sicheren Kommunikation zwischen dem Server und einer sicheren NVM-Vorrichtung gemäß einer Ausführungsform.

Fig. 6A ist ein Flussdiagramm des Sicherheitsprotokoll-Handshakes, z. B. für eine Firmware-over-the-Air-Aktualisierung (FOTA-Aktualisierung), zwischen dem Server und der NVM-Vorrichtung gemäß einer Ausführungsform.

Fig. 6B ist ein Flussdiagramm des sicheren Datentransfers, z. B. für FOTA, zwischen dem Server und der NVM-Vorrichtung gemäß einer Ausführungsform.

Fig. 7 veranschaulicht eine schematische Darstellung einer Maschine in der beispielhaften Form eines Rechensystems, in dem ein Satz von Anweisungen ausgeführt werden kann, um die Maschine zu veranlassen, irgendeine oder mehrere der hierin erörterten Methodiken durchzuführen.

DETAILLIERTE BESCHREIBUNG

[0011] Um die obigen Unzulänglichkeiten beim Absichern einer Datenkommunikation zwischen einem Server und einer NVM-Vorrichtung einer IoT-Vorrichtung zu beheben (und somit die Angriffsflächen der NMV-Vorrichtung und eines gekoppelten Host-Rechensystems zu verringern), kann eine Sicherheitsprotokoll-Kommunikationssitzung direkt zwischen dem Server und der NVM-Vorrichtung hergestellt werden. Das Sicherheitsprotokoll kann zum Beispiel eines von einem SSL(Secure Sockets Layer)-Protokoll oder einem TLS (Transport Layer Security)-Protokoll sein. Dadurch kann die NVM-Vorrichtung den Server direkt authentifizieren, z. B. akzeptiert sie nur Aktualisierungen von einem vertrauenswürdigen Server. Diese Lösung ermöglicht es der NVM-Vorrichtung ferner, sich gegenüber dem Server zu authentifizieren, da Sicherheitsprotokolle die gegenseitige Authentifizierung erleichtern. Dies bedeutet, dass der Server verifizieren kann, dass von der NVM-Vorrichtung empfangene Daten auf dem Weg dorthin nicht manipuliert wurden. Durch diese Lösung wird auch das Host-Rechensystem als integraler Bestandteil der Sicherheitsprotokoll-Kommunikationssitzung entfernt.

[0012] Stattdessen verpackt das Host-Rechensystem, wie in verschiedenen Ausführungsformen noch im Detail erläutert wird, TCP(Transport Control Protocol)-Pakete von dem Server neu zu SPI(Serial Peripheral Interface)-Paketen, die für die NVM-Vorrichtung erkennbar sind, die aber immer noch ein Sicherheitsprotokollpaket umfassen, das ursprünglich innerhalb des TCP-Pakets verkapselt war. Das Host-Rechensystem kann mit der NVM-Vorrichtung SPI-Pakete austauschen, um einen Sicherheitsprotokoll-Handshake und Datentransfer zu erleichtern. In Ausführungsformen können die Firmware und Anwendungsprogrammierschnittstellen (Application Programming Interfaces, APIs) der NVM-Vorrichtung aktualisiert werden, um beim Realisieren von Aspekten eines sicheren Handshakes und Herstellen eines sicheren Datentransfers mit dem Server unter Verwendung des Sicherheitsprotokolls eine Schnittstelle mit einem Krypto-Puffer (eines statischen Direktzugriffsspeichers (Static Random Access Memory, SRAM)) und einem kryptographischen Beschleuniger (beide auf der NVM-Vorrichtung befindlich) zu bilden.

[0013] In verschiedenen Ausführungsformen empfängt das Host-Rechensystem ebenfalls SPI-Pakete von der NVM-Vorrichtung und verkapselt die SPI-Pakete zu einzelnen TCP-Paketen, die an den Server zu übertragen sind. Beim Umgang mit dem Sicherheitsprotokollpaket kann das Host-Rechensystem die verschlüsselten Daten von der NVM-Vorrichtung oder dem Server nicht entschlüsseln, da es keinen Zugriff auf den Entschlüsselungsschlüssel (z. B. einen der Sitzungsschlüssel) hat. Ferner trägt das Host-Rechensystem (außer als Kommunikationsvermittler) nicht bedeutungstragend zu der Sicherheitsprotokoll-Authentifizierung bei, die nun direkt zwischen dem Server und der NVM-Vorrichtung stattfindet.

[0014] In einer Ausführungsform umfasst ein Gerät eine nichtflüchtige Speichervorrichtung (NVM-Vorrichtung), die ihrerseits eine Verarbeitungsvorrichtung umfasst, die an ein Host-Rechensystem gekoppelt ist. Die NVM-Vorrichtung soll ein Kommunikationspaket von einem Server über das Host-Rechensystem, das an die NVM-Vorrichtung gekoppelt und an den Server kommunikativ gekoppelt ist, empfangen. Das Kommunikationspaket umfasst Klartext-Daten mit einer Anforderung zum Initiieren einer sicheren Kommunikation. Die NVM-Vorrichtung soll ferner einen sicheren Handshake mit dem Server, über eine Kommunikation durch das Host-Rechensystem, unter Verwendung eines Sicherheitsprotokolls, das einen Sitzungsschlüssel erzeugt (z. B. einen Entschlüsselungsschlüssel eines Sitzungsschlüsselpaares), durchführen. Die NVM-Vorrichtung soll ferner über das Host-Rechensystem Daten von dem Server innerhalb eines Sicherheitsprotokollpakets empfangen. Die NVM-Vorrichtung kann ferner die Daten unter Verwendung mindestens der Sicherheitsprotokoll-Metadaten, die aus dem Sicherheitsprotokollpaket abgerufen werden, authentifizieren. Die NVM-Vorrichtung kann unter Verwendung des Sitzungsschlüssels die Daten (falls verschlüsselt) auch entschlüsseln, um Klartext-Daten zu erzeugen, und die Klartext-Daten in NVM-Speicherungselementen der NVM-Vorrichtung speichern.

[0015] In Ausführungsformen ist das Host-Rechensystem nicht in der Lage, die Daten (falls verschlüsselt) zu entschlüsseln, z. B. weil es nicht über den richtigen Sitzungsschlüssel (z. B. Entschlüsselungsschlüssel) verfügt, da die Sicherheitsprotokollsitzung direkt zwischen dem Server und der NVM-Vorrichtung initiiert wurde. Das Host-Rechensystem ist auch nicht in der Lage, die Daten zu authentifizieren, da es keinen Zugriff auf Sicherheitsprotokoll-Metadaten des Sicherheitsprotokollpakets hat und keine Kenntnis davon hat, welche Chiffre-Suite für die Authentifizierung zwischen der NVM-Vorrichtung und dem Server verwendet wird.

[0016] In einer weiteren Ausführungsform umfasst ein System eine NVM-Vorrichtung, die versucht, eine Verbindung zu einem Server über ein Host-Rechensystem herzustellen und das Host-Rechensystem in vernetzter Kommunikation mit dem Server und mit der NVM-Vorrichtung. In Ausführungsformen soll das Host-

Rechensystem ein Kommunikationspaket von dem Server an die NVM-Vorrichtung übertragen, wobei das Kommunikationspaket Klartext-Daten mit einer Anforderung zum Initiieren einer sicheren Kommunikation mit der NVM-Vorrichtung umfasst. Das Host-Rechensystem kann ferner die Durchführung eines sicheren Handshakes zwischen der NVM-Vorrichtung und dem Server erleichtern, wobei der sichere Handshake ein Sicherheitsprotokoll verwendet, um die sichere Kommunikation zu initiieren, bei der die NVM-Vorrichtung einen ersten Sitzungsschlüssel erzeugt, der für das Host-Rechensystem unzugänglich ist. Das Host-Rechensystem kann ferner von dem Server ein TCP(Transport Control Protocol)-Paket empfangen, das ein Sicherheitsprotokollpaket umfasst. Das Sicherheitsprotokollpaket umfasst verschlüsselte Daten, die der Server mit einem zweiten Sitzungsschlüssel verschlüsselt hat (z. B. den Verschlüsselungsschlüssel eines durch den Server erzeugten Sitzungsschlüsselpaares). Das Host-Rechensystem kann ferner einen TCP-Header des TCP-Pakets entfernen, um das Sicherheitsprotokollpaket freizulegen. Das Host-Rechensystem kann ferner ein SPI(Serial Peripheral Interface)-Paket über Anhängen eines SPI-Krypto-Schreib-Befehls und einer Sicherheitsprotokolloperations-Kennung an das Sicherheitsprotokollpaket erzeugen. Das Host-Rechensystem kann dann das SPI-Paket an die NVM-Vorrichtung übertragen.

[0017] Auf diese Weise kann das Herstellen und Verwenden einer sicheren Kommunikationssitzung zwischen einer Quelle (wie einem Server) und der NVM-Vorrichtung eine Anzahl von Vorteilen gegenüber einem nichtsicheren Host-Rechensystem bieten. Diese Vorteile umfassen die Fähigkeit, eine Quelle der Downloads oder Uploads zu authentifizieren, die Vertrauenswürdigkeit der heruntergeladenen oder hochgeladenen Daten sicherzustellen und die Integrität und Authentizität von heruntergeladenen und hochgeladenen Daten sicherzustellen.

[0018] In Ausführungsformen kann das Herstellen und Verwenden einer sicheren Kommunikationssitzung zwischen einer Quelle (wie einem Server) und der NVM-Vorrichtung eine Anzahl von Vorteilen selbst gegenüber einer Verwendung eines sicheren Host-Rechensystem bieten. Diese Vorteile umfassen, dass durch das Entfernen des Host-Rechensystems als mögliche Angriffsschicht die sichere Verbindung eine kleinere Angriffsfläche aufweist. Genauer gesagt werden die Klartext-Daten, die als Chiffretext-Daten an die NVM-Vorrichtung kommuniziert werden, nicht an irgendeinem Punkt entlang des Kommunikationsweges entschlüsselt, sondern erst dann, wenn sie ankommen und in einem sicheren kryptographischen Puffer der NVM-Vorrichtung gepuffert werden, wie noch näher erläutert wird. Darüber hinaus ist die Implementierung einer sicheren Lösung, die mit nichtsicheren Benutzeranwendungen des Host-Rechensystems koexistiert, nicht trivial. Eine Sicherheitsprotokollverbindung direkt zwischen dem Server und dem NVM-Speicher vereinfacht (und eliminiert möglicherweise) die Notwendigkeit, eine sichere Lösung für solche nichtsicheren Benutzeranwendungen zu implementieren.

[0019] Fig. 1A ist ein Blockdiagramm eines Systems 100 eines Internet-der-Dinge(IoT)-Knotens 101, der eine externe nichtflüchtige Speichervorrichtung (NVM-Vorrichtung) gemäß verschiedenen Ausführungsformen umfasst. Der IoT-Knoten 101 wird in einigen Kontexten auch als Kantenvorrichtung bezeichnet. Das System 100 wird an (oder über) ein Netzwerk 115 angeschlossen, das auch als Cloud bezeichnet wird und im Allgemeinen als eine oder mehrere Backbone-Verbindungen durch das Internet verstanden werden kann. Das System 100 kann ferner einen Server 105 umfassen, mit dem sich der IoT-Knoten 101 über das Netzwerk 115 verbinden soll, um Firmware-Aktualisierungen zu empfangen, Sensordaten oder anderen Signalgehalt bereitzustellen und dergleichen. Der Server 105 wird im Allgemeinen die Kommunikation über das TCP (Transport Control Protocol)-Internetprotokoll (IP), z. B. über TCP/IP, kommunizieren.

[0020] In verschiedenen Ausführungsformen kann der IoT-Knoten 101 als Multi-Chip-Modul oder als Halbleiterbaueinheit instanziiert sein und umfasst ein Host-Rechensystem 102, das an eine nichtflüchtige Speichervorrichtung (NVM-Vorrichtung) 110 gekoppelt ist. Die NVM-Vorrichtung 110 kann eine Flash-Vorrichtung, eine Festkörperspeicherungsvorrichtung, ein ferroelektrischer RAM (FRAM), ein magnetoresistiver RAM (MRAM) oder eine andere nichtflüchtige Speichervorrichtung sein.

[0021] In Ausführungsformen umfasst das Host-Rechensystem 102 neben anderen Komponenten einen Prozessor 104, eine Speicher-Controller-Einheit (Memory Controller Unit) 106 (z. B. eine Master-MCU) und einen Brückentreiber 108. Wie noch näher erläutert wird, kann das Host-Rechensystem 102 an die NVM-Vorrichtung 110 über einen Bus 117, wie zum Beispiel einen SPI-Bus (Serial Peripheral Interface Bus), einen 12C-Bus (Inter-Integrated Circuit Bus) oder eine andere Art von Bustransferprotokoll, gekoppelt werden. Der Brückentreiber 108 kann angepasst sein, um Übertragungssteuerungspakete (Transmission Control Packets, TCP) in SPI-Pakete und umgekehrt umzuwandeln, um die Kommunikation zwischen dem Server 105 und der NVM-Vorrichtung 110 zu erleichtern, wie noch näher erläutert wird.

[0022] In Ausführungsformen umfasst die NVM-Vorrichtung 110 eine Kommunikationsschnittstelle 130, einen Mikrocontroller 118 (z. B. eine Verarbeitungsvorrichtung der NVM-Vorrichtung 110), NVM-Speicherungselemente 120 (die als Speicherungs-Array von NVM-Speicherungszellen organisiert sein können), einen Slave-Speicher-Controller (Slave Memory Controller, SMC) 122, einen statischen Direktzugriffsspeicher (Static Random Access Memory, SRAM) 126 und einen kryptographischen („Krypto“)-Beschleuniger 140. Die Kommunikationsschnittstelle 130 kann einen Lese-/Schreib-Port 136 umfassen. Der SMC 122 kann z. B. einen SMC-Puffer 124 und einen SPI-Befehlsdecodierer 125, der SPI-basierte Befehle decodiert, umfassen. Ferner kann der SRAM 126 einen Krypto-Puffer 128 umfassen, der SPI-Pakete puffern soll, die kryptographische Operationen umfassen. In Ausführungsformen empfängt der SMC 122 Lese- und Schreiboperationen von der MCU 106 des Host-Rechensystems 102 und weist den Abschluss der Lese- und Schreibbefehle unter Bezugnahme auf den SRAM 126 und den NVM 120 an.

[0023] **Fig. 1B** ist ein Blockdiagramm des Systems 100 der **Fig. 1A**, das zusätzliche Teilkomponenten gemäß einer Ausführungsform veranschaulicht. In weiteren Ausführungsformen umfasst das Host-Rechensystem 102 z. B. einen SPI-Master 152, der Teil der MCU 106 sein kann. Des Weiteren umfasst der SMC 122 der NVM-Vorrichtung 110 ferner einen SPI-Slave 154 zur Kommunikation unter Verwendung von SPI-Paketen über den Bus 117 im Austausch mit dem SPI-Master 152. Tabelle 1 veranschaulicht einen Satz von beispielhaften Anwendungsprogrammierschnittstellen (APIs), die durch den SPI-Master 152 für die Kommunikation mit dem SPI-Slave 154 verwendbar ist, wobei „SPI“ für serielle periphere Schnittstelle (Serial Peripheral Interface) und „TCP“ für Übertragungssteuerungsprotokoll (Transmission Control Protocol) steht, z. B. unter Bezugnahme auf TCP von TCP/IP.

