República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) BR 112014006543-8 B1

(22) Data do Depósito: 04/10/2012

(45) Data de Concessão: 06/07/2021

(54) Título: SISTEMA DE MICROPROCESSADOR COM AMBIENTE DE TEMPO DE EXECUÇÃO SEGURO

(51) Int.Cl.: G06F 21/53; G06F 9/54.

(30) Prioridade Unionista: 07/10/2011 DE 10 2011 115 135.8.

(73) Titular(es): TRUSTONIC LIMITED.

(72) Inventor(es): STEPHAN SPITZ; MARKUS KÖHLER; ULLRICH MARTINI.

(86) Pedido PCT: PCT EP2012004159 de 04/10/2012

(87) Publicação PCT: WO 2013/050154 de 11/04/2013

(85) Data do Início da Fase Nacional: 19/03/2014

(57) Resumo: SISTEMA DE MICROPROCESSADOR COM AMBIENTE DE TEMPO DE EXECUÇÃO SEGURO. A presente invenção refere-se a um sistema de microprocessador que é implantado ou pode ser implantado em um terminal móvel e compreende: um sistema operacional normal destinado a gerar e manter um ambiente de tempo de execução não seguro e um sistema operacional de segurança destinado a gerar e manter um ambiente de tempo de execução seguro e uma interface de sistema operacional entre o sistema operacional normal e o sistema operacional de segurança, sendo que a dita interface operacional é destinada a controlar a comunicação entre o ambiente de tempo de execução não seguro e o ambiente de tempo de execução seguro no nível do sistema operacional e pelo menos uma interface de filtro que é destinada a controlar de modo seguro a comunicação entre o ambiente de tempo de execução não seguro e o ambiente de tempo de execução seguro em um nível diferente do nível do sistema operacional.

“SISTEMA DE MICROPROCESSADOR COM AMBIENTE DE TEMPO DE EXECUÇÃO SEGURO”.

[001] A presente invenção refere-se a um sistema de microprocessador implantável ou implantado em um terminal móvel com um ambiente de tempo de execução seguro.

[002] Um sistema de microprocessador compreende um microprocessador e uma ou mais memórias voláteis e não voláteis a partir de um ponto de vista funcional, particularmente pelo menos uma memória do sistema operacional, em que um sistema operacional é armazenado, uma ou mais memórias de aplicativo, nas quais os aplicativos que são executáveis pelo microprocessador sob o controle do sistema operacional são armazenados, e uma memória principal, na qual os dados que serão processados pelo microprocessador podem ser armazenados. O sistema operacional gera e mantém um ambiente de tempo de execução durante sua operação no qual os aplicativos são executáveis no sistema de microprocessador. O sistema de microprocessador geralmente é implantado em um circuito integrado, por exemplo, circuito integrado semicondutor, possivelmente também em uma pluralidade de circuitos integrados conectados.

[003] Um terminal móvel é entendido como significando um telefone celular, telefone inteligente, PDA (assistente pessoal digital) ou terminal similar.

[004] A denominação arquitetura ARM Trustzone denota uma conhecida arquitetura bipartida de tempo de execução da empresa ARM para um sistema de microprocessador, sendo que essa arquitetura compreende dois ambientes de tempo de execução. Um primeiro ambiente de tempo de execução, não seguro, chamado “Zona Normal” ou “Mundo Normal”, é controlado por um sistema operacional normal. Um segundo ambiente de tempo de execução, seguro ou confiável, chamado “Zona de Confiança” ou “Mundo Seguro” é controlado por um sistema operacional de segurança.

[005] O sistema operacional normal pode ser, por exemplo, um sistema operacional convencional como o Android, Windows Phone, Symbian ou

similares.

[006] A requerente do presente pedido de patente fabrica e vende um sistema operacional de segurança para circuitos integrados que serão implantados em terminais móveis, sob a designação comercial MOBICORE. Particularmente, os aplicativos de segurança essencial e algumas funções periféricas (por exemplo, *driver* de teclado) são controlados de modo seguro pelo sistema operacional de segurança MOBICORE. Aplicativos sob o sistema operacional de segurança também são designados como trustlets, em analogia aos termos “Trust” e “Applet”. Mais aplicativos e outras funções periféricas (por exemplo, *driver* de visor, NFC *driver* de interface, camada de interface de rádio RIL) são controlados pelo sistema operacional normal (também chamado Rich OS ou HLOS). O sistema operacional de segurança compreende uma chave criptográfica, a chave de autenticação Kauth, através da qual o sistema operacional de segurança pode se identificar em relação a um sistema de segundo plano. Opcionalmente, a chave de autenticação Kauth é provida adicionalmente para operar um canal de comunicação segura entre o circuito integrado ou o terminal móvel e o sistema de segundo plano. O sistema de segundo plano é, por exemplo, um sistema de segundo plano conhecido por si só de uma rede de rádio móvel.

[007] Os acessos na parte do sistema operacional normal, a partir do ambiente de tempo de execução não seguro ao ambiente de tempo de execução seguro, por exemplo, para chamar um aplicativo no ambiente de tempo de execução seguro (Trustlet), ou para trocar dados, são realizados como padrão exclusivamente por meio de um *driver* de segurança, por exemplo, *driver* Mobicore, que é implantado dentro do ambiente de tempo de execução não seguro, mas é controlado pelo sistema operacional de segurança e funciona no nível do sistema operacional. O *driver* de segurança pode, por exemplo, transferir tarefas do sistema operacional normal para o sistema operacional de segurança e encaminhar os resultados em relação às tarefas, sendo que ditos resultados providos pelo sistema operacional de

segurança para o sistema operacional normal. O *driver* de segurança media um canal de comunicação segura entre o sistema operacional normal e o sistema operacional de segurança, sendo que o dito canal é completamente suficiente para troca de volumes de dados relativamente pequenos.

