Actualização Segura e Automática de Aplicações em Terminais de Venda

Manuel Mendonça
mendonca27@sapo.pt

Nuno Ferreira Neves
nuno@di.fc.ul.pt

Departamento de Informática
Faculdade de Ciências da Universidade de Lisboa
Bloco C6, Campo Grande, 1749-016 Lisboa

RESUMO

Actualmente, a maioria das transacções electrónicas são efectuadas usando cartões de crédito ou débito em terminais POS localizados nos estabelecimentos comerciais. No entanto, o sucesso desta forma de pagamento tem custos associados à gestão e manutenção dos muitos equipamentos existentes. Em particular, existe um custo não desprezável relacionado com a actualização das aplicações que correm nesses equipamentos, uma vez que na maioria dos casos é necessária intervenção humana. Neste artigo descrevemos uma solução segura e automática para este problema.

PALAVRAS CHAVE

Sistemas de pagamento electrónico, terminais POS, transferência segura de ficheiros, electronic founds transfers (EFT)

1 INTRODUÇÃO

Apesar de ao longo dos últimos anos terem surgido diversas propostas para sistemas de pagamento electrónico e de existirem actualmente diversos canais por onde efectuar este tipo de operações, é de esperar que a maioria das transacções electrónicas continuem a usar as redes privadas dos bancos. O sucesso dessas redes de pagamento deve-se entre outras coisas, ao suporte de um número bastante elevado de terminais, quer sejam *Automatted Teller Machines* (ATM) ou *Point of Sale* (POS), e à comodidade e divulgação do cartão de débito/crédito.

Um dos principais custos associados com a gestão e manutenção destas redes advém da necessidade de manter actualizadas as versões de *software* que correm nos POS e nos ATMs. Diversas causas explicam a necessidade de alterações ao *software* durante a vida útil do equipamento: a criação de novos serviços, *bugs* não detectados durante a certificação e a adopção de normas internacionais.

Actualmente, na maior parte dos países, a actualização de *software* nestes terminais continua a ser efectuada de forma manual. O processo de actualização começa com a necessidade de desenvolvimento duma nova versão do programa. O Operador do Sistema de Pagamentos (OSP), a entidade responsável pela gestão da rede de pagamentos, recebe a nova aplicação e certifica-a executando testes que visam garantir a qualidade do produto. Posteriormente o *software* é entregue a empresas de manutenção que se encarregam do resto do processo. Isto requer que um técnico se desloque ao local de instalação do equipamento e, em muitos casos, o desmonte para substituir ou actualizar o integrado onde reside a aplicação. Nos equipamentos mais recentes a actualização faz-se por *download* da aplicação dum PC mas também necessita que o técnico se desloque ao local de instalação.

Este processo tem vários problemas relacionados com a eficiência, segurança, custos e tempo. Observemos por exemplo a rede portuguesa de pagamentos electrónicos, a rede Multibanco [1]. Esta rede possui cerca de 140.000 terminais POS. Se cada actualização de *software* tiver um custo de 50 euros e demorar cerca de 20 minutos, a actualização completa da rede demoraria cerca de 46 mil horas homem e teria um custo associado na ordem dos 7 milhões de euros. Do ponto de vista da segurança este processo assenta na completa confiança nas entidades que prestam o serviço de actualização, e o OSP possui um controlo limitado sobre a operação.

As especificações relativas à Europay, Master Card & Visa (EMV) [5,6,7]. sugerem que os POS devam possuir capacidade de *download* de novas versões de aplicações, mas não disponibilizam uma solução concreta para a resolução do problema. Durante a nossa pesquisa, deparamo-nos com duas empresas fabricantes deste tipo de equipamentos que afirmavam possuir uma solução proprietária mas que não se mostraram disponíveis para a explicar ou debater, pelo que não possuímos qualquer informação sobre o seu protocolo.

Neste artigo descrevemos uma solução para a actualização segura de *software* em terminais POS que pode ser utilizada num ambiente bancário real. Os protocolos são relativamente genéricos, pelo que, estamos convencidos que as nossas ideias podem ser adoptadas nas redes que usem protocolos proprietários de *Electronic Founds Transactions* (EFT) ou nas que utilizam o standard ISO 8583 [16].

2 PROTOCOLO EFT

A arquitectura usada mais frequentemente para organizar um sistema de pagamentos electrónicos é a configuração *cliente-servidor*, onde os POS (ou ATMs) são o cliente e o Servidor do Sistema de Pagamentos (SSP) é o servidor localizado nas instalações do OSP. Os terminais podem utilizar diversas tecnologias de comunicação desde a linha telefónica com X.25 [4] até à transmissão sem fios via GSM [18].

Embora algumas das redes de pagamentos actuais tenham desenvolvido protocolos EFT proprietários, a maioria utiliza o standard descrito na norma ISO 8583 que especifica o formato e o conteúdo das mensagens para as transferências financeiras. Genericamente o protocolo EFT está organizado em mensagens do tipo *pedido-resposta* e cada par destas mensagens é designado de *transacção*. Os terminais tomam sempre a iniciativa da comunicação contactando o OSP.