Tabelle 1

API	Funktion
spi_write()	Schreiben in SMC-Puffer 124
spi_read()	Lesen aus SMC-Puffer 124
tcp_to_spi()	Teilen eines einzelnen TCP-Pakets in N SPI-Pakete
spi_to_tcp()	Zusammenführen von N SPI-Paketen zu einem einzigen TCP-Paket

[0024] In Ausführungsformen kommuniziert der Mikrocontroller 118 über eine System-on-a-Chip(SoC)-Busarchitektur, wie z. B. Advanced-High-Performance-Bus(AHB)-Lite. Somit umfasst in einer Ausführungsform der Mikrocontroller 118 einen AHB-Lite-Master 148. Das AHB-Lite-Protokoll ist ein offener Standard der Advanced Microcontroller Bus Architecture (AMBA), der eine chipinterne Verschaltungsspezifikation für die Verbindung und die Verwaltung von Funktionsblöcken in einer SoC-Ausgestaltung bereitstellt. Obwohl hierin auf AHB-Lite Bezug genommen wird, sind andere Mikrocontroller-Bus-Architekturen denkbar.

[0025] In entsprechenden Ausführungsformen umfassen der SMC 122 und der Krypto-Beschleuniger 140 jeweils einen AHB-Lite-Slave 156A bzw. 156B, mit denen der AHB-Lite-Master 148 kommunizieren kann. Der Mikrocontroller 118 kann mit dem Krypto-Beschleuniger 140 zusammenarbeiten, um die kryptographischen Operationen durchzuführen, welche die Initiierung einer sicheren Kommunikationssitzung mit dem Server 105 ermöglichen, und um während des Datentransfers verschlüsselte Daten mit dem Server 105 auszutauschen, wie noch näher erörtert wird. In einer Ausführungsform ist der Krypto-Beschleuniger 140 eine FPGA(Field Programmable Gate Array)-Vorrichtung, die mit mxCrypto programmiert ist, welches eine umfangreiche Werkzeugsammlung ist, die Python-Erweiterungen für Verschlüsselung, Authentifizierung, Schlüsselaustausch, Secure-Socket-Operation, Transportschicht-Sicherheitsoperation und andere Arten von kryptographischen Operationen umfasst. Andere Arten von kryptographischen Werkzeugsammlungen sind denkbar.

[0026] In einer Ausführungsform veranschaulicht Tabelle 2 beispielhafte APIs, die durch den AHB-Master 148 des Mikrocontrollers 118 verwendet werden, um mit dem AHB-Lite-Slave 156A des SMC 122 zu kommunizieren. In einer Ausführungsform veranschaulicht Tabelle 3 beispielhafte APIs, die durch den AHB-Master 148 des Mikrocontrollers 118 verwendet werden, um mit dem AHB-Lite-Slave 156B des Krypto-Beschleunigers 140 zu kommunizieren.

Tabelle 2

API	Funktion
smc_to_sram()	Transfer von Daten vom SMC-Puffer zum SRAM
sram_to_smc()	Transfer von Daten vom SRAM zum SMC-Puffer

Tabelle 3

API	Funktion
mxcrypto_verify_signature()	Verwenden von mxCrypto-Fähigkeiten zur Signaturverifizierung
mxcrypto_sign()	Verwenden von mxCrypto-Fähigkeiten zum Signieren von Daten
mxcrypto_calculate_ec_point( )	Verwenden von mxCrypto zum Berechnen eines elliptischen Kurvenpunktes
mxcrypto_sha256( )	Berechnen des SHA256-Digest
decrypto_and_verify_hmac( )	Entschlüsseln der Daten, dann deren HMAC-Wert prüfen

[0027] **Fig. 2A** ist ein Blockdiagramm des Systems 100 der **Fig. 1A**, die Fernverbindungen und den SPI-Bus 117 innerhalb des IoT-Knotens 101 gemäß einer Ausführungsform veranschaulicht. In Ausführungsformen ist der Server 105 an den IoT-Knoten 101 über ein Paar von Fernverbindungen an die Cloud, z. B. das Netzwerk 115, kommunikativ gekoppelt. Der IoT-Knoten 101 kann das Host-Rechensystem 102 und die NVM-Vorrichtung 110 umfassen, die durch den SPI-Bus 117 gekoppelt sind.

[0028] **Fig. 2B** ist ein vereinfachtes Datenflussdiagramm 200, das der in **Fig. 2A** veranschaulichten Hardware gemäß einer Ausführungsform entspricht. In Ausführungsformen veranschaulicht das Datenflussdiagramm 200 den Datenfluss, der mit den auf dem Server 105 gespeicherten Klartext-Daten beginnt. Der Server 105 kann die Klartext-Daten mit TLS (oder einem anderen sicheren Internetprotokoll oder einem zugehörigen kryptographischen Algorithmus) verschlüsseln, um Chiffretext-Daten zu erzeugen (210). Der Server 105 kann die Chiffretext-Daten auch in einem TLS-Paket verpacken und das TLS-Paket innerhalb eines TCP/IP-Pakets senden (220).

[0029] In Ausführungsformen empfängt der IoT-Knoten 101 (z. B. das Host-Rechensystem 102) das TCP/IP-Paket (225). Der Brückentreiber 108 des Host-Rechensystems 102 kann dann die Chiffretext-Daten in Abschnitte von verschlüsselten Daten partitionieren, um aus jedem Abschnitt der verschlüsselten Daten ein SPI-Paket zu erzeugen (230). In Ausführungsformen leitet der SPI-Bus 117 die SPI-Pakete an die NVM-Vorrichtung 110 weiter (240). Jedes SPI-Paket kann das TLS-Paket mit dem verschlüsselten Abschnitt umfassen (240). Die NVM-Vorrichtung 110 kann dann z. B. unter Verwendung des Krypto-Beschleunigers 140 den verschlüsselten Teil entschlüsseln, um wieder die Klartext-Daten zu erzeugen (250). Gleichzeitig oder in Verbindung mit der Entschlüsselung kann die NVM-Vorrichtung 110 die Daten zeilenweise authentifizieren, während sie entschlüsselt werden und bevor sie auf das NVM 120 programmiert werden. Die Schritte dieses vereinfachten Datenflussdiagramms 200 werden unter Bezugnahme auf **Fig. 4** mit einer beispielhaften Implementierung erklärt.

[0030] **Fig. 3** ist ein Flussdiagramm eines Verfahrens 300 zum Initiieren einer sicheren Kommunikationssitzung und Austauschen verschlüsselter Daten zwischen einem Server und einer sicheren NVM-Vorrichtung gemäß einer Ausführungsform. Das Verfahren 300 kann durch Verarbeitungslogik durchgeführt werden, die Hardware (z. B. Beschaltung, speziell dafür vorgesehene Logik, programmierbare Logik, Mikrocode usw.), Software (wie Anweisungen, die auf einer Verarbeitungsvorrichtung laufen), Firmware oder eine Kombination davon beinhalten kann. In einer Ausführungsform wird das Verfahren 300 durch verschiedene Komponenten der NVM-Vorrichtung 110 durchgeführt.

[0031] Unter Bezugnahme auf **Fig. 3** kann das Verfahren 300 damit beginnen, dass die Verarbeitungslogik eine Verbindung zu dem Server 105 über das Host-Rechensystem 102 initiiert (310). Als Reaktion auf den Versuch kann das Verfahren 300 damit fortfahren, dass die Verarbeitungslogik ein Kommunikationspaket von dem Server 105 über das Host-Rechensystem 102, das kommunikativ an den Server gekoppelt ist, empfängt (320). Das Kommunikationspaket kann eine Anforderung zum Initiieren sicherer Kommunikationen umfassen. Das Verfahren 300 kann damit fortfahren, dass die Verarbeitungslogik einen sicheren Handshake

mit dem Server 105, über Kommunikation durch das Host-Rechensystem 102, durchführt, wobei ein Sicherheitsprotokoll verwendet wird, das einen Sitzungsschlüssel (z. B. einen Entschlüsselungsschlüssel eines Sitzungsschlüsselpaares) erzeugt (330). In Ausführungsformen erfolgt der sichere Handshake über eine Übertragung von Daten innerhalb von SPI-Paketen zwischen dem SPI-Slave 154 (des SMC 122) und dem SPI-Master 152 (des Host-Rechensystems 102).

[0032] Unter fortgesetzter Bezugnahme auf **Fig. 3** kann das Verfahren 300 damit fortfahren, dass die Verarbeitungslogik verschlüsselte Daten über das Host-Rechensystem 102 von dem Server 105 innerhalb eines Sicherheitsprotokollpakets empfängt (340). Das Host-Rechensystem 102 ist nicht in der Lage, die verschlüsselten Daten zu entschlüsseln. Das Verfahren 300 kann damit fortfahren, dass die Verarbeitungslogik unter Verwendung des Sitzungsschlüssels die verschlüsselten Daten entschlüsselt, um Klartext-Daten zu erzeugen (350). Das Verfahren 300 kann damit fortfahren, dass die Verarbeitungslogik die Klartext-Daten in NVM-Speicherungselementen der NVM-Vorrichtung speichert (360). Das Verfahren 300 kann damit fortfahren, dass die Verarbeitungslogik für das Host-Rechensystem 102 Zugriff auf die in den NVM-Speicherungselementen gespeicherten Klartext-Daten bereitstellt (370).

[0033] **Fig. 4A** ist ein Blockdiagramm der Software und Firmware (SW/FW) 400 der Hauptkomponenten des Systems 100 der **Fig. 1A-1B** gemäß einer Ausführungsform. Die SW/FW 400 kann zum Beispiel eine Anwendung 405 (etwa einen Firmware-Aktualisierungs-Produzenten), den SSL- oder TLS-Stapel 410 und den TCP/IP-Stapel 415 umfassen, die alle auf dem Server 105 laufen können. Die SW/FW 400 kann ferner den TCP/IP-Stapel 420 und einen SPI-Treiber 425 umfassen, die auf dem Host-Rechensystem 102 laufen. Der SPI-Treiber 425 kann mit dem Brückentreiber 108 des Host-Rechensystems 102 identisch oder in diesen integriert sein. Die SW/FW 400 kann ferner die SMC-Firmware 430, den SSL/TLS-Stapel 435 und eine Anwendung 440 (z. B. einen Firmware-Aktualisierungs-Verbraucher) umfassen, die auf der NVM-Vorrichtung 110 laufen.

[0034] **Fig. 4B** ist ein Blockdiagramm der Hardware dieser Hauptsystemkomponenten der **Fig. 1A-1B** und veranschaulicht ein Verfahren 450 für eine Sicherheitsprotokoll-Kommunikation zwischen dem Server 105 und der NVM-Vorrichtung 110 unter Verwendung des Host-Rechensystems 102 als Vermittler gemäß einer Ausführungsform. In einer Ausführungsform ist das Host-Rechensystem ein Zynq-7000-FPGA und die NVM-Vorrichtung ein Kintex-7-FPGA.

[0035] Das Verfahren 450 kann damit beginnen, dass der Server 105, z. B. innerhalb einer SSL/TLS-Sitzung, Klartext-Daten 451 in Chiffretext-Daten 454 (auch als verschlüsselte Daten bezeichnet) unter Verwendung eines Verschlüsselungsalgorithmus 453 (z. B. mit Advanced Encryption Standard (AES) im Galois/-Counter Mode (GCM), der auch Daten authentifiziert) verschlüsselt werden, wobei als Eingaben ein Schreibinitialisierungsvektor (IV) und ein Verschlüsselungsschlüssel verwendet werden. Der Verschlüsselungsschlüssel kann einer der Sitzungsschlüssel sein, die während eines Handshake-Prozesses mit der NVM-Vorrichtung 110 erzeugt werden, wie noch näher erläutert wird. Das Verfahren 450 kann damit fortfahren, dass der Server 105 die Chiffretext-Daten 454 in ein TCP-Paket 455, das einen TCP-Header aufweist, verkapselt. Die Verkapselung der Chiffretext-Daten 454 umfasst einen TLS-Header (oder einen Header eines anderen Sicherheitsprotokolls), welche zusammen als Sicherheitsprotokollpaket 456 bezeichnet werden können. Der TLS-Header kann gewisse TLS-basierte Metadaten (oder Metadaten, die auf einem anderen Sicherheitsprotokoll basieren) umfassen, die zum Transportieren verwendet werden und die für die TLS-basierte Kommunikationssitzung spezifisch sind.

[0036] Das Verfahren 450 kann damit fortfahren, dass der Server 105 das TCP-Paket 455 an das Host-Rechensystem 102 überträgt 457. Das Host-Rechensystem 102 kann deshalb von dem Server 105 das TCP-Paket 455, welches das Sicherheitsprotokollpaket 456 umfasst, empfangen. Wie erörtert, kann das Sicherheitsprotokollpaket 456 die Chiffretext-Daten 454 (z. B. verschlüsselte Daten) und den TLS-Header umfassen. Das Host-Rechensystem 102 kann ferner den TCP-Header des TCP-Pakets 455 entfernen, um das Sicherheitsprotokollpaket 456 freizulegen. Das Verfahren 450 kann damit fortfahren, dass das Host-Rechensystem 102 ein SPI(Serial Peripheral Interface)-Paket 461 über Anhängen eines SPI-Krypto-Schreib-Befehls (CMD-ID) und einer Sicherheitsprotokolloperations-Kennung (z. B. TLS-OP) an das Sicherheitsprotokollpaket erzeugt und das SPI-Paket 461 an die NVM-Vorrichtung 110 überträgt 465. Ein Krypto-Lesebefehl kann zwar auf ähnliche Weise übertragen werden, aber der Krypto-Lesebefehl würde nicht von den Chiffretext-Daten begleitet werden.

[0037] Das Verfahren 450 kann damit fortfahren, dass die NVM-Vorrichtung 110 das SPI-Paket 461 von dem Host-Rechensystem 102 empfängt. Das Verfahren 450 kann damit fortfahren, dass die NVM-Vorrichtung

unter anderem, wie unter Bezugnahme auf **Fig. 5** noch nähert erörtert wird, die Chiffretext-Daten 454 unter Verwendung eines Entschlüsselungsalgorithmus (z. B. des zuvor erörterten AES-GCM-Algorithmus) entschlüsselt 469, wobei als Eingaben der Schreibinitialisierungsvektor und ein Entschlüsselungsschlüssel verwendet werden. Der Entschlüsselungsschlüssel kann ein Sitzungsschlüssel sein, der während des sicheren Handshakes zwischen dem Server 105 und der NVM-Vorrichtung 110 erzeugt wird. Das Entschlüsseln kann die Klartext-Daten 451 erzeugen, die ursprünglich durch den Server 105 verschlüsselt wurden.

[0038] Auf diese Weise wandelt das Host-Rechensystem 102 das TCP-Paket 455 in ein SPI-Paket 461 um, über Vertauschen und/oder Entfernen gewisser Teile von deren Headern, wobei das SPI-Paket 461 dann über den SPI-Bus 117 an die NVM-Vorrichtung 110 übertragen werden kann und durch diese lesbar ist. Dabei liest das Host-Rechensystem 102 die Chiffretext-Daten 454 nicht, sondern leitet sie an die NVM-Vorrichtung 110 weiter. Würde ein Angreifer auch nur versuchen, auf die Chiffretext-Daten 454 auf dem Host-Rechensystem 102 zuzugreifen, wären die Daten ohne den Entschlüsselungsschlüssel bedeutungslos. Der Entschlüsselungsschlüssel ist jedoch nicht auf dem Host-Rechensystem 102 gespeichert und ist für das Host-Rechensystem 102 unzugänglich, da er in dem Krypto-Puffer 128 des SRAM 126 der NVM-Vorrichtung 110 gespeichert ist. Dadurch wird die Angriffsfläche auf dem Host-Rechensystem eliminiert und die sichere direkte Kommunikation zwischen dem Server 105 und der NVM-Vorrichtung 110 erheblich verbessert.