[008] Um aplicativo (Trustlet) ou função periférica implantado no ambiente de tempo de execução seguro pode ser acessado apenas pelo sistema operacional de segurança. Em contrapartida, o Trustlet não pode ser acessado diretamente do sistema operacional normal.

[009] Virtualmente todas as estruturas periféricas do terminal móvel (exceto, por exemplo, o teclado), ou seja, estruturas do terminal móvel que vão além do microprocessador central, são implantadas sob o sistema operacional normal e são, portanto, inseguras à extensão máxima, já que os *drivers* periféricos respectivos são implantados no ambiente de tempo de execução não seguro, “Zona Normal”.

[010] Alguns processos de comunicação das estruturas periféricas com o sistema operacional ou através do sistema operacional de segurança são de segurança essencial, por exemplo, aqueles nos quais dados secretos como, por exemplo, o número secreto do PIN, são processados, ou nos quais comandos de segurança relevantes são processados, por exemplo, um comando para confirmar ou iniciar uma transação de pagamento. Para processos de comunicação de segurança tão essencial, um *link* seguro de estruturas periféricas seria desejado para impedir a observação discreta de dados ou manipulação de comandos.

[011] A possibilidade mais direta para tornar a operação das estruturas periféricas mais segura é implementar os respectivos *drivers* periféricos sob o sistema operacional de segurança ao invés de sob o sistema operacional normal (O *driver* de teclado relativamente pequeno e compacto em um terminal móvel com MobiCore como sistema operacional de segurança é, por exemplo, muitas vezes, já implantado sob o sistema operacional de segurança como padrão). Isso tem a consequência, porém, de que todos os acessos à estrutura

periférica são necessariamente efetuados pelo sistema operacional de segurança. Já que, sob o sistema operacional de segurança, medidas complexas de segurança como autenticações e criptografias são exigidas para todos os acessos, mover os *drivers* do sistema operacional normal para o sistema operacional de segurança reduz a velocidade da operação do terminal móvel consideravelmente. Porém, uma proporção considerável de acessos a uma estrutura periférica não é de segurança essencial e eles poderiam prosseguir tão facilmente sob o sistema operacional normal não seguro, porém, mais rápido.

[012] Aplicativos com uma ampla variedade de finalidades de uso, por exemplo, transações monetárias (banco, pagamento) ou venda de bilhetes para meios de transporte (emissão de bilhete) ou pontos de bônus e sistemas de desconto (fidelidade) são amplamente implantados, pelo menos em parte, sob o controle do sistema operacional de segurança, de forma executável no ambiente de execução seguro. Neste caso, pelo menos partes do aplicativo são implantadas adicionalmente ou apenas sob o controle do sistema operacional normal, de forma executável no ambiente de execução, particularmente, partes do aplicativo que são direcionadas a uma interação com um usuário ou uma interface do usuário do terminal móvel. Se a comunicação é necessária entre partes do aplicativo nos ambientes de tempo de execução seguro e não seguro, isso acontece no nível do sistema operacional por meio do *driver* de segurança, que torna a comunicação lenta.

[013] A invenção é baseada no objetivo de prover um sistema de microprocessador implantável ou implantado em um terminal móvel com um ambiente de tempo de execução seguro, que o microprocessador permite uma comunicação segura e ao mesmo tempo eficiente entre o ambiente de tempo de execução não seguro e o ambiente de tempo de execução seguro.

[014] O objetivo é alcançado por meio de um sistema de microprocessador, de acordo com a reivindicação 1. Modalidades vantajosas da invenção são especificadas nas reivindicações dependentes.

[015] O sistema de microprocessador, de acordo com a invenção e de acordo com a reivindicação independente 1, é implantável ou implantado em um terminal móvel e compreende:

- um sistema operacional normal, que é destinado a gerar e manter um ambiente de tempo de execução não seguro,

- um sistema operacional de segurança, que é destinado a gerar e manter um ambiente de tempo de execução seguro, e

- uma interface do sistema operacional entre o sistema operacional normal e o sistema operacional de segurança, sendo que a dita interface do sistema operacional é destinada a controlar de modo seguro a comunicação entre o ambiente de tempo de execução não seguro e o ambiente de tempo de execução seguro no nível do sistema operacional.

[016] O sistema de microprocessador é caracterizado por pelo menos uma interface de filtro, que é destinada a controlar de modo seguro a comunicação entre o ambiente de tempo de execução não seguro e o ambiente de tempo de execução seguro em um nível diferente do nível do sistema operacional.

[017] A interface de filtro permite comunicação acelerada e ao mesmo tempo segura já que provê, além do canal de comunicação no nível do sistema operacional, um canal mais seguro de comunicação, que pode ser usado para comunicação mais direta do que a comunicação através do desvio do sistema operacional.

[018] Por conseguinte, a reivindicação 1 provê um sistema de microprocessador que permite a comunicação segura e ao mesmo tempo eficiente entre o ambiente de tempo de execução não seguro e o ambiente de tempo de execução seguro.

[019] A interface de filtro tem uma primeira interface de filtro parcial insegura, que é destinada a ser executada no ambiente de tempo de execução não seguro e uma segunda interface de filtro parcial segura, que é destinada a ser executada no ambiente de tempo de execução seguro. Como resultado, a

interface de filtro pode mediar entre os ambientes de tempo de execução seguro e não seguro.