Na norma ISO 8583, o primeiro campo das mensagens é designado por *Message Type Identifier* (MTI), e identifica univocamente o tipo de mensagem e o seu contexto. As mensagens são construídas usando um ou dois *bitmaps* que funcionam como referências dos elementos de dados presentes em cada mensagem. Alguns elementos são de dimensão fixa, outros de dimensão variável. A dimensão de cada elemento variável é determinada pelo seu prefixo funcionando como elementos *Tag Length Value* (TLV). Uma vez a mensagem chegada ao OSP, o SSP executa o serviço solicitado e responde com uma mensagem que informa o terminal do resultado do processamento.

Quer a mensagem de pedido como resposta encontram-se protegidas por um *Message Authentication Code* (MAC) [12,13], que assegura protecção contra ataques de integridade e de autenticidade. Cada POS partilha com o SSP um conjunto de chaves simétricas que são guardadas no módulo de segurança [3] do POS e que estão protegidas contra substituição ou observação.

Como exemplo, vamos descrever de seguida uma operação de compra. O comerciante usa o leitor do terminal para ler os dados da pista magnética ou do *chip* do cartão do cliente. Nestes dados estão incluídos diversos elementos que caracterizam o

cartão e a conta do cliente junto do seu banco [2]. O comerciante introduz a quantia da operação e o cliente introduz o seu *Personal Identification Number* (PIN) no *Pinpad* [14]. O POS constrói uma mensagem de pedido de autorização e envia-a para o SSP. O SSP verifica a integridade da mensagem e o conteúdo de cada um dos seus elementos. No caso de falha envia uma mensagem ao POS recusando o serviço. Em caso de sucesso o SSP envia uma mensagem ao banco do cliente a pedir o débito da quantia indicada na mensagem de pedido e envia também uma mensagem ao banco do comerciante a solicitar o crédito do mesmo montante. Se ambas as respostas indicarem a realização da operação com sucesso o SSP envia uma mensagem ao POS a indicar que a compra foi realizada.

3 TRANSFERÊNCIA SEGURA DE FICHEIROS

Tipicamente a transferência de ficheiros entre entidades bancárias é caracterizada pela troca entre sistemas de grandes volumes de informação num ambiente relativamente seguro. Em contraste, as transferências de ficheiros entre os terminais e o sistema de pagamentos é caracterizada pela transferência de pequenos volumes de informação num ambiente bastante mais inseguro do que o anterior.

A norma ISO/FDIS 15668 [17] aplica-se à transferência de diversos tipos de ficheiros no âmbito da actividade bancária, como por exemplo, software, transferência das transacções que foram realizadas em *offline*, dados técnicos relacionados com o sistema de pagamentos (ou banco tomador), dados aplicacionais (por exemplo, listas negras de cartões).

Em geral, os intervenientes numa transferência são: o originador, o emissor e o receptor. O originador é a entidade que produz os dados a serem transmitidos, o emissor a entidade responsável por transmitir os dados e por último o receptor a entidade a quem os dados se destinam. O processo de transferência depende essencialmente de duas importantes camadas, a de transporte, responsável pela transmissão dos dados, e a de segurança que providência os serviços de segurança.

Existem três grandes formas de transferência de ficheiros:

- Na *transferência de ficheiros protegidos* a camada de transporte não utiliza quais quer serviços de segurança, limita-se a fornecer os serviços de comunicações. Neste caso o ficheiro deve ser protegido antes da transferência, e a segurança é gerida pelo Originador e pelo Receptor. Não existe segurança adicionada ao nível das comunicações por parte do Emissor.
- Na *transferência segura de ficheiros* a segurança é tida em conta entre o Emissor e o Receptor. O Originador confia plenamente no Emissor. Neste caso a camada de transporte utiliza os serviços da camada de segurança e não é necessário que o ficheiro seja protegido antes da transferência.
- Na *transferência segura de ficheiros protegidos*, as funções de segurança são utilizadas pela camada aplicacional e pela camada de transporte. Por exemplo, o Originador cria um ficheiro, assina-o com a sua chave para assinatura e cifra o ficheiro com a chave secreta partilhada com o Receptor. Neste caso a principal preocupação vai no sentido de prevenir que alguém dentro da organização do Emissor possa ver o conteúdo do ficheiro, contudo, o Originador confia no processo de transferência na medida em que o Emissor efectuará a autenticação do Receptor e garantirá a integridade do ficheiro transmitido.

Uma vez que o protocolo que desenvolvemos para a actualização de aplicações em terminais POS seria aplicado em redes bancárias, decidimos usar uma arquitectura de

software para os terminais inspirada na norma ISO/FDIS 15668. O software foi assim decomposto em diferentes elementos aplicacionais, nomeadamente, no *loader,* no gestor de aplicações e na aplicação.

- O *loader* é um software seguro e de confiança, previamente carregado no terminal antes da transferência segura de ficheiros. É responsável pela implementação dos mecanismos de segurança para a transferência segura do gestor de aplicações e a sua execução segura;
- O *gestor de aplicações* é responsável pela implementação dos mecanismos de segurança para o carregamento de aplicações e a sua execução segura.
- As *aplicações* são usadas pelo terminal no âmbito dos pagamentos electrónicos e são constituídas por código objecto executável ou código objecto interpretável.