[0039] **Fig. 5** ist ein Blockdiagramm, das die Schritte veranschaulicht, die von dem Host-Rechensystem 102 und der NVM-Vorrichtung 110 (der loT-Vorrichtung 101) unternommen werden, um einen abgesicherten Schreibbefehl auf der NVM-Vorrichtung 110 auszuführen, der durch den Server 105 initiiert wurde, gemäß einer Ausführungsform. Wie unter Bezugnahme auf **Fig. 4B** erörtert, kann das Host-Rechensystem 102 das SPI-Paket 461 durch Anhängen eines Krypto-Schreib-Befehls, der mit einer Befehlskennung (CMD-ID) gekennzeichnet ist, und einer Sicherheitsprotokolloperations-Kennung (z. B. die TLD-OP) an das Sicherheitsprotokollpaket 456 erzeugen (**Fig. 4B**).

[0040] In verschiedenen Ausführungsformen kann ein SPI-Befehlsdecodierer 125 in dem SMC 122 der NVM-Vorrichtung 110 den Krypto-Schreib-Befehl (CMD-ID) innerhalb des SPI-Pakets detektieren. Als Reaktion auf das Detektieren der kryptographischen Speicheroperation, die durch die CMD-ID gekennzeichnet ist (die in anderen Fällen auch ein Krypto-Lesebefehl sein kann), kann der SMC 122 das SPI-Paket (abzüglich der CMD-ID) in dem Krypto-Puffer 128 puffern. Der Mikrocontroller 118 kann dann das SPI-Paket parsen, um verschlüsselte Daten (z. B. Chiffretext-Daten) abzurufen und die Sicherheitsprotokolloperations-Kennung (TLS-OP) und Sicherheitsprotokoll(oder SPI)-Metadaten aus dem TLS-Header abzurufen. Der Mikrocontroller 118 kann Abschnitte des SPI-Pakets aus dem Krypto-Puffer in den SRAM transferieren. An diesem Punkt kann der Mikrocontroller 118 die Ausführung des Sicherheitsprotokolls (z. B. TLS in diesem Beispiel) anweisen, um die sichere Schreiboperation abzuschließen. Die Ausführung des Sicherheitsprotokolls kann das Verarbeiten der Abschnitte des SPI-Pakets aus dem SRAM heraus gemäß dem Sicherheitsprotokoll umfassen, worin die Verifizierung der Sicherheitsprotokoll-Metadaten umfasst ist.

[0041] Genauer gesagt kann der Mikrocontroller 118 den kryptographischen Beschleuniger 140 anweisen, die Chiffretext-Daten zu entschlüsseln, um die Klartext-Daten 451 (**Fig. 4B**) innerhalb des Sicherheitsprotokollpakets, das der Server 105 ursprünglich verschlüsselt hat, zu erzeugen. Die Daten innerhalb des Sicherheitsprotokollpakets können einen Programmier-Lösch(Program Erase)-Befehl (P/E-Befehl) (z. B. zum Schreiben in eine Festkörperspeicherungsvorrichtung) oder eine andere spezifische Art von Schreibbefehl, eine Zieladresse (z. B. in dem Benutzer-Array des NVM 120), eine Länge der Zieldaten und die Zieldaten selbst (alle veranschaulicht) umfassen. Der Programmier-Lösch- oder P/E-Befehl kann die Löschung des gesamten Benutzer-Array von NVM-Speicherungselementen (z. B. wenn der NVM 120 ein EEPROM ist) anweisen oder Blöcke von NVM-Speicherungselementen (z. B. wenn der NVM 120 ein Flash-Speicher ist) auswählen. In einigen Ausführungsformen wird kein P/E-Befehl empfangen, deshalb wird ein Löschbefehl impliziert und es werden genügend Abschnitte des NVM 120 gelöscht, so dass die in den NVM 120 zu schreibenden Daten aufgenommen werden. Der Mikrocontroller 118 kann dann das Benutzer-Array oder den Abschnitt des Benutzer-Array (z. B. die NVM-Speicherungselemente an der Zieladresse) unter Verwendung der Zieladresse und der Längeninformation aus dem entschlüsselten SPI-Paket mit den Zieldaten programmieren. Wenn dies ein sicherer Lesebefehl wäre, gäbe es keine Zieldaten, und der Mikrocontroller würde an einer Zieladresse einen sicheren Lesevorgang von Daten einer bestimmten Länge durchführen.

[0042] **Fig. 6** ist ein Flussdiagramm eines Verfahrens 600 zum Herstellen einer Sicherheitsprotokoll-Kommunikationssitzung zum Zwecke des Herstellens einer sicheren Kommunikation zwischen dem Server und einer sicheren NVM-Vorrichtung gemäß einer Ausführungsform. Das Verfahren 600 kann durch Verarbeitungslogik durchgeführt werden, die Hardware (z. B. Beschaltung, speziell dafür vorgesehene Logik, programmierbare

Logik, Mikrocode usw.), Software (wie Anweisungen, die auf einer Verarbeitungsvorrichtung laufen), Firmware oder eine Kombination davon beinhalten kann. In einer Ausführungsform wird das Verfahren 600 durch verschiedene Komponenten der NVM-Vorrichtung 110 in Kommunikation mit dem Server 105 durchgeführt.

[0043] Unter Bezugnahme auf **Fig. 6**, kann das Verfahren 600 damit beginnen, dass die Verarbeitungslogik eine TCP-Verbindung zwischen der NVM-Vorrichtung 110 und dem Server 105 herstellt (610). Das Verfahren 600 kann damit fortfahren, dass die Verarbeitungslogik prüft, ob Firmware-Aktualisierungen von dem Server 105 vorliegen, obwohl es andere Gründe dafür geben kann, dass die NVM-Vorrichtung 110 mit dem Server kommuniziert (620). Das Verfahren 600 kann damit fortfahren, dass die Verarbeitungslogik einen Sicherheitsprotokoll-Handshake durchführt, wie er auch beim Initiieren einer SSL- oder TLS-Sitzung unter Verwendung von HTML-Code auftritt (630). Der sichere Handshake kann eine Reihe von Sequenzierungsoperationen umfassen, die zu einer Reihe von kryptographischen Operationen führen, welche einen oder mehrere Sitzungsschlüssel erzeugen (z. B. mindestens einen Sitzungsschlüssel für die NVM-Vorrichtung und denselben Sitzungsschlüssel für den Server). Das Verfahren 600 kann damit fortfahren, dass die Verarbeitungslogik einen sicheren Datentransfer (oder Datenaustausch) mit dem Server 105 durchführt (670). Das Verfahren 600 kann damit fortfahren, dass die Verarbeitungslogik die TCP-Verbindung beendigt (695).

[0044] Fig. 6A ist ein Flussdiagramm des Sicherheitsprotokoll-Handshakes 630, z. B. für eine Firmware-over-the-Air-Aktualisierung (FOTA-Aktualisierung), zwischen dem Server und der NVM-Vorrichtung, die auch als Client-Vorrichtung bezeichnet wird, gemäß einer Ausführungsform. Der Sicherheitsprotokoll-Handshake 630 kann in einer Anzahl von Phasen durchgeführt werden und umfasst verschiedene kryptographische Transaktionen und/oder Sicherheitsprotokoll-Transaktionen, um die Initiierung einer Sicherheitsprotokoll-Kommunikationssitzung zu realisieren. Diese Transaktionen können sich abhängig von dem Protokoll und von der Version des Protokolls, das gerade verwendet wird, ändern. Dementsprechend wird ein allgemeiner Rahmen erörtert und der Kommunikationsmechanismus wird erklärt. Bei der Erklärung wird auf die in den Tabellen 1-3 aufgeführten APIs verwiesen, und es werden weitere mögliche APIs vorgeschlagen, und es wird auf den allgemeinen Kommunikationsfluss verwiesen, der unter Bezugnahme auf **Fig. 1B** beschrieben wurde.

[0045] In verschiedenen Ausführungsformen geht Phase 1 als Reaktion auf eine Antwort von dem Server 105, dass eine Aktualisierung erforderlich ist, weiter und umfasst das Senden einer „Hallo“-Mitteilung an den Server 105 (632). Diese Client-Hallo-Mitteilung soll dem Server mitteilen, welche Funktionalität die NVM-Vorrichtung unterstützen kann, z. B. eine Liste von Chiffre-Suites. Eine Chiffre-Suite ist eine Kombination von kryptographischen Primitiven, die zur Authentifizierung und Verschlüsselung/Entschlüsselung verwendbar sind. In Phase 1 kann der Mikrocontroller 118 ein tls_create_packet(„Client-Hallo“) erzeugen, das an den SMC 122 gesendet wird. Der SMC 122 kann dann eine spi_read(„Client-Hallo“)-Mitteilung erzeugen, die an das Host-Rechensystem 102 gesendet wird. Das Host-Rechensystem 102 kann dann die spi_read(„Client-Hallo“)-Mitteilung in eine tcp_write(„Client-Hallo“)-Mitteilung umwandeln, die an den Server 105 gesendet wird.

[0046] In verschiedenen Ausführungsformen wird Phase 2 damit fortgesetzt, dass der Server 105 an die NVM-Vorrichtung 110 mit einem „Hallo“ antwortet, möglicherweise einen Serverschlüsselaustausch und eine Zertifikatssignaturverifikation durchführt (634). Dieses Server-Hallo kann eine Auswahl aus der von der NVM-Vorrichtung 110 empfangenen Liste möglicher Chiffre-Suiten umfassen, die an den Client kommuniziert wird. Darüber hinaus kann der Server 105 für die Zertifikatssignaturverifikation sein Zertifikat in einer tcp_write(„Zertifikats“)-Mitteilung an das Host-Rechensystem 102 senden. Das Host-Rechensystem 102 kann eine tcp_to_spi()-Umwandlung durchführen, um ein spi_write(„Zertifikat“) zu erzeugen, das an den SMC 122 gesendet wird. Der SMC 122 kann ein smc_to_sram(„Zertifikat“) erzeugen, das an den Mikrocontroller 118, z. B. eine Verarbeitungsvorrichtung, gesendet wird. Der Mikrocontroller 118 kann das smc_to_sram(„Zertifikat“) in eine tls_Prozess(„Zertifikats“)-Mitteilung umwandeln, die ein mxcrypto_verify_signature(„Zertifikat“) an den Krypto-Beschleuniger 140 auslöst.

[0047] In Ausführungsformen kann eine ähnliche Reihe von Schritten (wie eben beschrieben) durchgeführt werden, um den Serverschlüsselaustausch, der von dem Server initiiert wird, mit oder ohne Zugriff des Krypto-Beschleunigers 140 am Ende des Austauschs auszuführen. Der Serverschlüsselaustausch kann das Austauschen von Sitzungsschlüsseln ermöglichen, so dass die NVM-Vorrichtung und der Server einen gemeinsamen Satz von Sitzungsschlüsseln verwenden, z. B. mindestens einen Verschlüsselungsschlüssel und einen Entschlüsselungsschlüssel für den Server und einen weiteren Satz von Verschlüsselungs- und Entschlüsselungsschlüsseln für die NVM-Vorrichtung 110. Eine Schreibe-Zertifikat-Anforderung kann mit einer ähnlichen Reihe von Schritten durchgeführt werden, bei denen der Server das Zertifikat des Clients

von der NVM-Vorrichtung in einer „Zertifikatsanforderungs“-Mitteilung anfordert. Um Phase 2 abzuschließen, kann der Server 105 eine tcp_write(„Server-Hallo-Erledigt“)-Mitteilung senden, die gesendet und umgewandelt wird, bis sie an den Mikrocontroller 118 gemeldet wird. Die Server-Hallo-Erledigt-Mitteilung kann angeben, dass der Server mehr Informationen benötigt, um fortzufahren.

[0048] In verschiedenen Ausführungsformen wird Phase 3 fortgesetzt mit der Zertifikatspaketerstellung, dem Client-Schlüsselaustausch, dem gemeinsamen elliptischen Kurvenpunkt von Server und NVM-Vorrichtung, eine kryptographische Signatur (z. B. mxcrypto_sign) wird auf die Handshake-Mitteilungen angewendet und ein kryptographischer Algorithmus (z. B. mxcrypto_sha256) wird auf die Handshake-Mitteilungen angewendet (638). In Ausführungsformen wird der Client-Schlüsselaustausch durch den Mikrocontroller 118 initiiert und am Server abgeschlossen, z. B. mit einer Umwandlung des Client-Zertifikats von einem spi_read(„Zertifikat“) in ein tcp_write(„Zertifikat“) am Host-Rechensystem 102. Der Mikrocontroller 118 kann dann mxcrypto_sign(„Handshake-Mitteilungen“) mit dem Krypto-Beschleuniger 140 initiieren, um eine Sicherheitsprotokoll-Signatur auf die Handshake-Mitteilungen anzuwenden.

[0049] Der Mikrocontroller 118 kann dann einen Zertifikatsverifizierungsprozess initiieren, der durch den SRAM zu dem SMC, zu dem Host-Rechensystem und weiter zu dem Server geht. Die Zertifikatsverifizierung kann eine Signatur früherer Mitteilungen über eine „Zertifikatsverifizierungs“-Mitteilung umfassen. Der Mikrocontroller 118 kann ferner einen Chiffre-Spezifikationsänderungs(„Change Cipher Spec“)-Prozess initiieren, der über die gleiche Reihe von Komponenten und mit der gleichen Umwandlung zwischen SPI- und TCP-Paketen an den Server 105 übertragen wird. Diese Client-„Change-Cipher-Spec“-Mitteilung soll dem Server mitteilen, dass die NVM-Vorrichtung bereit ist, die von dem Server gewählte Chiffre-Suite zu verwenden. Der Mikrocontroller 118 kann ferner die Anwendung eines kryptographischen Algorithmus (z. B. mxcrypto_sha256(„Handshake-Mitteilungen“) mit dem kryptographischen Beschleuniger 140 initiieren, um TLS-Mitteilungen mit einem Verschlüsselungsschlüssel der Sitzungsschlüssel zu verschlüsseln.

[0050] In verschiedenen Ausführungsformen wird Phase 4 damit fortgesetzt, dass der Server 105 ein tcp_write(„change cipher spec“) an das Host-Rechensystem 102 sendet, das über eine tcp_to_spi()-API in ein spi_write(„change cipher spec“) umgewandelt und an den SMC 122 gesendet wird. Diese Server-„Change-Cipher-Spec“-Mitteilung erlaubt es dem Server, der NVM-Client-Vorrichtung zu bestätigen, dass der Server bereit ist, die gewählte Chiffre-Suite zu verwenden. An der NVM-Vorrichtung kann der SMC über den Mikrocontroller 118 den Befehl zur Änderung der Chiffre-Spezifikation an den SRAM senden. Der Server 105 kann dann eine „Beendet“-Mitteilung initiieren, die durch das Host-Rechensystem 102, den SMC und weiter an den Mikrocontroller 118 der NVM-Vorrichtung gesendet wird. Diese „Beendet“-Mitteilung oder der „Beendet“-Befehl kann dazu da sein, der NVM-Vorrichtung mitzuteilen, dass die laufende Kommunikation nun durch die Chiffre-Suite abgesichert ist, die der Server aus den auf der NVM-Vorrichtung verfügbaren Chiffre-Suites ausgewählt hat. Damit ist der Sicherheitsprotokoll-Handshake abgeschlossen. Dies sind einige wahrscheinliche Schritte zu einem solchen Sicherheitsprotokoll-Handshake; es können zusätzliche oder weniger Schritte durchgeführt werden, wie es für einen Fachmann auf dem Gebiet für Handshakes sicherer Protokolle offensichtlich ist.