[020] Em conformidade com uma alternativa, a interface de filtro provida é uma interface de filtro de *hardware*, que é destinada a controlar de modo seguro a comunicação entre o ambiente de tempo de execução não seguro e o ambiente de tempo de execução seguro no nível do *hardware*. A comunicação segura entre os ambientes de tempo de execução seguro e não seguro pode, então, ocorrer imediatamente e diretamente no nível do *hardware* e, assim, de forma acelerada.

[021] O controle no nível do *hardware* é opcionalmente efetuado como controle acionando-se diretamente os componentes de *hardware* como, por exemplo, estruturas periféricas como visor, cartão SIM, etc. (consulte abaixo), ou opcionalmente como controle por meio de um sistema de barramento, por exemplo, por meio de um barramento I2C.

[022] Em conformidade com a alternativa, de acordo com a qual a interface de filtro provida é uma interface de filtro de *hardware*, o sistema de microprocessador compreende, ainda, opcionalmente:

> pelo menos um *driver* para uma estrutura periférica do terminal móvel, sendo que o dito pelo menos um *driver* é acionável pelo sistema operacional normal e executável no ambiente de tempo de execução não seguro, e um *driver* – preferivelmente reduzido – para a estrutura periférica, sendo que o dito *driver* é acionável pelo sistema operacional de segurança e executável no ambiente de tempo de execução seguro e preferivelmente com apenas um subconjunto das funcionalidades do *driver* que é executável no ambiente de tempo de execução não seguro. Neste caso, a interface de filtro de *hardware* é destinada a controlar de modo seguro a comunicação entre o *driver* para uma estrutura periférica que é executável no ambiente de tempo de execução não seguro e o *driver* preferivelmente reduzido para a estrutura periférica que é executável

no ambiente de tempo de execução seguro, no nível do *hardware*.

[023] A interface de filtro de *hardware*, portanto, permite a comunicação direta entre partes do *driver* implantadas sob o sistema operacional normal e partes do *driver* implantadas sob o sistema operacional de segurança através da interface de filtro de *hardware*.

[024] Como a estrutura periférica do terminal, por exemplo, faz-se provisão de: uma interface NFC, uma camada de interface de rádio RIL envolvida em configurar as conexões para dentro da rede móvel de rádio, um elemento de segurança SE usado para autenticação e criptografia (por exemplo, cartão SIM ou cartão USIM ou cartão Secure MicroSD ou Elemento Seguro Inserido implantado permanentemente), um visor, particularmente um visor de tela sensível a toque (touch-screen), uma rotina de verificação de PIN, que pode ser usada para verificar a entrada do número secreto PIN (número de identificação pessoal) por meio do teclado, rotinas discadas para discar para redes diferentes da rede de rádio móvel, como por exemplo, a Internet (IP) ou redes virtuais privadas (VPN), memória de armazenamento temporário (por exemplo, armazenamento temporário de quadro).

[025] Opcionalmente, se os *drivers* forem implantados para uma pluralidade de estruturas periféricas no sistema do microprocessador, uma interface de *hardware* comum é provida para todas as estruturas periféricas, sendo que a dita interface de *hardware* permite acessos entre os ambientes de tempo de execução não seguro e seguro no nível do *hardware* exclusivamente entre os *drivers* associados à mesma estrutura periférica e impedindo os acessos entre os *drivers* associados às estruturas periféricas diferentes no nível do *hardware*. Em virtude do fato de que acessos diretos no nível do *hardware* são reservados apenas para *drivers* mutuamente correspondentes em relação à mesma estrutura periférica, um acesso impróprio ao *driver* por *malware*, por exemplo, um Cavalo de Troia ou similar, é evitado.

[026] Os acessos entre os *drivers* associados a diferentes estruturas periféricas podem ocorrer no máximo por meio do *driver* de segurança no nível

do sistema operacional, desde que esses excessos sejam permissíveis em conformidade com uma verificação do *driver* de segurança. O *driver* de segurança identificaria e repeliria o malware, por exemplo.

[027] Opcionalmente, a interface de filtro de *hardware* é destinada a verificar (por exemplo, no ambiente de tempo de execução não seguro) acessos direcionados ao *driver* para uma estrutura periférica no que se refere à relevância para segurança e, no caso em que o acesso é identificado como relevante para segurança, para transmitir o acesso para o *driver* reduzido no ambiente de tempo de execução seguro e, no caso em que o acesso é identificado como não relevante para segurança, para transmitir o acesso para o *driver* no ambiente de tempo de execução não seguro.

[028] Opcionalmente, o *driver* para uma estrutura periférica, que é executável no ambiente de tempo de execução seguro, é reduzido ao efeito de apenas compreender funcionalidades que são necessárias para processar acessos relevantes à segurança. Em contrapartida, o *driver* para uma estrutura periférica que é executável no ambiente de tempo de execução não seguro compreende todas as funcionalidades do *driver*.

[029] De acordo com uma alternativa adicional, a interface de filtro provida é uma interface de filtro de aplicativo, que é destinada a controlar de forma segura a comunicação entre o ambiente de tempo de execução não seguro e o ambiente de tempo de execução seguro no nível do aplicativo. A interface de filtro de aplicativo provê um canal de comunicação direto e seguro, no nível do aplicativo, que permite comunicação mais rápida do que através do desvio através do nível do sistema operacional e do *driver* de segurança que opera no nível do sistema operacional.

[030] O sistema de microprocessador, opcionalmente, ainda, compreende pelo menos um aplicativo que é pelo menos parcialmente acionável pelo sistema operacional normal e executável no ambiente de tempo de execução não seguro, e é pelo menos parcialmente acionável pelo sistema operacional de segurança e executável no ambiente de tempo de execução

seguro. Nesse caso, a interface de filtro de aplicativo é destinada a controlar de forma segura a comunicação entre o aplicativo pelo menos parcial que é executável no ambiente de tempo de execução não seguro e o aplicativo pelo menos parcial que é executável no ambiente de tempo de execução seguro, no nível do aplicativo.