É necessário que a função de segurança implemente diversos serviços, nomeadamente, autenticação da origem da mensagem, autenticação do receptor da mensagem, integridade, não repudiação da origem e a não repudiação da recepção. A confidencialidade é necessária só para se garantir o segredo do código aplicacional transmitido.

Na transferência de ficheiros é possível utilizar-se criptografia simétrica ou assimétrica, mas, como a especificação EMV utiliza criptografia assimétrica, e uma vez que é expectável que esta norma seja adoptada em breve pela maior parte dos países europeus, optámos por basear o esquema de segurança da nossa solução também em criptografia assimétrica.

4 ARQUITECTURA DO SISTEMA DE ACTUALIZAÇÃO

A realização do sistema de actualização depende da capacidade que o OSP possui em manter registo dos vários elementos que caracterizam cada terminal POS, como, o identificador do terminal, a versão da aplicação em execução e o modelo do equipamento.

A tarefa de actualização começa pela identificação da necessidade de actualizar a aplicação do POS (ver Figura 1). É necessário que uma nova versão seja produzida pelo produtor de aplicações (Prod) e entregue ao OSP. O OSP certifica a aplicação pela execução de um número de testes que assegurem a conformidade do software com a especificação da rede de pagamentos. Quando o processo de certificação tiver terminado a aplicação é transferida para os Servidores de Descarga de Aplicações (SDA) que ficam encarregues de transferir as aplicações para os POS, libertando o SSP dessa tarefa.

Quando o SSP recebe uma mensagem dum POS deve concluir se esse terminal deve ou não ser actualizado. Esta decisão deve ser tomada cruzando diversa informação sobre o terminal, nomeadamente, o seu identificador (enviado na mensagem), versão da aplicação em execução e a existência duma aplicação para actualização já certificada. Se concluir que o terminal deve ser actualizado, na mensagem de resposta (quer o serviço solicitado seja ou não concluído com sucesso), o SSP informa o terminal de que este deve iniciar o processo de actualização. Quando o POS contactar novamente o SSP, pede-lhe os dados necessários para iniciar o processo -- os dados de comunicação do SDA e os elementos necessários à realização da transferência segura.

Quando o POS dispuser destes dados, liga-se ao SDA e inicia a transferência. Quando terminar, o POS informa o SSP do resultado da transferência para que da próxima vez possa decidir se deve proceder ou não à actualização do terminal.

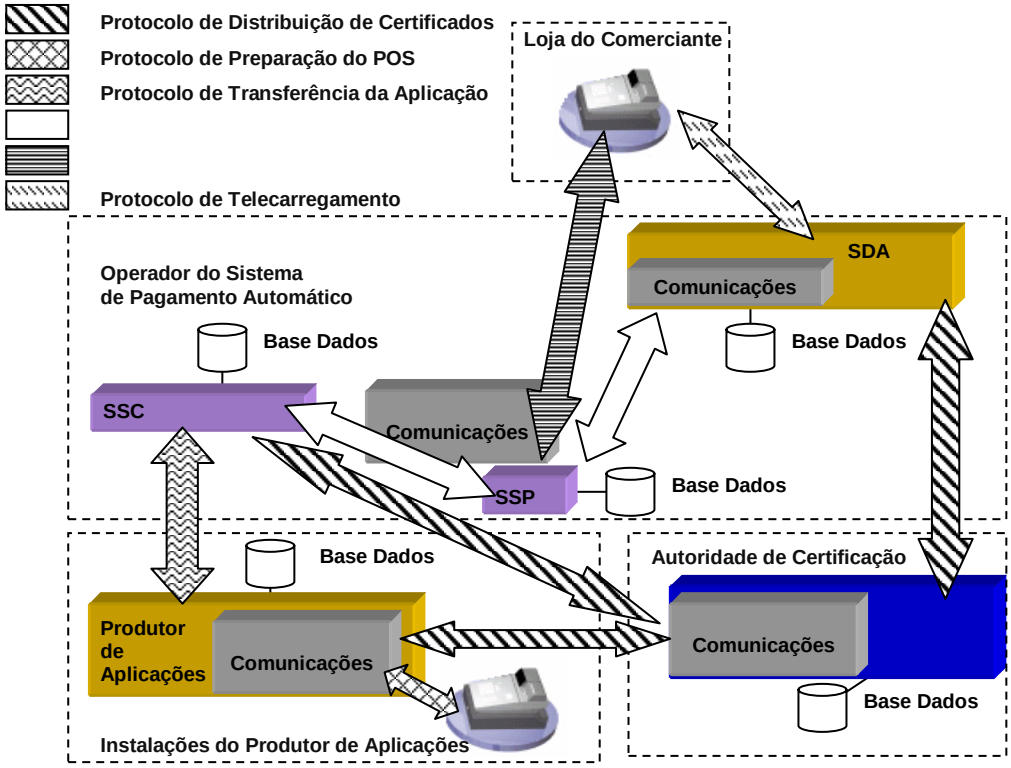

Figura 1 Arquitectura do Sistema de Actualização

A base do sistema de pagamentos é composta pelo terminal POS e pelo SSP, os elementos necessários à realização das transacções do protocolo EFT. Para que o sistema de actualização seja automático é necessário que o Prod possa enviar as aplicações directamente para o OSP. Para isso o Prod passa a dispor dum servidor (SProd) que envia directamente as aplicações para o Servidor do Serviço de Certificação (SSC) e recebe dele a notificação da conclusão da certificação ou das falhas às quais é necessário proceder à correcção. A comunicação entre o SProd e o SSC tem que ser segura para prevenir ataques que poderiam tentar, por exemplo, alterar as aplicações.