[0051] **Fig. 6B** ist ein Flussdiagramm des sicheren Datentransfers 670, z. B. für FOTA, zwischen dem Server 105 und der NVM-Vorrichtung 110 gemäß einer Ausführungsform. Der sichere Datentransfer 670 kann damit beginnen, dass der Server 105 eine tcp_write(„Starte Firmware-Aktualisierung“) an das Host-Rechensystem 102 sendet (672). Das Host-Rechensystem 102 kann die tcp_to_spi()-AP! verwenden, um den Befehl in eine spi_write(„Starte Firmware-Aktualisierung“)-Mitteilung umzuwandeln, die an den SMC 122 gesendet wird. Der SMC 122 kann diese Mitteilung an den Mikrocontroller 118 übertragen. Der Mikrocontroller 118 kann dann bestimmen, ob die TLS-Metadaten (z. B. in dem TLS-Header) dieser Mitteilung gültig sind (674). Ist dies nicht der Fall, kann der Mikrocontroller 118 jeglichen sicheren Datentransfer abbrechen und zurückkehren zum Horchen auf Speicher-Lese- oder Schreibbefehle. Falls die TLS-Metadaten gültig sind, empfängt der Server 105 eine Bestätigung von der NVM-Vorrichtung 110 und kann mit dem Schreiben in das NVM 120 Zeile für Zeile beginnen.

[0052] Genauer gesagt wandelt die Host-Rechenvorrichtung 102 einen tcp_write(„Zeile 1“)-Befehl in einen SPI-Befehl um, der durch den SMC 122 weiter an den Mikrocontroller 118 gesendet wird (676). Der Mikrocontroller 118 kann dann den Krypto-Beschleuniger 140 auslösen, um die Chiffretext-Daten (z. B. der geparsten Daten) der Zeile 1 zu entschlüsseln (678) und den aus den entschlüsselten Daten erzeugten Hash-basierten Mitteilungs-Authentifizierungscode (Hash-Based Message Authentication Code, HMAC) zu verifizieren (680). Die Verifizierung des HMAC ist eine Implementierung der Verifizierung, andere Implementierungen, wie sie durch andere Chiffre-Suite-Codes verwendet werden, sind denkbar. Wenn der HMAC (oder ein

anderer Chiffre-Suite-Code) nicht verifiziert wird, kann der Mikrocontroller 118 den Datentransferprozess abbrechen (wie zuvor als Reaktion auf die nicht verifizierten TLS-Metadaten). Sobald der Mikrocontroller 118 von dem Krypto-Beschleuniger 140 eine Verifikationsbestätigung empfängt, kann der Mikrocontroller 118 die Zeile (z. B. Zeile 1) der Daten der Firmware-Aktualisierung in die Zieladresse in dem geparsten SPI-Paket schreiben (682). Der Mikrocontroller 118 kann ferner dem Server 105 über den SMC 122 und das Host-Rechensystem 102 melden, dass die Datenzeile (z. B. Zeile 1) erfolgreich geschrieben wurde (684). Der Server 105 kann dann bestimmen, ob eine zusätzliche Datenzeile in der Firmware-Aktualisierung vorhanden ist (688). Wenn ja, kann der Server 105 einen nächsten sicheren Schreibbefehl erzeugen, der wiederum (wie hierin erörtert wurde) in ein SPI-Paket umgewandelt wird, das an den SMC 122 und schließlich an den Mikrocontroller 118 gesendet wird, um die Blöcke 676 bis 684 für jede weitere Zeile zu wiederholen. Auf diese Weise behält der Server 105 die Kontrolle über das weitere Schreiben der Firmware-Aktualisierung in die NVM-Vorrichtung 110 und jede Datenzeile wird entschlüsselt und authentifiziert, bevor eine weitere Zeile geschrieben wird.

[0053] Sobald es keine weiteren Zeilen mehr gibt, kann der Server 105 Schritte unternehmen, um die Firmware-Aktualisierung zu beenden (690). Zum Beispiel kann der Server 105 einen tcp_write(„Beende Firmware-Aktualisierung“)-Befehl senden, der, sobald er seinen Weg zum Mikrocontroller 118 gefunden hat, den Mikrocontroller 118 veranlasst, den Krypto-Beschleuniger 140 auszulösen, um einen Digest-Hash einer neuen Abbildung zu erzeugen, z. B. mxcrypto_sha256(„Neue Abbildung“). Dieses Hash-Ergebnis kann zurück an den Mikrocontroller 118 und weiter an den Server durch den SMC 122 und die Host-Rechenvorrichtung 102 gesendet werden. Der Server 105 kann das Hash-Ergebnis mit einem zuvor gespeicherten Hash einer Abbildung der Firmware-Aktualisierung vergleichen und dadurch die erfolgreich installierte Firmware-Aktualisierung bestätigen. Die TCP-Verbindung mit der Host-Rechenvorrichtung 102, welche die Firmware-Aktualisierung mit der NVM-Vorrichtung 110 erleichterte, kann dann beendigt werden (Block 695 in **Fig. 6**).

[0054] Die hierin erörterten Verfahren werden zwar in erster Linie unter Bezugnahme auf die Durchführung einer Firmware-Aktualisierung an der NVM-Vorrichtung 110 erläutert, aber der Server 105 (oder eine andere entfernte Rechenvorrichtung) kann auch aus anderen Gründen, die zusätzliche Funktionalität beinhalten können, sicher mit der NVM-Vorrichtung 110 kommunizieren. Beispielsweise kann der Server 105, nachdem ein sicherer Handshake durchgeführt wurde, in der Lage sein, Zugriffssteuerungsbefehle, für die der Server 105 berechtigt ist, zu senden, als wäre der Server 105 das Host-Rechensystem 102. Diese Zugriffssteuerungsbefehle können zum Beispiel die Fähigkeit umfassen, Abschnitte der NVM 120 zu sperren oder zu entsperren, unterschiedliche Formen von Lese-, Ausführungs- und/oder Schreibsteuerungen für Abschnitte des NVM 120 oder für bestimmte in der NVM 120 gespeicherte Programme oder Firmware einzustellen.

[0055] Ferner kann der Server 105 ein Diagnoseprogramm in der NVM-Vorrichtung 110 aus der Ferne initiieren oder zumindest die durch die NVM-Vorrichtung 110 erzeugten Diagnoseinformationen aus der Ferne und sicher abrufen. Der Zugriff auf solche Diagnosedaten kann es dem Server 105 erlauben, schnell zu ermitteln, ob ein Aspekt der Hardware- und/oder Softwarefunktionalität angibt, dass die NVM-Vorrichtung 110 kompromittiert wurde, und daher eine Trennung der sicheren Netzwerksitzung, die zwischen dem Server 105 und der NVM-Vorrichtung 110 aufgebaut wurde, gerechtfertigt ist. Sobald die sichere Netzwerksitzung getrennt ist, werden die Verfahren der **Fig. 6**, **Fig. 6A** und **Fig. 6B** neu gestartet, um eine neue sichere Sitzung herzustellen. Dieses Verfahren kann aktualisiert werden, um zu prüfen, ob das Hardware- oder Softwareproblem, welches auch immer die Notwendigkeit der Trennung der sicheren Netzwerksitzung hervorgerufen haben mag, gelöst wurde, bevor vertrauliche Daten im sicheren Datentransfer ausgetauscht werden (**Fig. 6B**).

[0056] Fig. 7 veranschaulicht eine schematische Darstellung einer Maschine in der beispielhaften Form eines Rechensystems 700, in dem ein Satz von Anweisungen ausgeführt werden kann, um die Maschine zu veranlassen, irgendeine oder mehrere der hierin erörterten Methodiken durchzuführen. In alternativen Implementierungen kann die Maschine mit anderen Maschinen in einem LAN, einem Intranet, einem Extranet oder dem Internet verbunden (z. B. vernetzt) sein. Die Maschine kann in der Funktion eines Servers oder einer Client-Vorrichtung in einer Client-Server-Netzwerkumgebung oder als Peer-Maschine in einer Peer-to-Peer-Netzwerkumgebung (oder verteilten Netzwerkumgebung) arbeiten. Bei dieser Maschine kann es sich um ein Host-Rechensystem oder einen Host-Computer, eine Fahrzeug-Rechenvorrichtung, einen Server, eine Netzwerkvorrichtung für ein Automobilnetzwerk, wie ein Controller Area Network (CAN) oder ein Local Interconnected Network (LIN), oder um irgendeine Maschine, die in der Lage ist, einen Satz von Anweisungen (sequenziell oder anders) auszuführen, die von dieser Maschine zu unternehmenden Aktionen spezifizieren, handeln. Ferner wird zwar nur eine einzelne Maschine veranschaulicht, der Begriff „Maschine“ soll jedoch auch so verstanden werden, dass er jegliche Sammlung von Maschinen, die einzeln oder gemeinsam einen

Satz (oder eine Vielzahl von Sätzen) von Anweisungen zur Durchführung irgendeiner oder mehrerer der hierin besprochenen Methodiken ausführen, umfasst. Die Implementierungen der umwandelnden Seiten und Sektionen können in dem Rechensystem 700 implementiert werden.

[0057] Das Rechensystem 700 umfasst eine Verarbeitungsvorrichtung 702, einen Hauptspeicher 704 (z. B. Nur-Lese-Speicher (ROM), Flash-Speicher, dynamischen Direktzugriffsspeicher (DRAM) (wie einen synchronen DRAM (SDRAM) oder DRAM (RDRAM) usw.), einen statischen Speicher 706 (z. B. Flash-Speicher, statischen Direktzugriffsspeicher (SRAM) usw.) und eine Datenspeicherungsvorrichtung 718, die über einen Bus 730 miteinander kommunizieren.

[0058] Die Verarbeitungsvorrichtung 702 stellt eine oder mehrere Allzweck-Verarbeitungsvorrichtungen, wie etwa eine Mikroprozessorvorrichtung, eine Zentraleinheit oder dergleichen dar. Insbesondere kann die Verarbeitungsvorrichtung eine CISC(Complex Instruction Set Computing)-Mikroprozessorvorrichtung, eine RISC (Reduced Instruction Set Computer)-Mikroprozessorvorrichtung, eine VLIW(Very Long Instruction Word)-Mikroprozessorvorrichtung oder eine Verarbeitungsvorrichtung, die andere Anweisungssätze implementiert, oder Verarbeitungsvorrichtungen, die eine Kombination von Anweisungssätzen implementieren, sein. Die Verarbeitungsvorrichtung 702 kann auch eine Spezialzweck-Verarbeitungsvorrichtung sein oder es können mehrere Spezialzweck-Verarbeitungsvorrichtungen sein, wie etwa eine anwendungsspezifische integrierte Schaltung (ASIC, Application Specific Integrated Circuit), ein FPGA (Field Programmable Gate Array), eine Digitalsignalprozessorvorrichtung (DSP), eine Netzwerk-Verarbeitungsvorrichtung oder dergleichen. In einer Implementierung kann die Verarbeitungsvorrichtung 702 einen oder mehrere Verarbeitungsvorrichtungskerne umfassen. Die Verarbeitungsvorrichtung 702 ist konfiguriert, um die Anweisungen 726 zum Durchführen der hierin erörterten Operationen auszuführen. In einer Implementierung kann die Verarbeitungsvorrichtung 702 Teil des Servers 105, des Host-Rechensystems 102 oder der NVM-Vorrichtung 110 sein.

[0059] Alternativ kann das Rechensystem 700 andere Komponenten umfassen als hierin beschrieben umfassen. Das Rechensystem 700 kann ferner eine Netzwerkschnittstellenvorrichtung 708 umfassen, die kommunikativ mit einem Netzwerk 720 gekoppelt ist. Das Rechensystem 700 kann auch eine Videoanzeigeeinheit 710 (z. B. eine Flüssigkristallanzeige (LCD)), eine alphanumerische Eingabevorrichtung 712 (z. B. eine Tastatur), eine Cursorsteuervorrichtung 714 (z. B. eine Maus), eine Signalerzeugungsvorrichtung 716 (z. B. einen Lautsprecher) oder andere Peripherievorrichtungen umfassen. Darüber hinaus kann das Rechensystem 700 eine Graphikverarbeitungseinheit 722, eine Videoverarbeitungseinheit 728 und eine Audioverarbeitungseinheit 732 umfassen. In einer anderen Implementierung kann das Rechensystem 700 einen Chipsatz (nicht veranschaulicht) umfassen, der sich auf eine Gruppe von integrierten Schaltkreisen oder Chips bezieht, die ausgestaltet sind, um mit der Verarbeitungsvorrichtung 702 zu arbeiten, und die Kommunikationen zwischen der Verarbeitungsvorrichtung 702 und externen Vorrichtungen steuert. Zum Beispiel kann der Chipsatz ein Satz von Chips auf einer Hauptplatine sein, der die Verarbeitungsvorrichtung 702 mit sehr schnellen Vorrichtungen, wie den Hauptspeicher 704 und Graphikcontrollern, sowie die Verarbeitungsvorrichtung 702 mit langsameren Peripheriebussen von Peripherieeinrichtungen, wie USB-, PCI- oder ISA-Bussen verknüpft.

[0060] Die Datenspeicherungsvorrichtung 718 kann ein computerlesbares Speicherungsmedium 724 umfassen, auf dem die Anweisungen 726 gespeichert sind, die eine oder mehrere der hierin beschriebenen Methodiken verkörpern. Die Anweisungen 726 können auch ganz oder zumindest teilweise innerhalb des Hauptspeichers 704 als Anweisungen 726 und/oder innerhalb der Verarbeitungsvorrichtung 702 als Verarbeitungslogik während ihrer Ausführung durch das Rechensystem 700 liegen; der Hauptspeicher 704 und die Verarbeitungsvorrichtung 702 bilden ebenfalls computerlesbare Speicherungsmedien.

[0061] Das computerlesbare Speicherungsmedium 724 kann auch verwendet werden, um Anweisungen 726 unter Nutzung der Verarbeitungsvorrichtung 702, wie mit Bezug auf **Fig. 1A-1B** beschrieben, und/oder einer Software-Bibliothek, die Verfahren enthält, welche die obigen Anwendungen aufrufen, zu speichern. Zwar wird das computerlesbare Speicherungsmedium 724 in einer beispielhaften Implementierung als ein einzelnes Medium gezeigt, aber der Begriff „computerlesbares Speicherungsmedium“ sollte so verstanden werden, dass er ein einzelnes Medium oder eine Vielzahl von Medien (z. B. eine zentralisierte oder verteilte Datenbank, und/oder zugehörige Zwischenspeicher und Server) umfasst, die den einen oder die mehreren Sätze von Anweisungen speichern. Unter dem Begriff „computerlesbares Speicherungsmedium“ ist auch zu verstehen, dass er jegliches Medium umfasst, das in der Lage ist, einen Satz von Anweisungen zur Ausführung durch die Maschine zu speichern, zu kodieren oder zu führen, welche die Maschine veranlassen, irgendeine oder mehrere der Methodiken der Implementierungen durchzuführen. Der Begriff „computerlesbares Spei-

cherungsmedium“ soll dementsprechend so verstanden werden, jedoch nicht darauf beschränkt sein, dass er Festkörperspeicher und optische und magnetische Medien umfasst.

[0062] In der obigen Beschreibung sind zahlreiche Details dargelegt. Es wird sich jedoch für einen Durchschnittsfachmann, der den Nutzen dieser Offenbarung hat, verstehen, dass Ausführungsformen der vorliegenden Offenbarung auch ohne diese spezifischen Details praktiziert werden können. In einigen Fällen werden bekannte Strukturen und Vorrichtungen nicht im Detail, sondern in Blockdiagrammform gezeigt, um eine Verschleierung der Beschreibung zu vermeiden.