[031] O aplicativo pode ser, por exemplo, um aplicativo bancário ou de pagamento para transações monetárias, ou um aplicativo de emissão de bilhetes para compra e administração de bilhetes para uma ampla variedade de propósitos, por exemplo, meios de transporte. Pelo menos funcionalidades relevantes para a segurança do aplicativo são opcionalmente implantadas sob o sistema operacional de segurança. Funcionalidades suplementares e irrelevantes para a segurança são opcionalmente implantadas, adicional ou exclusivamente, sob o sistema operacional normal.

[032] Em contrapartida aos *drivers* em que o acionamento na maioria dos casos será implantado predominantemente sob o sistema operacional normal, no caso de aplicativos como aplicativos bancários, por exemplo, a esmagadora maioria normalmente será implantada sob o sistema operacional de segurança.

[033] Opcionalmente, se uma pluralidade de aplicativos for implantada no sistema de microprocessador, uma interface de filtro de aplicativo comum é provida para todos os aplicativos, sendo que a dita interface de filtro de aplicativo permite acessos entre os ambientes de tempo de execução não seguro e seguro no nível do aplicativo, exclusivamente entre as partes associadas ao mesmo aplicativo no ambiente de tempo de execução não seguro e, respectivamente, seguro, impedindo acessos entre partes associadas aos aplicativos diferentes no nível do aplicativo. Em virtude do fato de que os acessos diretos no nível do aplicativo são reservados apenas para as partes mutuamente correspondentes do mesmo aplicativo, um acesso impróprio ao aplicativo por malware, por exemplo, um Cavalo de Troia ou similar, é evitado.

[034] Acessos de aplicativo entre diferentes aplicativos através do limite

entre os ambientes de tempo de execução seguro e não seguro podem ocorrer, no máximo, apenas no nível do sistema operacional através do *driver* de segurança.

[035] Opcionalmente, a interface de filtro de aplicativo é destinada a verificar (por exemplo, no ambiente de tempo de execução não seguro) acessos direcionados ao aplicativo, no que se refere à relevância para segurança e, no caso de o acesso ser identificado como relevante para segurança, para transmitir o acesso para o aplicativo pelo menos parcial no ambiente de tempo de execução seguro e, no caso de o acesso ser identificado como não relevante para segurança, para transmitir o acesso para o aplicativo pelo menos parcial no ambiente de tempo de execução não seguro.

[036] A interface de filtro de aplicativo pode ser provida como uma alternativa ou adicionalmente à interface de filtro de *hardware*. Opcionalmente, a interface de filtro de aplicativo e a interface de filtro de *hardware* são ambas providas e destinadas à interação.

[037] Em concordância com um desenvolvimento da invenção, a interface de filtro é acoplada a um dispositivo de saída de usuário, por exemplo, um alto-falante, alerta vibratório ou similar a um terminal móvel (por exemplo, telefone celular), de forma que no caso de qualquer acesso efetuado através da interface de filtro, do ambiente de tempo de execução não seguro para o ambiente de tempo de execução seguro ou/e do ambiente de tempo de execução seguro para o ambiente de tempo de execução não seguro, ou seja, através do limite entre os ambientes de tempo de execução seguro e não seguro, uma mensagem de usuário detectável por um usuário do microprocessador é emitida pelo dispositivo de saída do usuário. Como resultado, o usuário é informado de qualquer acesso ocorrendo através do limite entre os ambientes de tempo de execução seguro e não seguro e pode, por exemplo, reconhecer se dados essenciais para a segurança chegam e são encaminhados da interface de filtro para o ambiente de tempo de execução seguro ou/e se dados são emitidos do ambiente de tempo de execução seguro

para o ambiente de tempo de execução não seguro. A saída de usuário pode ser configurada, por exemplo, como uma melodia que é emitida pelo terminal móvel.

[038] Opcionalmente, para acessos através do limite entre os ambientes de tempo de execução seguro e não seguro, do ambiente de tempo de execução não seguro para o ambiente de tempo de execução seguro, por um lado, e do ambiente de tempo de execução seguro para o ambiente de tempo de execução não seguro, por outro lado, saídas de usuário diferentes, perceptível para o usuário, são emitidas. Como forma de exemplo, melodias diferentes são emitidas dependendo da direção do acesso, do ambiente de tempo de execução não seguro para o ambiente de tempo de execução seguro ou do ambiente de tempo de execução seguro para o ambiente de tempo de execução não seguro.

[039] Em concordância com uma variante, a mensagem de usuário é sempre operada durante a operação do microprocessador dentro do ambiente de tempo de execução seguro. Nessa variante, desta forma, a saída de mensagem de usuário no caso de um acesso do ambiente de tempo de execução não seguro para o ambiente de tempo de execução seguro é emitida temporariamente imediatamente após o acesso, para notificar o usuário que a operação está agora ocorrendo no ambiente de tempo de execução seguro. No caso de um acesso do ambiente de tempo de execução seguro para o ambiente de tempo de execução não seguro, nessa variante, a mensagem do usuário é emitida temporariamente imediatamente antes do acesso para alertar o usuário de que vai deixar o ambiente de tempo de execução seguro dali em diante. Alternativamente ou adicionalmente, uma mensagem de usuário pode ser emitida após o ambiente de tempo de execução seguro ser deixado pela execução do programa. Da mesma forma, alternativamente ou adicionalmente, uma mensagem de usuário pode ser emitida antes do ambiente de tempo de execução seguro ser inserido pela execução do programa.