Os SDAs, como já referido anteriormente, foram adicionados ao sistema para libertar o SSP da tarefa de transferir a aplicação para o POS. Desta forma o desempenho da rede de pagamentos não é alterada pelo facto de passar a dispor dum serviço de actualização de aplicações seguro e automático. Os SDA recebem e armazenam as aplicações certificadas e enviadas pelo do SSC.

O papel da Autoridade de Certificação (CA) é da maior importância, uma vez que gere os certificados de chave pública dos vários componentes do sistema de actualização. A CA executa várias funções: autentica de forma idónea e segura as entidades que requerem a emissão de novos certificados (conforme definido pela sua política de segurança); gere e mantêm a *Certification Revocation List* (CRL) sempre que é necessário cancelar um certificado (e.g. a chave secreta foi comprometida); também é responsável pelo arquivo de todos os dados gerados e responsável também pela destruição segura e efectiva de todos os dados desnecessários.

5 PROTOCOLO DE ACTUALIZAÇÃO

O protocolo de actualização de aplicações EFT é implementado por um conjunto de sub-protocolos, cada um deles responsável por uma tarefa específica (ver Figura 1). Uma vez que a maior parte das mensagens são protegidas usando o mesmo mecanismo,

damos aqui uma descrição genérica (a única excepção são as mensagens do protocolo EFT que são protegidas usando o esquema baseado em MAC, conforme referido na secção 2). Sempre que um componente ***A*** envia uma mensagem ***data*** a um componente ***B***, constrói uma mensagem com ***<data, E(KrA, Hash(data)), CERT_KuA>*** onde ***E(KrA, Hash(data))*** é a assinatura e significa cifrar o *hash* de ***data*** (isto é ***Hash(data)***) com a chave privada ***Kr*** de ***A***. A mensagem também inclui todos os certificados com as chaves públicas necessários à validação da assinatura, que neste caso é ***CERT_KuA***. Uma vez recebida a mensagem, o receptor efectua as seguintes verificações: ***CERT_KuA*** é validado usando a chave pública da CA que está armazenada em ***CERT_KuCA***; a assinatura é verificada decifrando ***E(KrA, Hash(data))*** com ***KuA***, e comparando o resultado com o hash de ***data***. Se forem iguais então a mensagem está correcta.

Protocolo de Distribuição de Certificados Este protocolo distribui as chaves criptográficas e os certificados produzidos pela CA. Estas chaves e certificados são usados para assegurar a autenticação e não repudiação da informação trocada entre os diversos componentes da arquitectura. O SDA, SSC e o SProd devem executar as seguintes acções:

- Obter uma cópia do certificado da chave pública da CA (CERT_KuCA);
- Gerar um par de chaves assimétricas, pública e privada, e guardá-las de forma segura ((KuXXX, KrXXX) onde XXX será: SDA, SSC ou SProd);
- Solicitar à CA a criação do certificado contendo a chave pública (CERT_KuXXX onde XXX será: SDA, SSC ou SProd).

O certificado com a chave pública da CA tem que ser distribuído pelos componentes da arquitectura de forma segura uma vez que todas as verificações de assinatura dependem dela.

Tipicamente a criação dum certificado requer intervenção humana devido a alguns passos de autenticação/autorização. A política de segurança pode impor, por exemplo, que exista uma autorização explícita do OSP antes de se poder criar um certificado. Este tipo de autorização é interessante porque limita as entidades habilitadas a produzir assinaturas para o software a actualizar.

Protocolo de Preparação do POS O fabricante do POS tem que executar algumas acções de preparação do POS antes de o poder enviar para um estabelecimento comercial. Estas operações incluem a instalação do software de base, o *loader*, o gestor de aplicações e o armazenamento seguro das seguintes chaves:

- par de chaves assimétricas do POS (KuPOS, KrPOS);
- Uma cópia do certificado com a chave pública do POS (CERT_KuPOS). Este certificado pode ser assinado com a chave privada do Prod (em vez da CA);
- Uma cópia do certificado com a chave pública do Prod (CERT_KuProd);
- Uma cópia do certificado com a chave pública da CA (CERT_KuCA).