[0063] Ein Modul, wie es hierin verwendet wird, bezieht sich auf irgendeine Kombination von Hardware, Software und/oder Firmware. Beispielsweise umfasst ein Modul Hardware, wie einen Mikrocontroller, die mit einem nichttransitorischen Medium assoziiert ist, um Code zu speichern, der für die Ausführung durch den Mikrocontroller angepasst ist. Daher bezieht sich der Verweis auf ein Modul in einer Implementierung auf die Hardware, die speziell konfiguriert ist, um den Code, der auf einem nichttransitorischen Medium vorhanden ist, zu erkennen und/oder auszuführen. Darüber hinaus bezieht sich in einer weiteren Implementierung die Verwendung eines Moduls auf das nichttransitorische Medium einschließlich des Codes, der speziell angepasst ist, um durch den Mikrocontroller ausgeführt zu werden, um vorgegebene Operationen durchzuführen. Und wie gefolgert werden kann, kann sich in einer noch weiteren Implementierung der Begriff Modul (in diesem Beispiel) auf die Kombination des Mikrocontrollers und des nichttransitorischen Mediums beziehen. Häufig variieren und überschneiden sich Modulgrenzen, die als getrennt dargestellt werden. Zum Beispiel können ein erstes und ein zweites Modul Hardware, Software, Firmware oder eine Kombination davon gemeinsam nutzen, während möglicherweise eine unabhängige Hardware, Software oder Firmware beibehalten wird. In einer Implementierung umfasst die Verwendung des Begriffs Logik auch Hardware, wie Transistoren, Register oder andere Hardware, wie programmierbare Logikbausteine.

[0064] Die Verwendung des Ausdrucks „konfiguriert, um“ in einer Implementierung bezieht sich auf das Einrichten, Zusammenstellen, Herstellen, Verkaufsangebot, Importieren und/oder Ausgestalten eines Geräts, einer Hardware, einer Logik oder eines Elements zum Durchführen einer speziell dafür vorgesehenen oder bestimmten Aufgabe. In diesem Beispiel ist ein Gerät oder ein Element davon, das nicht in Betrieb ist, dennoch „konfiguriert, um“ eine speziell vorgesehene Aufgabe durchzuführen, falls sie bzw. es ausgestaltet, gekoppelt und/oder verschaltet ist, die speziell vorgesehene Aufgabe durchzuführen. Als rein veranschaulichendes Beispiel kann ein Logikgatter während des Betriebs eine 0 oder eine 1 bereitstellen. Aber ein Logikgatter, das „konfiguriert ist, um“ ein Freigabesignal für einen Takt bereitzustellen, umfasst nicht jedes potenzielle Logikgatter, das eine 1 oder 0 bereitstellen kann. Stattdessen ist das Logikgatter ein solches, das auf eine Art und Weise gekoppelt ist, dass während des Betriebs der Ausgang 1 oder 0 den Takt aktivieren soll. Es sei nochmals darauf hingewiesen, dass die Verwendung des Begriffs „konfiguriert, um" keine Operation erfordert, sondern sich stattdessen auf den latenten Zustand eines Geräts, einer Hardware und/oder eines Elements abzielt, wobei das Gerät, die Hardware und/oder das Element in dem latenten Zustand ausgestaltet ist, um eine bestimmte Aufgabe durchzuführen, wenn das Gerät, die Hardware und/oder das Element in Betrieb ist.

[0065] Darüber hinaus bezieht sich die Verwendung des Ausdrucks „um, ... zu“, „in der Lage zu“ und oder „betriebsfähig zu“ in einer Implementierung auf irgendein Gerät, irgendeine Logik, irgendeine Hardware und/oder irgendein Element, die bzw. das so ausgelegt ist, dass die Verwendung des Geräts, der Logik, der Hardware und/oder des Elements in einer spezifizierten Weise ermöglicht wird. Es ist wie oben zu beachten, dass sich die Verwendung von „um, ... zu“, „in der Lage zu“ oder „betriebsfähig zu“ in einer Implementierung auf den latenten Zustand eines Geräts, einer Logik, einer Hardware und/oder eines Elements bezieht, wobei das Gerät, die Logik, die Hardware und/oder das Element nicht in Betrieb ist, sondern so ausgelegt ist, um die Verwendung eines Geräts in einer spezifizierten Weise zu ermöglichen.

[0066] Ein Wert, wie hierin verwendet, umfasst irgendeine bekannte Darstellung einer Zahl, eines Zustands, eines logischen Zustands oder eines binären logischen Zustands. Häufig wird die Verwendung von Logikebenen, Logikwerten oder logischen Werten auch als 1en und 0en bezeichnet, was lediglich binäre Logikzustände darstellt. Zum Beispiel bezieht sich eine 1 auf eine hohe Logikebene und 0 bezieht sich auf eine niedrige Logikebene. In einer Ausführungsform kann eine Speicherungszelle, wie ein Transistor oder eine Flashzelle, in der Lage sein, einen einzigen logischen Wert oder mehrere logische Werte zu halten. Es wurden jedoch auch andere Darstellungen von Werten in Computersystemen verwendet. Zum Beispiel kann die Dezimalzahl Zehn auch als ein Binärwert von 1010 und als ein hexadezimaler Buchstabe A dargestellt werden. Daher umfasst ein Wert jegliche Darstellung von Informationen, die in einem Computersystem gehalten werden können.

[0067] Einige Abschnitte der detaillierten Beschreibung werden in Form von Algorithmen und symbolischen Darstellungen von Operationen auf Datenbits innerhalb eines Computerspeichers dargestellt. Diese algorithmischen Beschreibungen und Darstellungen sind die Mittel, die von einem Fachmann auf dem Datenverarbeitungsgebiet verwendet werden, um einem anderen Fachmann den maßgeblichen Inhalt seiner Arbeit in möglichst effektiver Weise zu vermitteln. Ein Algorithmus wird hier und im Allgemeinen als eine folgerichtige Sequenz von Schritten, die zu einem gewünschten Ergebnis führt, verstanden. Diese Schritte erfordern physikalische Manipulationen physikalischer Größen. Üblicherweise, aber nicht notwendigerweise, nehmen diese Größen die Form von elektrischen oder magnetischen Signalen an, die gespeichert, übertragen, kombiniert, verglichen und in anderer Weise manipuliert werden können. Es hat sich zuweilen als günstig herausgestellt, hauptsächlich aus Gründen der üblichen Verwendung, diese Signale als Bits, Werte, Elemente, Symbole, Zeichen, Begriffe, Zahlen oder dergleichen zu bezeichnen.

[0068] Es sollte jedoch berücksichtigt werden, dass all diese und ähnliche Begriffe mit den geeigneten physikalischen Größen zu assoziieren sind und lediglich praktische Bezeichnungen sind, die auf diese Größen angewendet werden. Sofern nicht aus den obigen Erörterungen ausdrücklich etwas anderes angegeben ist, versteht es sich, dass in der gesamten Beschreibung Erörterungen, die Begriffe wie „empfangen“, „einstellen“ oder dergleichen benutzen, sich auf die Aktionen und/oder Prozesse eines Rechensystems oder einer ähnlichen elektronischen Rechenvorrichtung beziehen, welche Daten, die als physikalische (z. B. elektronische) Größen innerhalb der Register und Speicher des Rechensystems dargestellt werden, manipulieren und umwandeln in andere Daten, die in ähnlicher Weise als physikalische Größen innerhalb der Speicher oder Register oder anderen derartigen Informationsspeicherungs-, Übertragungs- oder Anzeigevorrichtungen des Rechensystems dargestellt werden.

[0069] Die Worte „Beispiel“ oder „beispielhaft“ werden hierin verwendet, um als Beispiel, Fall oder Veranschaulichung zu dienen. Jeglicher Aspekt oder jegliche Ausgestaltung, der oder die hierin als „Beispiel” oder „beispielhaft“ beschrieben wird, ist nicht unbedingt als bevorzugt oder vorteilhaft gegenüber anderen Aspekten oder Ausgestaltungen auszulegen. Vielmehr soll die Verwendung der Worte „Beispiel“ oder „beispielhaft“ dazu dienen, Konzepte in einer konkreten Art zu präsentieren. Der in dieser Anmeldung verwendete Begriff „oder“ soll eher ein einschließendes „oder“ als ein ausschließliches „oder“ sein Das heißt, sofern nicht anders angegeben oder aus dem Kontext klar ersichtlich, mit „X umfasst A oder B“ ist beabsichtigt, dass es jegliche beliebige der natürlichen inklusiven Permutationen bedeuten kann. Das heißt, falls X A umfasst; X B umfasst oder X sowohl A als auch B umfasst, dann ist „X umfasst A oder B“ unter jedem der vorangehenden Fälle erfüllt. Darüber hinaus sind die Artikel „ein“, „einer“ und „eines“, wie in dieser Anmeldung und den anhängenden Ansprüchen verwendet, allgemein so auszulegen, dass sie „eines oder mehr“ bedeuten, sofern nicht anders angegeben oder aus dem Kontext ersichtlich wird, dass auf eine Singularform verwiesen wird. Außerdem ist mit der Verwendung des Begriffs „eine Ausführungsform“ oder „eine einzelne Ausführungsform“ oder „eine Ausführungsform“ oder „eine einzelne Ausführungsform“ durchgehend nicht beabsichtigt, dass damit die gleiche Ausführungsform oder Ausführungsform gemeint sein soll, es sei denn, es wird so beschrieben.

[0070] Hierin beschriebene Ausführungsformen können sich auch auf ein Gerät zum Durchführen der hierin enthaltenen Operationen beziehen. Dieses Gerät kann speziell für die erforderlichen Zwecke aufgebaut sein oder es kann Allzweck-Hardware beinhalten, die durch eine darin gespeicherte Firmware selektiv aktiviert oder rekonfiguriert wird. Eine solche Firmware kann in einem nichttransitorischen computerlesbaren Speicherungsmedium gespeichert werden, wie etwa, jedoch nicht beschränkt auf, NVMs, Nur-Lese-Speicher (ROMs), Direktzugriffsspeicher (RAMs), EPROMs, EEPROMs, Flash-Speicher oder irgendeine Art von Medien, die zum Speichern elektronischer Anweisungen geeignet sind. Unter dem Begriff „computerlesbares Speichermedium“ ist zu verstehen, dass er ein einzelnes Medium oder mehrere Medien umfasst, die einen oder mehrere Sätze von Anweisungen speichern. Unter dem Begriff „computerlesbares Medium“ ist auch zu verstehen, dass er jegliches Medium umfasst, das in der Lage ist, einen Satz von Anweisungen zur Ausführung durch die Hardware zu speichern, zu kodieren oder zu führen, welche die Hardware veranlassen, irgendeine oder mehrere der Methodiken der vorliegenden Ausführungsform durchzuführen. Unter dem Begriff „computerlesbares Speicherungsmedium“ ist demgemäß zu verstehen, dass er unter anderem Festkörperspeicher, optische Medien, elektromagnetische Medien, irgendein Medium umfasst, das in der Lage ist, einen Satz von Anweisungen zur Ausführung durch Hardware zu speichern, und das die Hardware veranlasst, irgendeine oder mehrere der Methodiken der vorliegenden Ausführungsformen durchzuführen.

[0071] Die obige Beschreibung legt zahlreiche spezifische Details, wie etwa Beispiele für spezifische Systeme, Komponenten, Verfahren und so weiter dar, so dass mehrere Ausführungsformen der vorliegenden Offenbarung gut verstanden werden können. Für den Fachmann wird es jedoch ersichtlich sein, dass mindestens einige Ausführungsformen der vorliegenden Offenbarung ohne diese spezifischen Details praktiziert

werden können. In anderen Fällen werden bekannte Komponenten oder Verfahren nicht im Detail beschrieben oder werden in einem einfachen Blockdiagrammformat präsentiert, um eine unnötige Verschleierung der vorliegenden Offenbarung zu vermeiden. Somit sind die oben dargelegten spezifischen Details lediglich beispielhaft. Bestimmte Ausführungsformen können von diesen beispielhaften Details abweichen und dennoch als innerhalb des Schutzbereichs der vorliegenden Offenbarung liegend erachtet werden.

[0072] Es versteht sich, dass die obige Beschreibung veranschaulichend und nicht einschränkend sein soll. Viele andere Ausführungsformen werden dem Fachmann beim Lesen und Verstehen der obigen Beschreibung ersichtlich sein. Der Schutzbereich der Offenbarung sollte deshalb unter Bezugnahme auf die anhängenden Ansprüche zusammen mit dem vollen Schutzbereich von Äquivalenten, auf die solche Ansprüche Anrecht haben, bestimmt werden.

[0073] In der obigen Beschreibung werden zu Erläuterungszwecken zahlreiche spezifische Details dargelegt, um ein vertieftes Verständnis der vorliegenden Offenbarung bereitzustellen. Für den Fachmann ist es jedoch offensichtlich, dass die vorliegenden Offenbarung auch ohne diese spezifischen Details ausgeführt werden können. In anderen Fällen werden bekannte Schaltungen, Strukturen und Techniken nicht im Einzelnen, sondern vielmehr in einem Blockdiagramm gezeigt, um eine unnötige Verschleierung des Verständnisses dieser Beschreibung zu vermeiden.

[0074] Wenn in der Beschreibung „eine Ausführungsform“ genannt wird, bedeutet dies, dass ein bestimmtes Merkmal, eine bestimmte Struktur oder Charakteristik, welches oder welche in Verbindung mit der Ausführungsform beschrieben wird/werden, in mindestens einer Ausführungsform der Offenbarung umfasst ist. Der Ausdruck „in einer Ausführungsform“, der an verschiedenen Stellen in dieser Beschreibung zu finden ist, bezieht sich nicht notwendigerweise auf dieselbe Ausführungsform.

Patentansprüche

1. Ein Gerät, das Folgendes beinhaltet:
einen Speicher-Controller;
einen statischen Direktzugriffsspeicher (SRAM), der an den Speicher-Controller gekoppelt ist, wobei der SRAM für ein Host-Rechensystem, das kommunikativ an einen Server gekoppelt ist, unzugänglich ist; und
eine an das Host-Rechensystem gekoppelte nichtflüchtige Speichervorrichtung (NVM-Vorrichtung), wobei die NVM-Vorrichtung eine Verarbeitungsvorrichtung umfasst zum:
Empfangen eines Kommunikationspakets von dem Server über das Host-Rechensystem, wobei das Kommunikationspaket eine Anforderung zum Einleiten sicherer Kommunikationen beinhaltet;
Durchführen eines sicheren Handshakes mit dem Server, über eine Kommunikation durch das Host-Rechensystem, unter Verwendung eines Sicherheitsprotokolls;
Empfangen von Daten, über das Host-Rechensystem, von dem Server innerhalb eines Sicherheitsprotokollpakets;
Speichern, als Reaktion auf das Detektieren eines Krypto-Schreib-Befehls in dem Sicherheitsprotokollpaket, des Sicherheitsprotokollpakets in einen Krypto-Puffer des SRAM;
Parsen des in dem Krypto-Puffer gespeicherten Sicherheitsprotokollpakets, um die Daten abzurufen;
Abrufen einer Sicherheitsprotokolloperations-Kennung und von Sicherheitsprotokoll-Metadaten aus einem Header des Sicherheitsprotokollpakets;
Transferieren von Abschnitten des Sicherheitsprotokollpakets von dem Krypto-Puffer zu dem SRAM; und
Verarbeiten der Abschnitte des Sicherheitsprotokollpakets aus dem SRAM heraus gemäß dem Sicherheitsprotokoll, um die Verifizierung von aus dem Sicherheitsprotokollpaket abgerufenen Sicherheitsprotokoll-Metadaten einzuschließen.