[040] A invenção é explicada em maiores detalhes abaixo, com base

nas modalidades exemplificativas e com referência ao desenho, em que na ilustração diagramática:

[041] A Figura 1 mostra um sistema de microprocessador que compreende uma interface de filtro de aplicativo, em concordância com uma modalidade da invenção;

[042] A Figura 2 mostra um sistema de microprocessador que compreende uma interface de filtro de *hardware*, em concordância com uma modalidade da invenção;

[043] A Figura 3 mostra um sistema de microprocessador que compreende uma interface de filtro de aplicativo e uma interface de filtro de *hardware*, em concordância com uma das modalidades da invenção.

[044] A Figura 1 mostra um sistema de microprocessador com uma arquitetura ARM, sendo que tem um ambiente de tempo de execução não seguro 100 e um ambiente de tempo de execução seguro 200, e sendo que tem uma interface de filtro de aplicativo TLC (Conector Trustlet) implantada em cada caso parcialmente no ambiente de tempo de execução não seguro 100 e no ambiente de tempo de execução seguro 200, em concordância com uma modalidade da invenção. O ambiente de tempo de execução não seguro 100 é controlado por um sistema operacional normal HLOS. O ambiente de tempo de execução seguro 200 é controlado por um sistema operacional de segurança MobiCore. Acessos entre o ambiente de tempo de execução não seguro 100 e o ambiente de tempo de execução seguro 200, no nível do sistema operacional, são realizados por meio de um *driver* de segurança Mobicore. A maior parte APP-S de um aplicativo APP, por exemplo, de um aplicativo de transações de pagamento, é implantada sob o sistema operacional de segurança MobiCore, de uma forma executável no ambiente de tempo de execução seguro 200. O restante APP-N do aplicativo APP, por exemplo, um aplicativo de transações de pagamento, é implantado sob o sistema operacional normal HLOS, de uma forma executável no ambiente de tempo de execução não seguro 100. A interface de filtro de aplicativo TLC (Conector

Trustlet), de acordo com a invenção, permite um acesso direto entre a parte APP-S do aplicativo APP que é implantada sob o sistema operacional de segurança MobiCore, de uma forma executável no ambiente de tempo de execução seguro 200, e a parte APP-N do aplicativo APP que é implantada sob o sistema operacional normal HLOS, de uma forma executável no ambiente de tempo de execução não seguro 100, diretamente no nível do aplicativo. Um *driver* para uma estrutura periférica PER, por exemplo, um visor sensível ao toque, é implantado de forma convencional sob o sistema operacional normal HLOS, em concordância com a Figura 1, e pode se comunicar com o ambiente de tempo de execução seguro 200 no máximo através do *driver* de segurança MobiCore. Um novo *driver* reduzido PER-S implantado sob o sistema operacional de segurança MobiCore pode ser contatado a partir do ambiente de tempo de execução não seguro 100, no máximo, através do *driver* de segurança Mobicore. No ambiente de tempo de execução não seguro 100, o *driver* periférico PER se comunica com o sistema operacional normal HLOS através de um sistema de barramento I2C, de uma forma controlada pelo *driver* Linux I2C. No ambiente de tempo de execução seguro 200, o *driver* periférico reduzido PER-S se comunica com o sistema operacional de segurança MobiCore através de um sistema de barramento I2C implantado no ambiente de tempo de execução seguro 200, de uma forma controlada pelo mesmo *driver* I2C. A comunicação entre os *drivers* periféricos PER e PER-S pode ocorrer, no máximo, no nível do sistema operacional por meio do *driver* MobiCore.

[045] A Figura 2 mostra um sistema de microprocessador que compreende uma interface de filtro de *hardware*, designada como *driver* I2C virtual, em concordância com uma modalidade da invenção. Partes com a mesma designação, como na Figura 1, correspondem às partes da Figura 1, a menos que descrito no presente documento de outro modo. Um *driver* para uma estrutura periférica PER, por exemplo, um visor sensível ao toque, é implantado sob o sistema operacional normal HLOS. Um novo *driver* periférico

reduzido PER-S para a estrutura periférica é implantado sob o sistema operacional de segurança MobiCore. O *driver* I2C virtual (interface de filtro de *hardware* exemplificativa) implantado em cada caso parcialmente no ambiente de tempo de execução não seguro 100 e no ambiente de tempo de execução seguro 200 permite uma comunicação segura entre o *driver* periférico PER implantado sob o sistema operacional normal HLOS e o *driver* periférico reduzido PER-S MobiCore, implantado sob o sistema operacional de segurança, no nível do *hardware*, mais precisamente no nível do barramento I2C. No ambiente de tempo de execução não seguro 100, o *driver* periférico PER comunica-se com o sistema operacional normal HLOS através do *driver* Linux I2C. No ambiente de tempo de execução seguro 200, o *driver* periférico reduzido PER-S comunica-se através do mesmo *driver* I2C, com o sistema operacional de segurança MobiCore. A interface de filtro de *hardware*, destinada como *driver* I2C virtual, provê um canal de comunicação segura entre o *driver* Linux I2C do sistema de barramento sob HLOS e o *driver* I2C do sistema de barramento sob MobiCore, em um nível de *hardware*, que é mais próximo do nível do *driver* periférico do que do nível do sistema operacional. Na modalidade exemplificativa da Figura 2, as partes APP-N, APP-S do aplicativo APP podem se comunicar entre si, no máximo, apenas no nível do sistema operacional, principalmente por meio do *driver* MobiCore.