Cada terminal possui um par de chaves distinto que assegura que estando um POS comprometido não implica o compromisso de toda a rede. Para reduzir o risco de exposição da chave, as chaves privadas dos POS devem ser destruídas pelo Prod depois de armazenadas no módulo de segurança do POS. O Prod é responsável pela geração do certificado com a chave pública do POS. Esta opção é interessante do ponto de vista económico e de eficiência porque é mais simples produzir um certificado localmente (sem a intervenção da CA) tendo uma política de segurança mais relaxada. Contudo, a validação da assinatura do POS requer a posse de ambos os certificados, do Prod e da CA (CERT_KuCA verifica CERT_KuProd, e CERT_KuProd verifica CERT_KuPOS).

A inclusão do certificado do fabricante é usado para restringir quem pode criar novas versões de software – um POS só aceita actualizações assinadas pelo mesmo fabricante. À primeira vista pode parecer uma limitação desnecessária do protocolo. Contudo, tem importantes implicações ao nível da segurança porque previne certo tipo de ataques. Por exemplo, um falso SSP é incapaz de substituir uma nova actualização por uma outra maliciosa, assinada por um Prod amigo (mas também malicioso).

Protocolo de Transferência da Aplicação Este protocolo define como as novas versões das aplicações EFT e os relatórios de certificação são trocados entre os Prod e o SSC. Sempre que uma nova aplicação é produzida, o Prod envia uma cópia do código da aplicação assinada com a sua chave privada para aprovação. Depois, a aplicação é sujeita a um conjunto de testes para garantir que funciona conforme esperado. Seguidamente, o SSC envia ao Prod um relatório assinado reportando se a nova versão da aplicação foi aceite ou se não passou nos testes de certificação (no último caso também inclui uma lista contendo a descrição dos testes que falharam). Do ponto de vista da segurança, o processo de certificação é importante porque permite detectar certo tipo de ataques que podem ser executados por um Prod adversário. Basicamente, com um conjunto de testes exaustivos, pode-se prevenir a inserção nos POS da rede de software com problemas. Se o SSC estiver sob controlo dum adversário, ainda que momentaneamente, ela (ou ele) não será capaz de produzir aplicações correctamente assinadas, quanto muito poderia tentar substituir a nova aplicação por outra mais antiga (não esquecer que um POS só aceita actualizações do próprio fabricante). Contudo, este ataque pode ser ultrapassado se for adoptada a regra de que só poderão ocorrer actualizações de aplicações que possuam versões superiores.

Protocolo de Gestão Interna Este protocolo reúne todas as acções internas ao OSP necessárias para realizar as actualizações de software. Basicamente deve executar duas acções:

- *Transferência do código para os SDAs*: O SSC envia a cada SDA uma mensagem assinada com a cópia do código e a lista dos modelos de POS que devem ser actualizados. Depois de armazenar o código, o SDA está pronto a receber pedidos de actualização dos POS;
- *Notificar o SSP sobre a actualização*: O SSC envia uma mensagem assinada ao SSP contendo a lista dos modelos de POS que devem ser actualizados, o número de versão do código, e a assinatura do código efectuada pelo Prod (i.e, baseado na KrProd). Esta informação é guardada na base de dados do SSP.

Durante o funcionamento normal, o POS apenas comunica com o SSP. Pelo que, tem que ser o SSP a informar ao terminal que deve proceder ao início da actualização duma nova versão da aplicação EFT.

Protocolo EFT No protocolo EFT standard, desencadeado por uma acção do comerciante, é o POS que tem a iniciativa da comunicação, começando pelo envio do pedido de serviço ao SSP. O SSP na mensagem de resposta pode ordenar ao POS a execução de um determinado serviço, através da inclusão do MTI desse serviço na resposta. Assim temos que introduzir quatro novas transacções EFT para informar o POS sobre novas actualizações e actividades de gestão. As novas transacções EFT são:

- *Transacção de Início de Actualização* (IA): indica que uma nova actualização está disponível e fornece a seguinte informação: dados de configuração sobre o SDA que deve ser contactado (e.g. parâmetros de comunicação), número de versão do código, e a assinatura do código da aplicação efectuada pelo Prod;

- *Transacção de Fim de Actualização* (FA): serve para manter o SSP actualizado quanto às actualizações já realizadas. Depois de completar a actualização e a execução da nova aplicação, o POS contacta o SSP para indicar que a actualização terminou com sucesso (ou que ocorreu um erro, neste caso essa informação é enviada pela aplicação que não chegou a ser substituída);
- *Transacção de Versões de Chaves*: O POS envia as versões de chaves (e certificados) que estão armazenadas no seu módulo de segurança;
- *Transacção de Actualização de Chaves*: O SSP pode actualizar as chaves armazenadas no POS. Esta transacção tem que ser efectuada com algumas precauções porque um SSP malicioso pode usá-la para comprometer toda a rede.

Estas novas transacções são incluídas no protocolo EFT standard, pelo que partilham da mesma infra-estrutura de segurança onde as mensagens são protegidas com um esquema baseado em MAC.

Protocolo de Transferência Este protocolo transfere a aplicação para o POS através do SDA. Um exemplo desta transferência pode ser observado na Figura 2. Para realizar esta tarefa existem três diferentes transacções que devem ser executadas entre o POS e o SDA.