2. Gerät gemäß Anspruch 1, wobei das Sicherheitsprotokoll eines von dem SSL(Secure Sockets Layer)-Protokoll oder dem TLS(Transport Layer Security)-Protokoll beinhaltet und wobei der sichere Handshake eine Reihe von Sequenzierungsoperationen umfasst, die zu einer Reihe von kryptographischen Operationen führen.

3. Gerät gemäß Anspruch 1, wobei die NVM-Vorrichtung eine Flash-Speichervorrichtung ist, und wobei:
der Speicher-Controller einen SPI(Serial Peripheral Interface)-Slave umfasst, der an einen SPI-Master des Host-Rechensystems gekoppelt ist, wobei der sichere Handshake über die Übertragung von Daten innerhalb von SPI-Paketen, die zwischen dem SPI-Slave und dem SPI-Master ausgetauscht werden, durchgeführt wird; und
ein kryptographischer Beschleuniger in der NVM-Vorrichtung konfiguriert ist, um über die Ausführung einer

kryptographischen Werkzeugsammlung, die in den kryptographischen Beschleuniger programmiert ist, kryptographische Operationen durchzuführen.

4. Gerät gemäß Anspruch 1, wobei die Verarbeitungsvorrichtung ferner Folgendes vornehmen soll:
Authentifizieren der Daten unter Verwendung mindestens der Sicherheitsprotokoll-Metadaten, die aus dem Sicherheitsprotokollpaket abgerufen werden; und Speichern der Daten in NVM-Speicherungselementen der NVM-Vorrichtung.

5. Gerät gemäß Anspruch 4, wobei die Verarbeitungsvorrichtung den Zugriff, durch das Host-Rechensystem, auf die in den NVM-Speicherungselementen gespeicherten Daten bereitstellen soll.

6. Gerät gemäß Anspruch 3, wobei zum Verarbeiten der Abschnitte des Sicherheitsprotokollpakets gemäß dem Sicherheitsprotokoll die Verarbeitungsvorrichtung ferner mit dem kryptographischen Beschleuniger interagieren soll, um Folgendes vorzunehmen:
Authentifizieren einer Zeile der Daten gemäß einem Chiffre-Suite-Code;
Entschlüsseln der Zeile der Daten, falls verschlüsselt, wodurch eine Zeile Klartext-Daten erzeugt wird, über die Verwendung eines Sitzungsschlüssels;
Speichern der Zeile Klartext-Daten in NVM-Speicherungselementen der NVM-Vorrichtung; und
Zurückmelden, über den Speicher-Controller und das Host-Rechensystem an den Server, dass die Zeile der Daten erfolgreich in die NVM-Speicherungselemente geschrieben wurde.

7. Ein Verfahren, das Folgendes beinhaltet:
Empfangen, durch eine Verarbeitungsvorrichtung einer nichtflüchtigen Speichervorrichtung (NVM-Vorrichtung), eines Kommunikationspakets von einem Server über ein Host-Rechensystem, das an die NVM-Vorrichtung gekoppelt ist und kommunikativ an den Server gekoppelt ist, wobei das Kommunikationspaket eine Anforderung zum Einleiten sicherer Kommunikationen beinhaltet;
Durchführen, durch die Verarbeitungsvorrichtung, eines sicheren Handshakes mit dem Server über eine Kommunikation durch das Host-Rechensystem unter Verwendung eines Sicherheitsprotokolls;
Empfangen, unter Verwendung der Verarbeitungsvorrichtung, von verschlüsselten Daten über das Host-Rechensystem von dem Server innerhalb eines Sicherheitsprotokollpakets;
Speichern, durch die Verarbeitungsvorrichtung als Reaktion auf das Detektieren eines Krypto-Schreib-Befehls innerhalb des Sicherheitsprotokollpakets, des Sicherheitsprotokollpakets in einen Krypto-Puffer des statischen Direktzugriffsspeichers (SRAM) der NVM-Vorrichtung;
Parsen, durch die Verarbeitungsvorrichtung, des in dem Krypto-Puffer gespeicherten Sicherheitsprotokollpakets, um die verschlüsselten Daten abzurufen;
Abrufen, durch die Verarbeitungsvorrichtung, einer Sicherheitsprotokolloperations-Kennung und von Sicherheitsprotokoll-Metadaten aus einem Header aus dem Sicherheitsprotokollpaket;
Transferieren, durch die Verarbeitungsvorrichtung, von Abschnitten des Sicherheitsprotokollpakets von dem Krypto-Puffer zu dem SRAM; und
Verarbeiten, durch die Verarbeitungsvorrichtung, der Abschnitte des Sicherheitsprotokollpakets aus dem SRAM heraus gemäß dem Sicherheitsprotokoll, um die Verifizierung der Sicherheitsprotokoll-Metadaten einzuschließen.

8. Verfahren gemäß Anspruch 7, wobei das Durchführen des sicheren Handshakes das Austauschen von Sicherheitsprotokolldaten innerhalb von SPI(Serial Peripheral Interface)-Paketen mit dem Host-Rechensystem beinhaltet.

9. Verfahren gemäß Anspruch 7, wobei das Sicherheitsprotokoll eines von dem SSL(Secure Sockets Layer)-Protokoll oder dem TLS(Transport Layer Security)-Protokoll beinhaltet und wobei der sichere Handshake eine Reihe von Sequenzierungsoperationen umfasst, die zu einer Reihe von kryptographischen Operationen führen.

10. Verfahren gemäß Anspruch 7, wobei das Durchführen des sicheren Handshakes mit dem Server die Erzeugung eines Sitzungsschlüssels eines Paares von Sitzungsschlüsseln beinhaltet, wobei der Sitzungsschlüssel für das Host-Rechensystem unzugänglich ist, wobei das Verfahren ferner Folgendes beinhaltet:
Entschlüsseln, durch die Verarbeitungsvorrichtung unter Verwendung des Sitzungsschlüssels, der verschlüsselten Daten, um Klartext-Daten zu erzeugen; und Speichern, durch die Verarbeitungsvorrichtung, der Klartext-Daten in NVM-Speicherungselementen der NVM-Vorrichtung.

11. Verfahren gemäß Anspruch 10, das ferner das Bereitstellen des Zugriffs, durch das Host-Rechensystem, auf die in den NVM-Speicherungselementen gespeicherten Klartext-Daten beinhaltet.

12. Verfahren gemäß Anspruch 7, das ferner das Authentifizieren der verschlüsselten Daten unter Verwendung mindestens der Sicherheitsprotokoll-Metadaten, die aus dem Sicherheitsprotokollpaket abgerufen werden, beinhaltet.

13. Verfahren gemäß Anspruch 10, wobei das Verarbeiten ferner Folgendes beinhaltet:
Authentifizieren, durch einen kryptographischen Beschleuniger der NVM-Vorrichtung, einer Zeile der verschlüsselten Daten des Sicherheitsprotokollpakets gemäß einem Chiffre-Suite-Code;
Entschlüsseln, durch den kryptographischen Beschleuniger unter Verwendung des Sitzungsschlüssels, der Zeile der verschlüsselten Daten, wobei eine Zeile Klartext-Daten erzeugt wird;
Speichern der Zeile Klartext-Daten in den NVM-Speicherungselementen; und Zurückmelden, über das Host-Rechensystem an den Server, dass die Zeile der verschlüsselten Daten erfolgreich in die NVM-Speicherungselemente geschrieben wurde.

Es folgen 10 Seiten Zeichnungen

Anhängende Zeichnungen

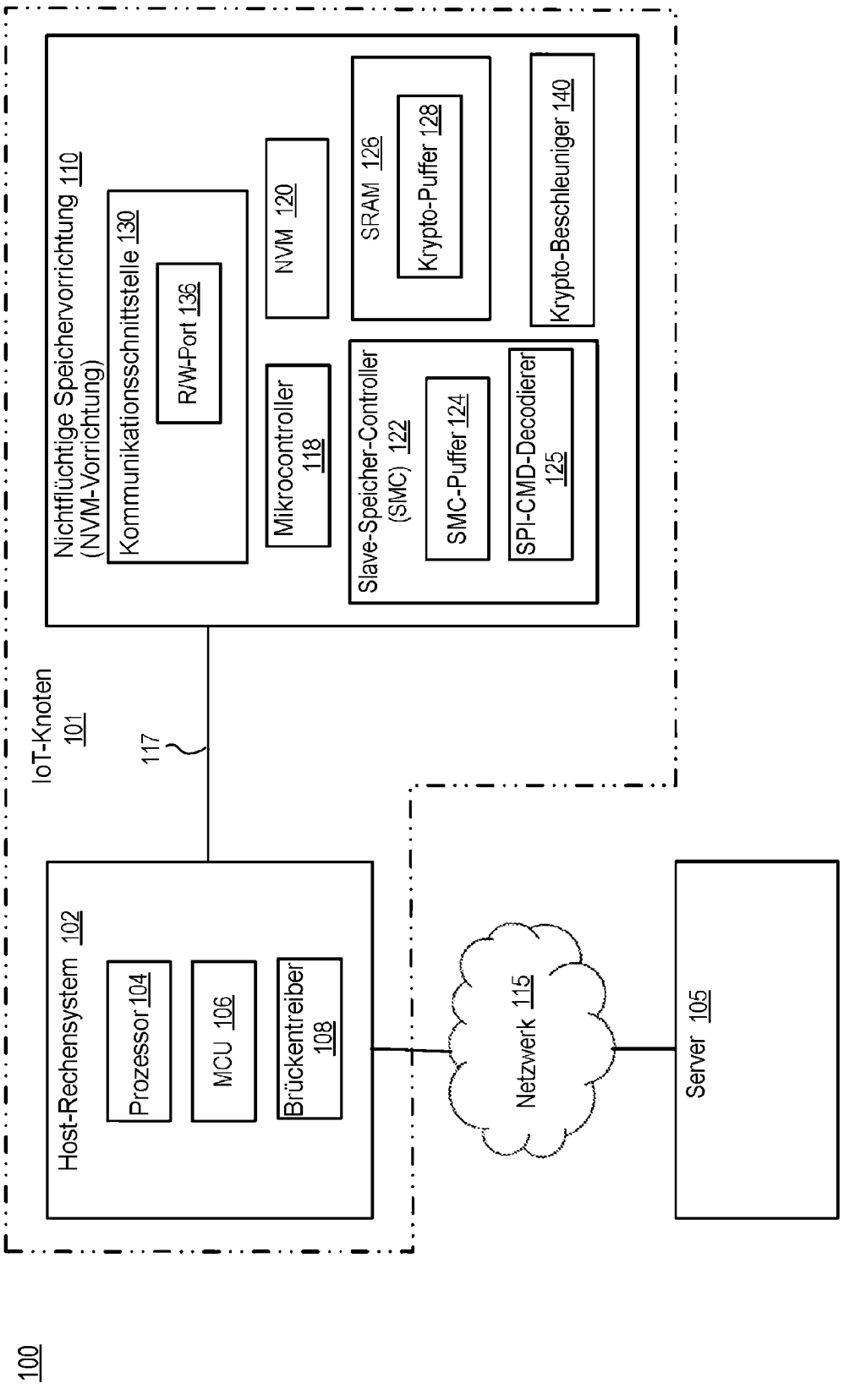


FIG.1A

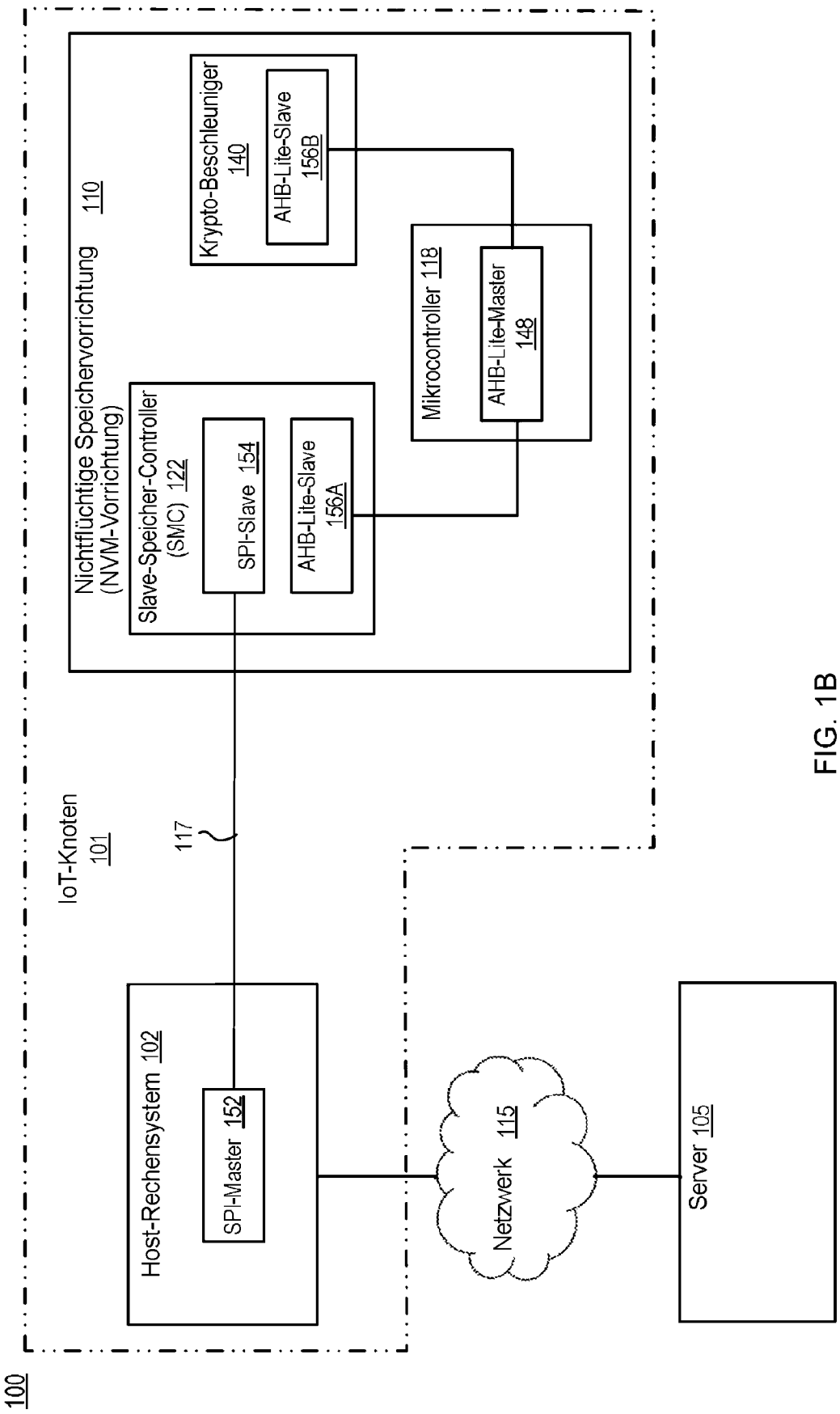

100
IoT-Knoten
101
Nichtflüchtige Speichervorrichtung (NVM-Vorrichtung) 110
Slave-Speicher-Controller (SMC) 122
SPI-Slave 154
AHB-Lite-Slave 156A
Krypto-Beschleuniger 140
AHB-Lite-Slave 156B
Mikrocontroller 118
AHB-Lite-Master 148
117
Host-Rechensystem 102
SPI-Master 152
Netzwerk 115
Server 105