[046] A Figura 3 mostra um sistema de microprocessador que compreende uma interface de filtro de aplicativo TLC e uma interface de filtro de *hardware*, *driver* I2C virtual, em concordância com uma modalidade da invenção. A Figura 3, portanto, em primeiro lugar, ilustra uma combinação das modalidades da Figura 1 e Figura 2. Além do mais, a interface de filtro de aplicativo TLC e a interface de filtro de *hardware*, *driver* I2C virtual, cooperam. Na parte do ambiente de tempo de execução não seguro 100, a interface de filtro de *hardware* (*driver* I2C virtual) se comunica com o *driver* periférico PER através do sistema de barramento do *driver* Linux I2C. Em direção ao ambiente de tempo de execução seguro 200, a interface de filtro de *hardware* (*driver* I2C

virtual) se comunica com a interface de filtro de aplicativo TLC e, assim, muda do nível do *hardware* para o nível do aplicativo. Finalmente, a interface de filtro de aplicativo TLC se comunica dentro do ambiente de tempo de execução seguro 200 no nível do aplicativo com o *driver* periférico reduzido PER-S.

REIVINDICAÇÕES

1. Sistema de microprocessador, implantável ou implantado em um terminal móvel e que compreende:

- um sistema operacional normal (HLOS), que é destinado a gerar e manter um ambiente de tempo de execução não seguro (100),

- um sistema operacional de segurança (MobiCore), que é destinado a gerar e manter um ambiente de tempo de execução seguro (200), em que o ambiente de tempo de execução não seguro (100) e o ambiente de tempo de execução seguro (200) possuem um nível de sistema de operação, um nível de hardware e um nível do aplicativo, e

- uma interface de sistema operacional (*driver* MobiCore) entre o sistema operacional normal (HLOS) e o sistema operacional de segurança (MobiCore), em que a dita interface de sistema operacional destinada a controlar de modo seguro a comunicação entre o ambiente de tempo de execução não seguro (100) e o ambiente de tempo de execução seguro (200) no nível do sistema operacional em virtude das operações de acesso entre o ambiente de tempo de execução não seguro (100) e o ambiente de tempo de execução seguro (200) no nível do sistema operacional, através de um *driver* de segurança,

caracterizado pelo fato de que

pelo menos uma interface de filtro (*driver* I2C virtual; TLC), que possui uma primeira interface de filtro parcial não segura configurada para execução no ambiente de tempo de execução não seguro (100) e uma segunda interface de filtro parcial segura configurada para execução no ambiente de tempo de execução seguro (200), e que é destinada a controlar de modo seguro a comunicação entre o ambiente de tempo de execução não seguro (100) e o ambiente de tempo de execução seguro (200) em um nível diferente do nível do sistema operacional.

2. Sistema de microprocessador, de acordo com a reivindicação 1, caracterizado pelo fato de que a interface de filtro provida é uma interface de filtro de *hardware* (*driver* I2C virtual), que é destinada a controlar de modo

seguro a comunicação entre o ambiente de tempo de execução não seguro (100) e o ambiente de tempo de execução seguro (200) no nível do *hardware*.

3. Sistema de microprocessador, de acordo com a reivindicação 2, caracterizado pelo fato de que o controle no nível do *hardware* é efetuado como controle via um sistema de barramento (I2C).

4. Sistema de microprocessador, de acordo com a reivindicação 2 ou 3, que compreende adicionalmente:

pelo menos um *driver* (PER) para uma estrutura periférica do terminal móvel, sendo que pelo menos um dito *driver* é acionável pelo sistema operacional normal (HLOS) e em que é executável no ambiente de tempo de execução não seguro (100), e

caracterizado pelo fato de compreender ainda,

um *driver* preferivelmente reduzido (PER-S) para a estrutura periférica, sendo que dito *driver* é acionável pelo sistema operacional de segurança (MobiCore) e que é executável no ambiente de tempo de execução seguro (200) e que, preferivelmente, tenha apenas um subconjunto das funcionalidades do *driver* (PER) que é executável no ambiente de tempo de execução não seguro (100),

e de que

a interface de filtro de *hardware* (*driver* I2C virtual) é destinada a controlar de modo seguro a comunicação entre o *driver* (PER) para uma estrutura periférica que é executável no ambiente de tempo de execução não seguro (100) e o *driver* preferivelmente reduzido (PER-S) para a estrutura periférica que é executável no ambiente de tempo de execução seguro (200), no nível do *hardware*.

5. Sistema de microprocessador, de acordo com a reivindicação 4, caracterizado pelo fato de que a interface de filtro de *hardware* (*driver* I2C virtual), é destinada a verificar os acessos direcionados ao *driver* (PER, PER-S) para uma estrutura periférica no que se refere à relevância para segurança e, no caso de o acesso ser identificado como relevante para segurança, para

transmitir o acesso para o *driver* preferivelmente reduzido (PER-S) no ambiente de tempo de execução seguro (200) e, no caso de o acesso ser identificado como não relevante para a segurança, para transmitir o acesso para o *driver* (PER) no ambiente de tempo de execução não seguro (100).

6. Sistema de microprocessador, de acordo com qualquer uma das reivindicações 1 a 5,

caracterizado pelo fato de que a interface de filtro provida é uma interface de filtro de aplicativo (TLC), que é destinada a controlar de modo seguro a comunicação entre o ambiente de tempo de execução não seguro (100) e o ambiente de tempo de execução seguro (200) no nível do aplicativo.