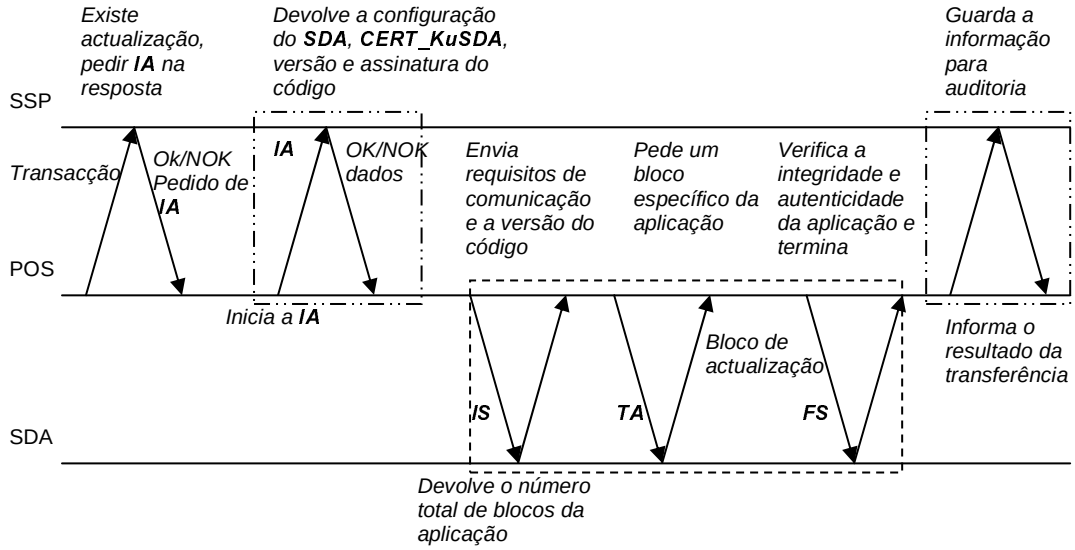

Figura 2: Exemplo de Transferência

- *Inicio de Sessão de Transmissão* (IS): O POS envia uma mensagem assinada ao SDA fornecendo elementos sobre a versão do programa pretendido e os seus parâmetros de comunicação, tais como, máximo tamanho do bloco (MTB) de dados e a lista dos algoritmos de compressão suportados. A resposta enviada pelo SDA inclui o tamanho total da aplicação, o número total de MTB que vão ser enviados (a aplicação pode ter que ser fragmentada se o seu tamanho ultrapassar o tamanho do MTB);
- *Transferência da Aplicação* (TA): em cada par de mensagens, o POS pede um bloco de dados da aplicação específico e o SDA transmite esse bloco. No caso de falha, o POS pode sempre reiniciar o processo a partir do último bloco correctamente recebido;

- *Fim de Sessão de Transmissão* (FS): depois da recepção de todos os blocos da aplicação, o POS confirma a origem da aplicação usando a assinatura criada pelo Prod (que foi fornecida pelo SSP). Depois, usa esta transacção para informar o SDA sobre o sucesso (ou falha) da transferência. Depois coloca em execução a nova aplicação.

Existem diversas causas que podem interromper a transferência da aplicação. Por exemplo, pode ocorrer um atraso anormal das comunicações que pode resultar em *timeout* quer do lado do POS como do lado do SDA; ou a sessão pode ter sido cancelada porque existe uma aplicação mais recente para ser actualizada. Nestes casos, dependendo da razão, o POS deve ser informado, pela utilização de diferentes códigos de erro, se pode continuar a transferência em curso ou começar do início.