FIG. 1B

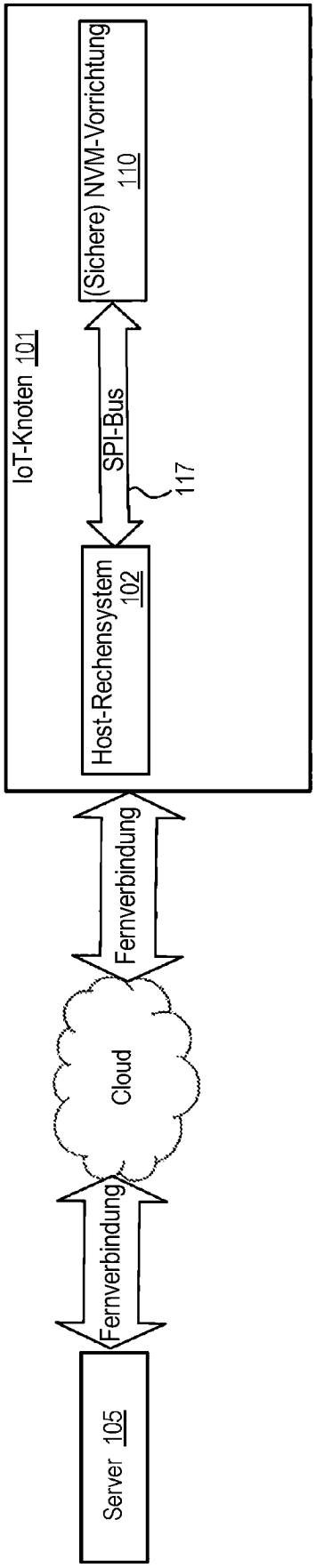

IoT-Knoten 101
(Sichere) NVM-Vorrichtung 110
SPI-Bus
117
Host-Rechensystem 102
Fernverbindung
Cloud
Fernverbindung
Server 105

FIG. 2A

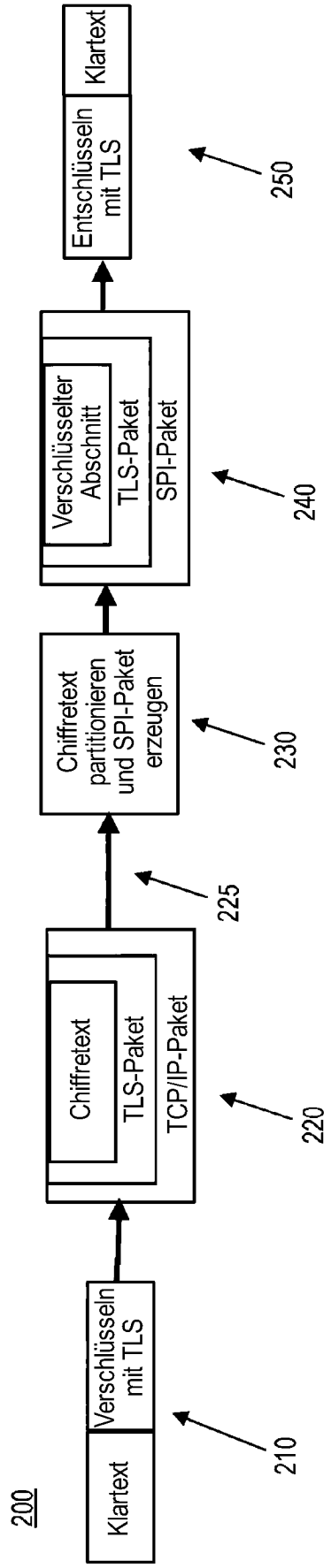

200
Klartext
Verschlüsseln mit TLS
210
Chiffretext
TLS-Paket
TCP/IP-Paket
220
225
Chiffretext partitionieren und SPI-Paket erzeugen
230
Verschlüsselter Abschnitt
TLS-Paket
SPI-Paket
240
Entschlüsseln mit TLS
Klartext
250

FIG. 2B

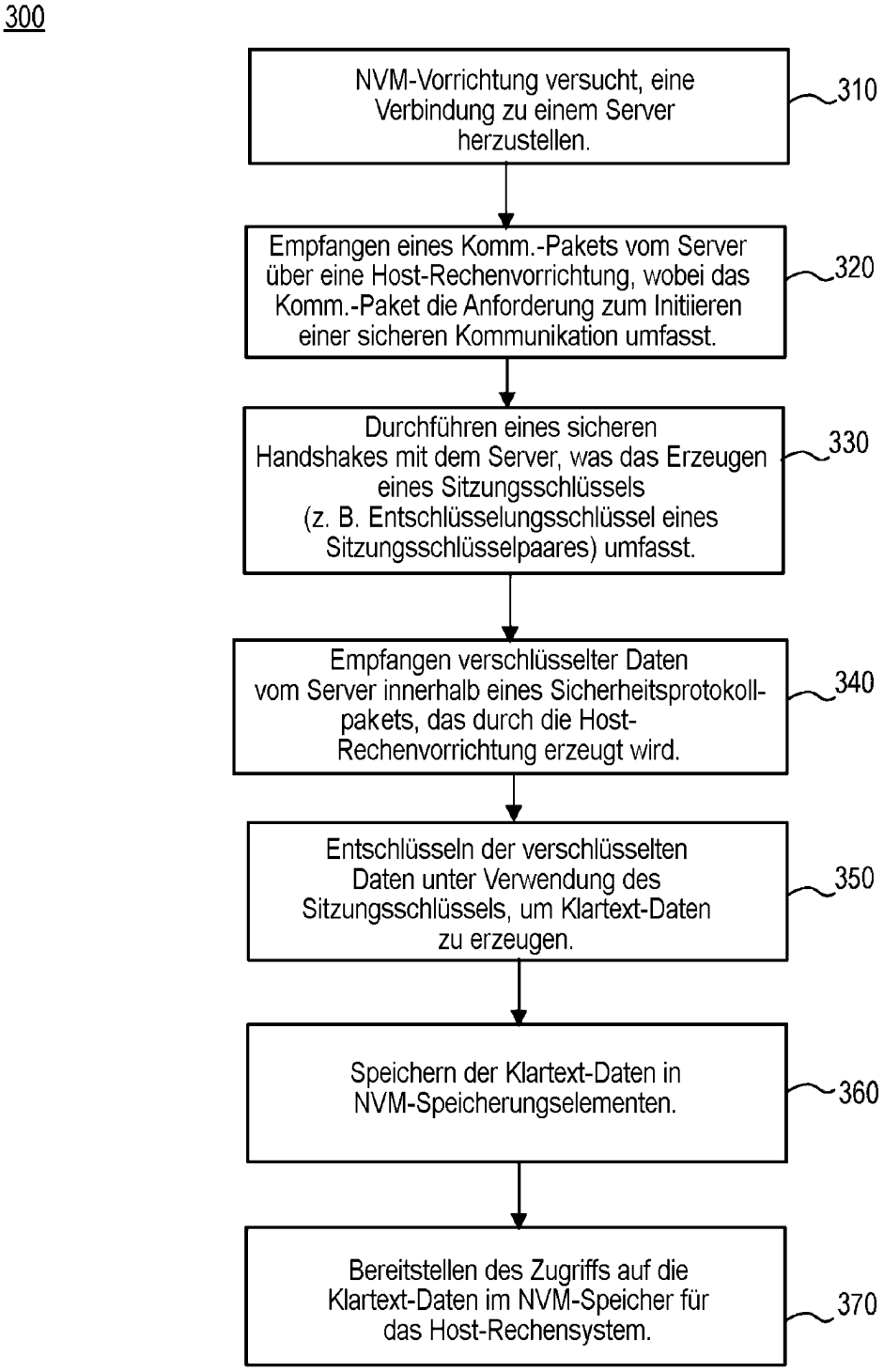

300
NVM-Vorrichtung versucht, eine Verbindung zu einem Server herzustellen.
310
Empfangen eines Komm.-Pakets vom Server über eine Host-Rechenvorrichtung, wobei das Komm.-Paket die Anforderung zum Initiieren einer sicheren Kommunikation umfasst.
320
Durchführen eines sicheren Handshakes mit dem Server, was das Erzeugen eines Sitzungsschlüssels (z. B. Entschlüsselungsschlüssel eines Sitzungsschlüsselpaares) umfasst.
330
Empfangen verschlüsselter Daten vom Server innerhalb eines Sicherheitsprotokoll-pakets, das durch die Host-Rechenvorrichtung erzeugt wird.
340
Entschlüsseln der verschlüsselten Daten unter Verwendung des Sitzungsschlüssels, um Klartext-Daten zu erzeugen.
350
Speichern der Klartext-Daten in NVM-Speicherungselementen.
360
Bereitstellen des Zugriffs auf die Klartext-Daten im NVM-Speicher für das Host-Rechensystem.
370

FIG. 3

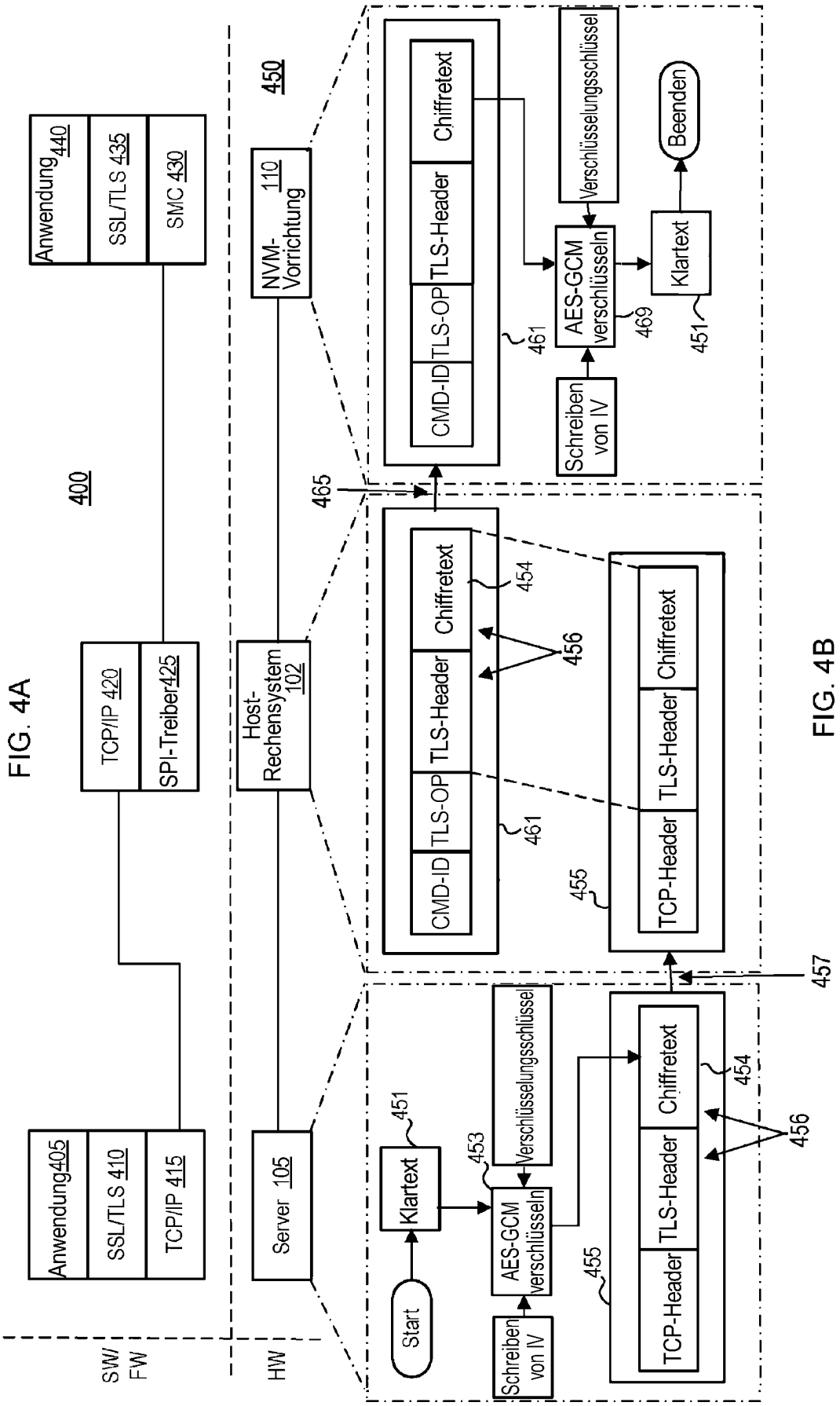

Anwendung 440
SSL/TLS 435
SMC 430
400
TCP/IP 420
SPI-Treiber 425
Anwendung 405
SSL/TLS 410
TCP/IP 415
SW/FW
HW
Host-Rechensystem 102
Server 105
450
NVM-Vorrichtung 110
465
CMD-ID
TLS-OP
TLS-Header
Chiffretext
461
454
456
455
457
Klartext
451
453
Verschlüsselungsschlüssel
AES-GCM verschlüsseln
469
Schreiben von IV
Start
Beenden
TCP-Header

FIG. 4A

FIG. 4B

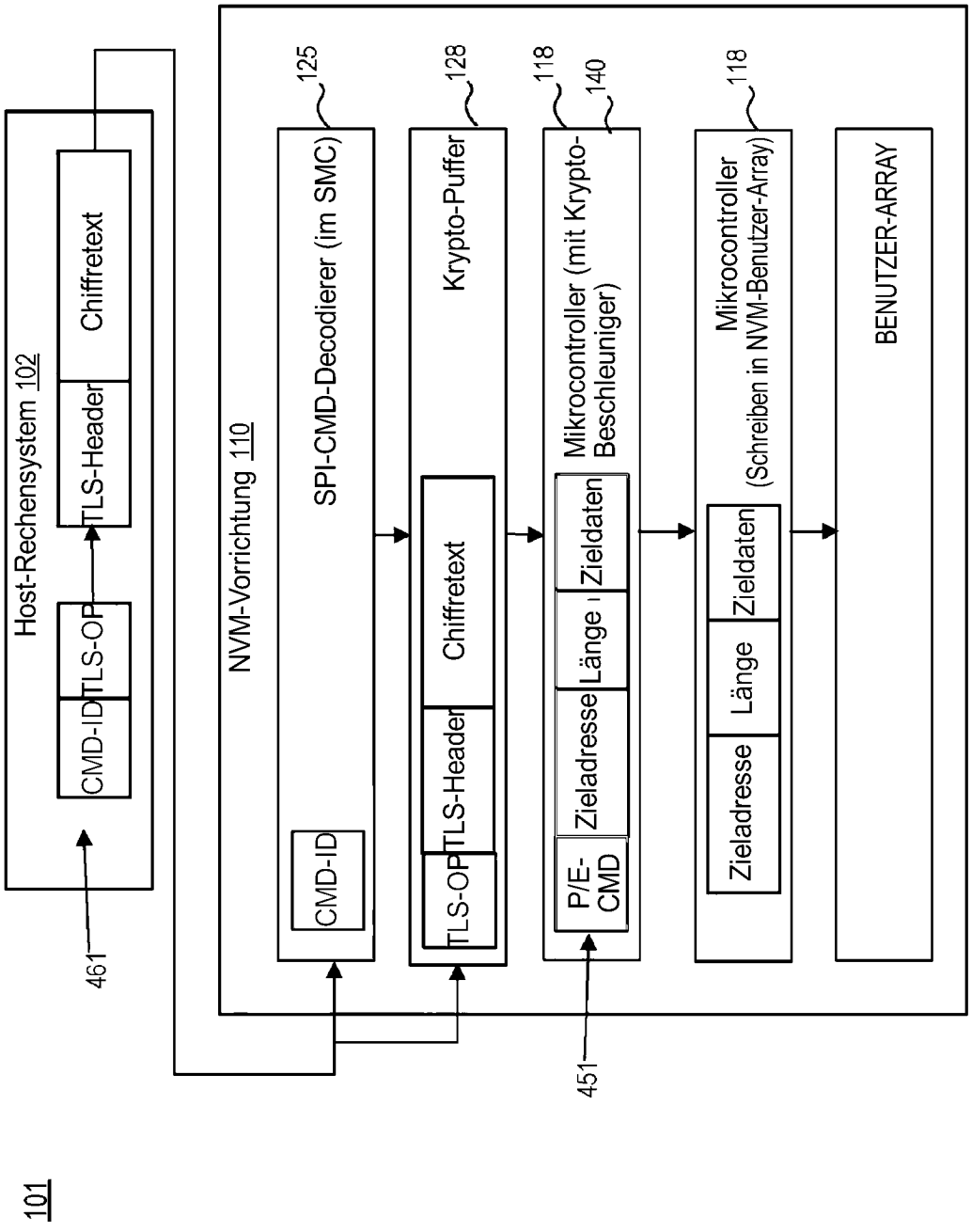

101
Host-Rechensystem 102
CMD-ID
TLS-OP
TLS-Header
Chiffretext
461
NVM-Vorrichtung 110
125
SPI-CMD-Decodierer (im SMC)
CMD-ID
128
Krypto-Puffer
TLS-OP
TLS-Header
Chiffretext
118
140
Mikrocontroller (mit Krypto-Beschleuniger)
P/E-CMD
Zieladresse
Länge
Zieldaten
451
118
Mikrocontroller (Schreiben in NVM-Benutzer-Array)
Zieladresse
Länge
Zieldaten
BENUTZER-ARRAY