7. Sistema de microprocessador, de acordo com a reivindicação 6, que compreende adicionalmente:

pelo menos um aplicativo (APP),

caracterizado pelo fato de que,

o aplicativo (APP) é pelo menos parcialmente (APP-N) acionável pelo sistema operacional normal (HLOS) e executável no ambiente de tempo de execução não seguro (100), e é pelo menos parcialmente (APP-S) acionável pelo sistema operacional de segurança (MobiCore) e executável no ambiente de tempo de execução seguro (200),

e de que

a interface de filtro de aplicativo (TLC) é destinada a controlar de modo seguro a comunicação entre o aplicativo pelo menos parcial (APP-N) que é executável no ambiente de tempo de execução não seguro (100) e o aplicativo pelo menos parcial (APP-S) que é executável no ambiente de tempo de execução seguro (200), no nível do aplicativo.

8. Sistema de microprocessador, de acordo com a reivindicação 7, caracterizado pelo fato de que a interface de filtro de aplicativo (TLC) é destinada a verificar acessos direcionados ao aplicativo (APP) no que se refere à relevância para segurança e, no caso de o acesso ser identificado como relevante para a segurança, a transmitir o acesso para o aplicativo pelo menos

parcial (APP-S) no ambiente de tempo de execução seguro (200) e, no caso de o acesso ser identificado como não relevante para segurança, para transmitir o acesso para o aplicativo pelo menos parcial (APP-N) no ambiente de tempo de execução não seguro (100).

9. Sistema de microprocessador, de acordo com qualquer uma das reivindicações 1 a 8, caracterizado pelo fato de que a interface de filtro (*driver* I2C virtual; TLC) é acoplada a um dispositivo de saída do usuário, de forma que na ocasião de qualquer acesso efetuado através da interface de filtro (*driver* I2C virtual; TLC) do ambiente de tempo de execução não seguro (100) para o ambiente de tempo de execução seguro (200) ou/e do ambiente de tempo de execução seguro (200) para o ambiente de tempo de execução não seguro (100), uma mensagem de usuário detectável por um usuário do microprocessador é emitida pelo dispositivo de saída do usuário.

10. Sistema de microprocessador, de acordo com a reivindicação 9, caracterizado pelo fato de que a interface de filtro (*driver* I2C virtual; TLC) é acoplada ao dispositivo de saída do usuário de forma que, para os acessos a partir do ambiente de tempo de execução não seguro (100) para o ambiente de tempo de execução seguro (200), por um lado, e a partir do ambiente de tempo de execução seguro (200) para o ambiente de tempo de execução não seguro (100), por outro lado, são emitidas saídas de usuário diferentes, distinguíveis pelo usuário.

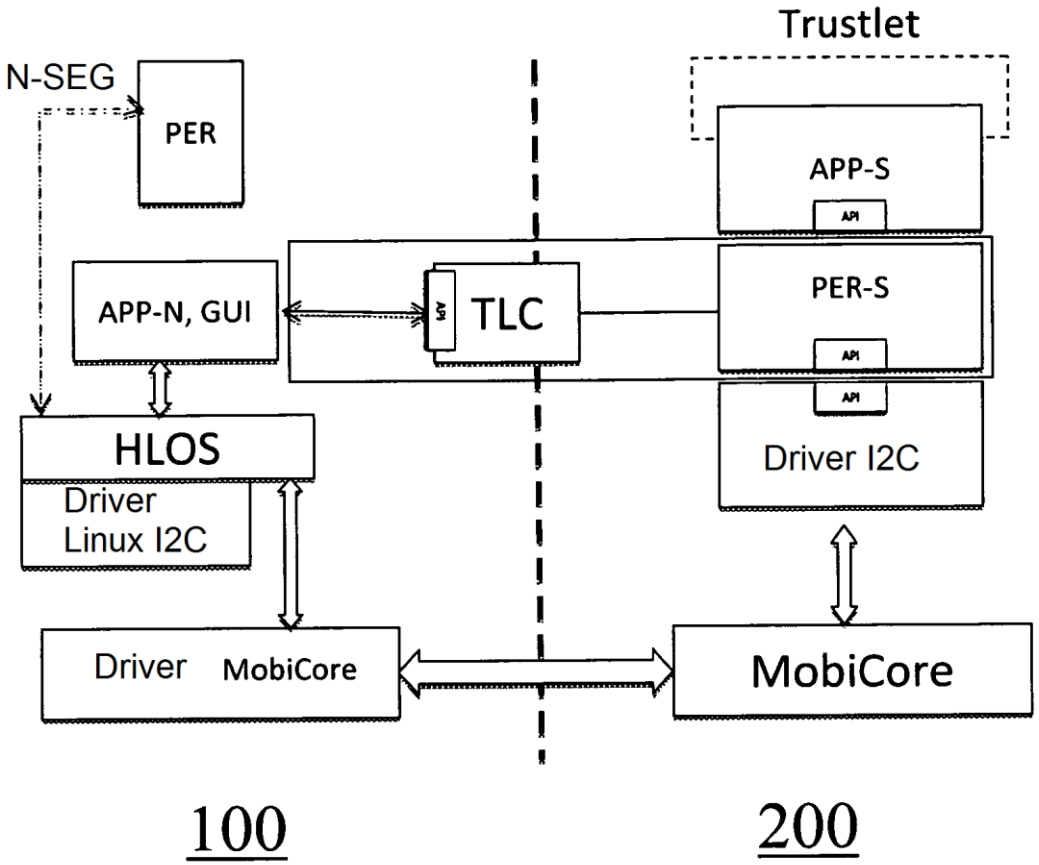

Trustlet
N-SEG
PER
APP-S
API
APP-N, GUI
API
TLC
PER-S
API
API
HLOS
Driver I2C
Driver
Linux I2C
Driver MobiCore
MobiCore
100
200