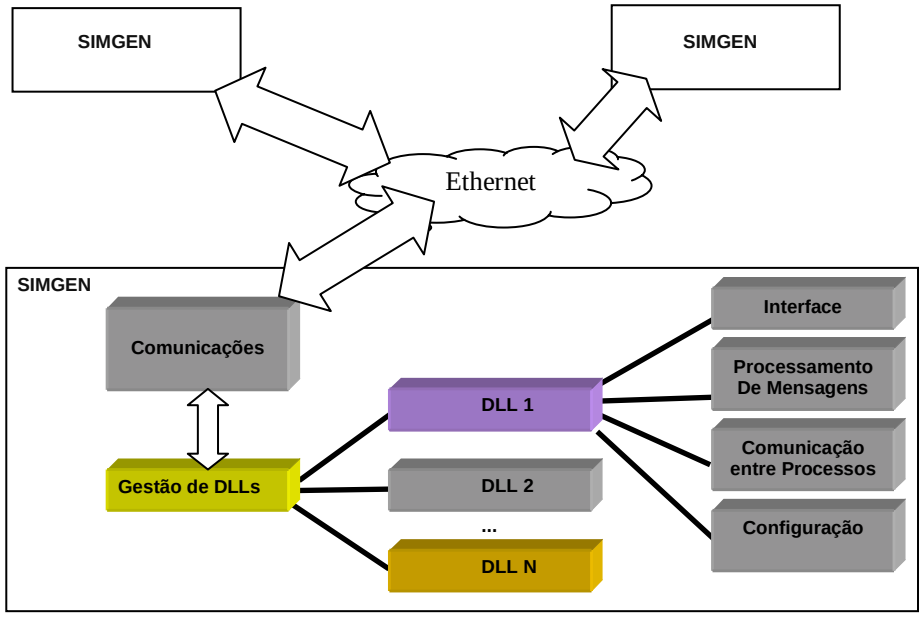

Figura 3 : Estrutura da SIMGEN

Para ultrapassar os casos em que a existência de alguma anomalia impeça um POS de iniciar de forma automática o processo de transferência, deve ser disponibilizada na interface do operador do equipamento uma função que permita desencadear o processo de forma manual, isto é, ordenar o POS a comunicar com o SDA para iniciar o processo de transferência. Nesse caso, os diversos parâmetros necessários ao início do processo, que normalmente são enviados pelo SSP na mensagem IA, deverão ser introduzidos de forma manual por intermédio dessa interface.

No pior dos casos o processo de actualização decorre como até hoje, ou seja, com a intervenção do técnico.

6 PROTÓTIPO DO PROTOCOLO E AVALIAÇÃO

Tendo sido impossível obter em tempo útil os meios necessários para efectuar a implementação da solução proposta, optou-se pela criação duma simulação dum sistema de pagamentos que realiza a actualização de software para POS.

O simulador contém uma representação de cada um dos componentes do sistema, nomeadamente o POS, SSP, SDA, SSC, SProd e CA, faz a gestão da infra-estrutura de comunicações TCP/IP e permite realizar a monitorização das mensagens trocadas.

A aplicação de base é suficientemente genérica para ser reutilizada por todos os componentes, mas também bastante flexível para suprir as necessidades específicas de cada um deles. A sua flexibilidade advém do uso de DLLs que se registam na aplicação base e que definem o processamento específico.

Descrevemos seguidamente duas das características de funcionamento que a fazem adaptar à construção de aplicações do tipo Servidor, Cliente ou ambos simultaneamente. Em primeiro lugar permite uma espera activa de mensagens sobre um porto TCP. Neste caso ao receber uma mensagem, esta aplicação pesquisa todas as DLLs registadas até determinar a que possui capacidade para efectuar o processamento da mensagem recebida. Em segundo, para a implementação da parte cliente, esta aplicação disponibiliza às DLLs um conjunto de funções que lhes permitem abrir um canal de comunicações, tomar a iniciativa de enviar mensagens e esperar pela respectiva resposta.

Genericamente esta aplicação, SIMGEN, possui na sua estrutura, um gestor de comunicações e um gestor de DLLs que permite a adição e remoção de DLLs (ver Figura 3). Todas as DLLs compatíveis com esta aplicação devem possuir um conjunto de funções de interface que permite à SIMGEN reconhecer a sua compatibilidade e comunicar da mesma forma com todas elas. Esse conjunto de funções disponibiliza:

- A apresentação duma interface humana adequada;
- Passagem de eventos;
- Envio e recepção de mensagens;
- Comunicação entre DLLs;
- Guardar e restaurar parâmetros de comunicação.

Um ficheiro de configuração assegura no momento de arranque da SIMGEN que todas as DLLs que formam um simulador são carregadas, e que todos os ficheiros de configuração pertencentes a cada uma das DLLs são também lidos. Combinando diversas DLLs obtém-se assim a simulação dum componente específico do sistema.

Para efeitos de demonstração e avaliação de resultados optou-se por implementar uma CA construída com base na aplicação SIMGEN. Nesta CA, os pedidos de emissão e revogação de certificados são efectuados através duma interface humana onde o utilizador regista:

- Data de início e fim de validade – O certificado é válido entre as duas datas que compõem este período;
- Titular – Identifica o titular do certificado;
- Chave Pública – A chave pública do titular do certificado.

Os pedidos de cópias de certificados e as respostas contendo o correspondente certificado são transmitidos via TCP/IP, ou seja, qualquer entidade pode enviar uma mensagem à CA fazendo o pedido de cópia do certificado, identificando o titular de cuja cópia do certificado pretende receber.

Na implementação do POS ficou disponível uma interface que permite a definição de parâmetros genéricos relativos ao protocolo EFT, nomeadamente, os dados de comunicação do SSP e a chave assimétrica para o cálculo do MAC. As chaves pública e privada do POS são geradas pelo SProd e através dum ficheiro colocados em conjunto com os certificados de chave pública no POS. No POS foi construída uma interface que permite a parametrização e monitorização do processo de transferência. Através dela é possível definir os seguintes parâmetros de actualização:

- Velocidade do processo de transferência;
- Tamanho máximo do bloco de dados;
- Taxa de erros, i.e., percentagem de blocos errados durante a transmissão;

- Habilitar / não habilitar o cálculo e verificação do MAC das mensagens;
- Habilitar / não habilitar a geração e verificação das assinaturas das mensagens.