FIG. 5

600

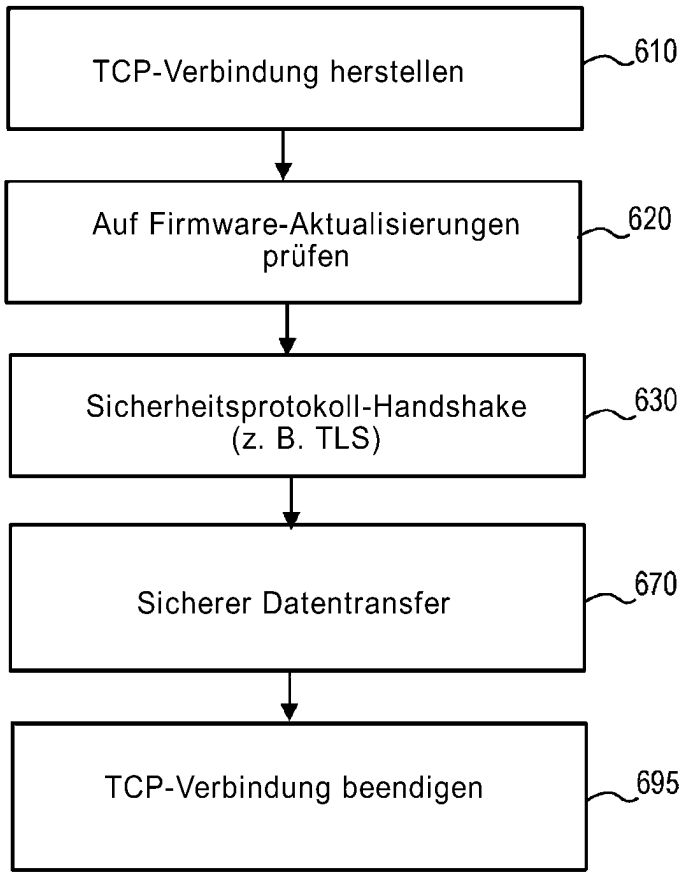

TCP-Verbindung herstellen
610
Auf Firmware-Aktualisierungen prüfen
620
Sicherheitsprotokoll-Handshake (z. B. TLS)
630
Sicherer Datentransfer
670
TCP-Verbindung beendigen
695

FIG. 6

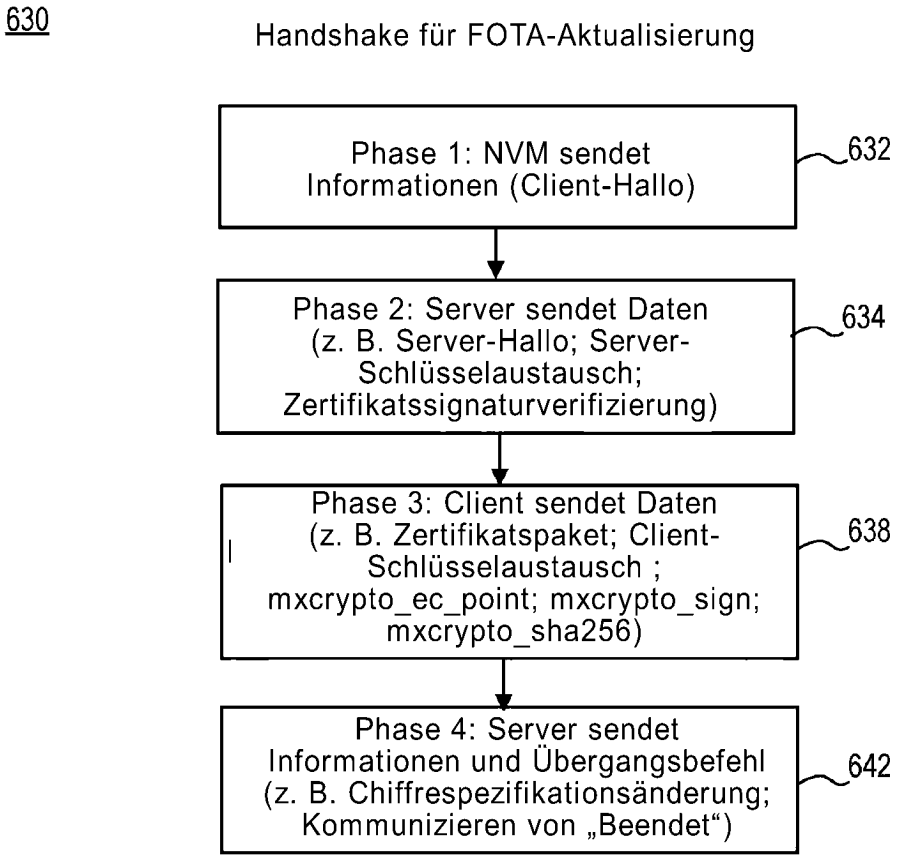

630
Handshake für FOTA-Aktualisierung
Phase 1: NVM sendet Informationen (Client-Hallo)
632
Phase 2: Server sendet Daten (z. B. Server-Hallo; Server-Schlüsselaustausch; Zertifikatssignaturverifizierung)
634
Phase 3: Client sendet Daten (z. B. Zertifikatspaket; Client-Schlüsselaustausch ; mxcrypto_ec_point; mxcrypto_sign; mxcrypto_sha256)
638
Phase 4: Server sendet Informationen und Übergangsbefehl (z. B. Chiffrespezifikationsänderung; Kommunizieren von „Beendet“)
642

FIG. 6A

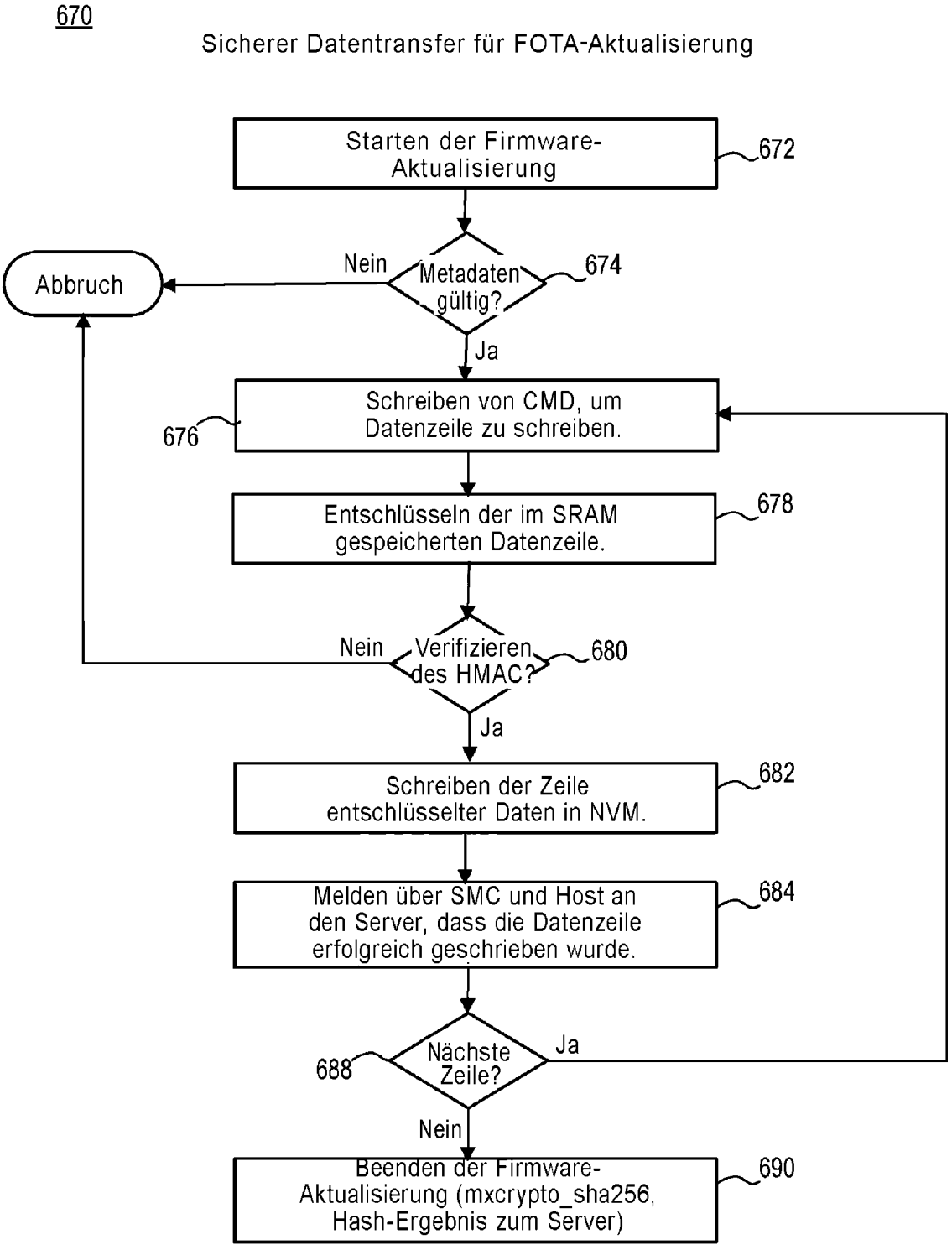

670
Sicherer Datentransfer für FOTA-Aktualisierung
Starten der Firmware-Aktualisierung
672
Metadaten gültig?
674
Nein
Abbruch
Ja
Schreiben von CMD, um Datenzeile zu schreiben.
676
Entschlüsseln der im SRAM gespeicherten Datenzeile.
678
Verifizieren des HMAC?
680
Nein
Ja
Schreiben der Zeile entschlüsselter Daten in NVM.
682
Melden über SMC und Host an den Server, dass die Datenzeile erfolgreich geschrieben wurde.
684
Nächste Zeile?
688
Ja
Nein
Beenden der Firmware-Aktualisierung (mxcrypto_sha256, Hash-Ergebnis zum Server)
690

FIG. 6B

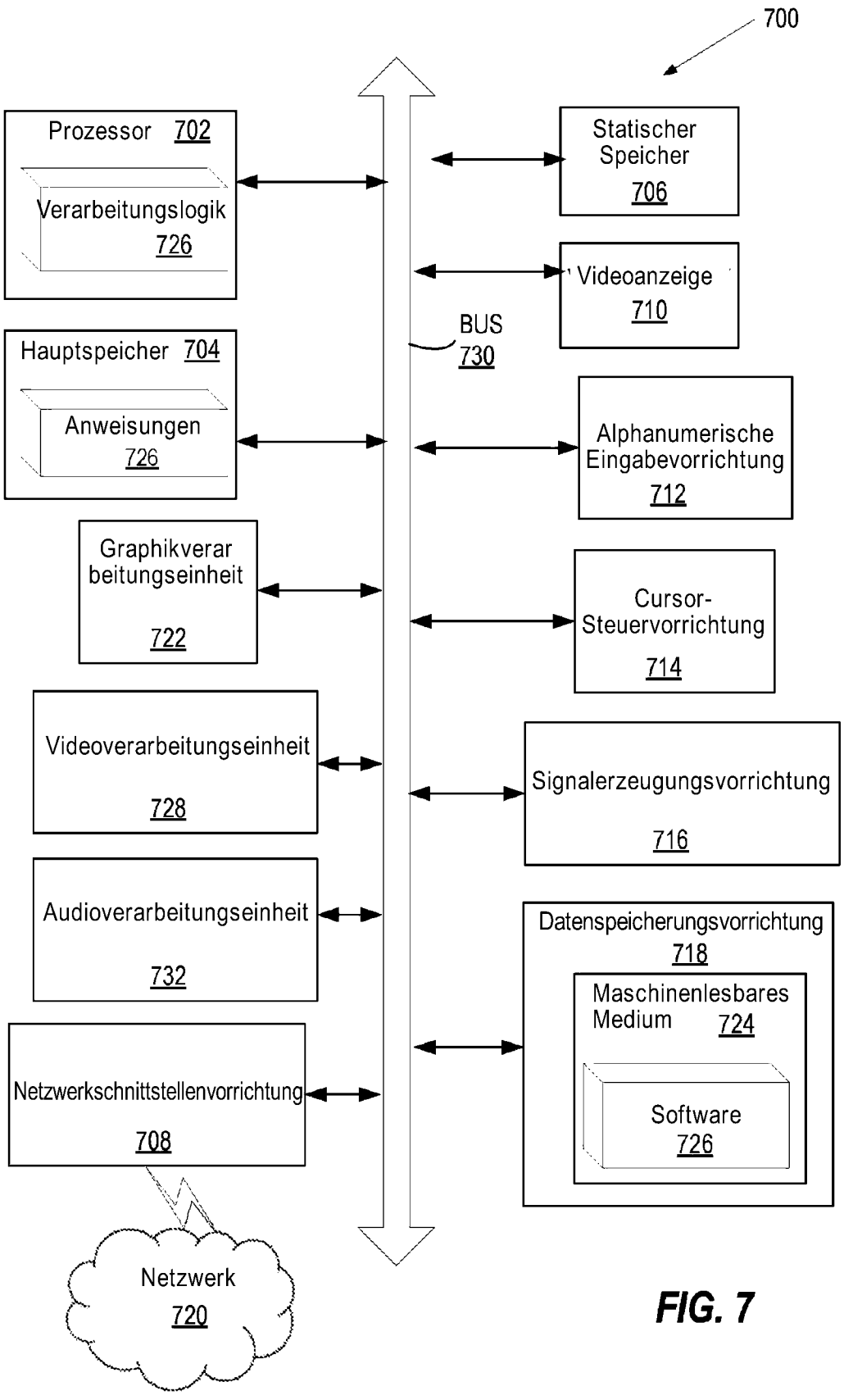


FIG. 7