FIG. 1

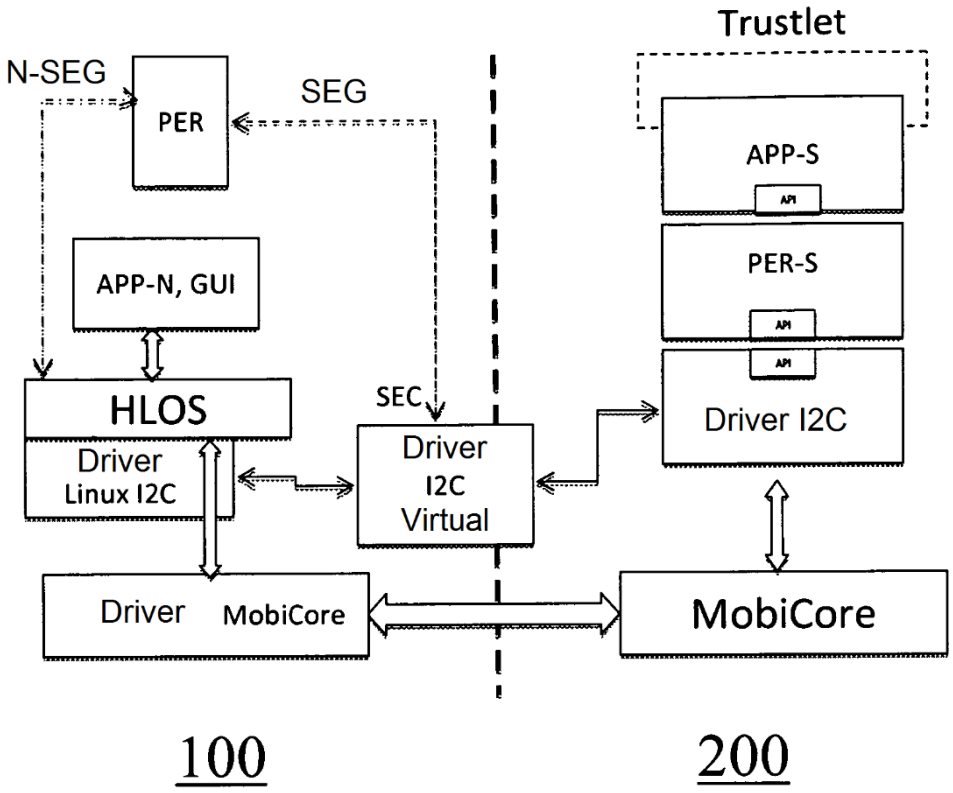

Trustlet
N-SEG
PER
SEG
APP-S
API
APP-N, GUI
PER-S
API
API
HLOS
SEC
Driver I2C
Driver
Linux I2C
Driver
I2C
Virtual
Driver
MobiCore
MobiCore
100
200

FIG. 2

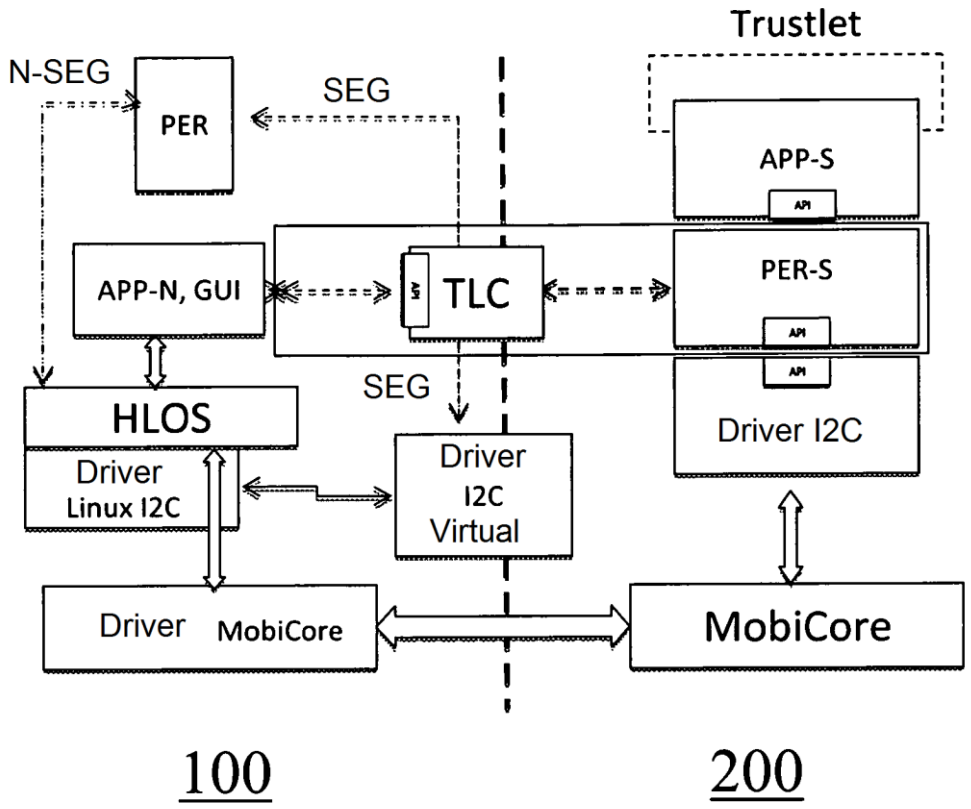

Trustlet
N-SEG
PER
SEG
APP-S
API
APP-N, GUI
TLC
PER-S
API
API
SEG
HLOS
Driver I2C
Driver
Linux I2C
Driver
I2C
Virtual
Driver MobiCore
MobiCore
100
200

FIG. 3