A mesma interface também permite a monitorização de diversos dados do processo de transferência, nomeadamente:

- identificador do SDA designado para transferir o ficheiro;
- A chave pública do SDA recebida do SSP;
- Os dados de comunicação do SDA designado para proceder à transferência;
- Tamanho total da aplicação;
- Assinatura da aplicação gerada pelo SProd;
- Número total de blocos necessários à transmissão total do ficheiro;
- Número do bloco em transmissão;
- Número de erros do processo de transmissão;
- Velocidade de transmissão.

O ambiente de simulação foi constituído por uma rede Ethernet a funcionar a 10MBits/s que liga dois PCs, PC1 e PC2. O PC1 foi equipado com um processador AMD Duron @ 1,4GHz com 128MBytes e o PC2 equipado com um processador Pentium Celeron @700MHz também com 128MBytes.

Nesta implementação foi usada para função de hash [8,11] e como algoritmo de cifra o RSA [9,10,15] com chaves de 1024 bits. Foi estabelecido que os blocos de dados durante a transferência do ficheiro teriam um tamanho máximo de 2048 Bytes.

Através da interface do SSP inseriram-se na base de dados do OSP os dados do novo POS: identificador, modelo e versão de *software*. Em resultado dessa acção o SSP gerou a chave de MAC que foi inserida no POS. A partir deste momento passa a ser possível realizar transacções EFT entre o POS e o SSP.

O protocolo de Transferência da Aplicação é desencadeado na interface do SProd indicando o ficheiro que contem a nova aplicação e fazendo-o gerar um ficheiro contendo a assinatura do código da aplicação. Estes dois ficheiros são inseridos no OSP através da interface do SSC caracterizando a versão do *software* e os modelos de POS a que se destinam. Nesta implementação o SDA tem acesso ao ficheiro que contem a aplicação uma vez que tanto o SDA como o SSC estão na mesma máquina.

As medições efectuadas foram dirigidas em particular para as transacções de actualização entre o POS e o SDA. O desempenho do protocolo foi obtido para duas situações distintas, a transferência sem e com geração/verificação de assinaturas. Para ambas as situações registaram-se os tempos de transferência de ficheiros de transporte de aplicações com dimensões compreendidas entre 500Kbytes e 2 Mbytes. Admitindo que o ficheiro de transporte pode conter uma aplicação em formato comprimido a aplicação na realidade pode ser bastante superior a esta medida.

As velocidades de transferência usadas nos ensaios variaram entre 9,6Kbit/s e 10Mbit/s. Estas velocidades foram escolhidas em função da sua utilização. A velocidade de 9,6Kbit/s é mais utilizada na grande maioria dos terminais POS ligados ao SSP por meio de linha telefónica. A escolha da velocidade de 10MBit/s disponibiliza indicadores sobre o comportamento do protocolo à medida que se começam a utilizar novas tecnologias de transmissão.

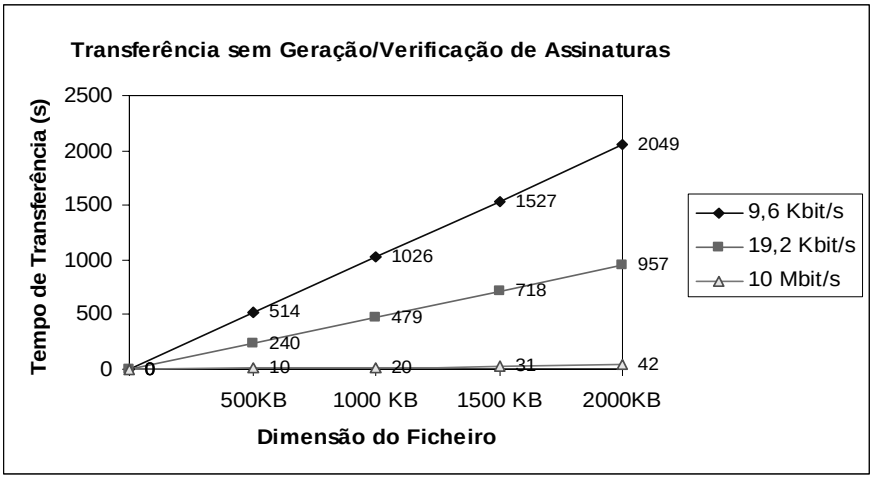

Figura 4 : Transferência sem Geração/Actualização de Assinaturas

Os resultados apresentados na Figura 4 correspondem ao desempenho do protocolo de transmissão não havendo geração ou verificação das assinaturas. A análise dos resultados apresentados permite-nos concluir que a baixas velocidades de transmissão existe uma proporcionalidade inversa entre a velocidade de transmissão e o tempo de transmissão. Esse fenómeno desaparece quando a velocidade de transmissão aumenta, em particular a 10Mbit/s, devido ao tempo de processamento das mensagens. Existe um outro fenómeno da mesma natureza entre os tempos de transmissão e o tamanho do ficheiro transmitido uma vez fixada a velocidade. Assiste-se a uma proporcionalidade directa, ficheiros com o dobro do tamanho demoram cerca do dobro do tempo a serem transmitidos.

Ao introduzir-se a capacidade de geração e verificação de assinaturas durante o processo de transferência verifica-se uma degradação do desempenho do sistema, conforme se mostra na Figura 5.

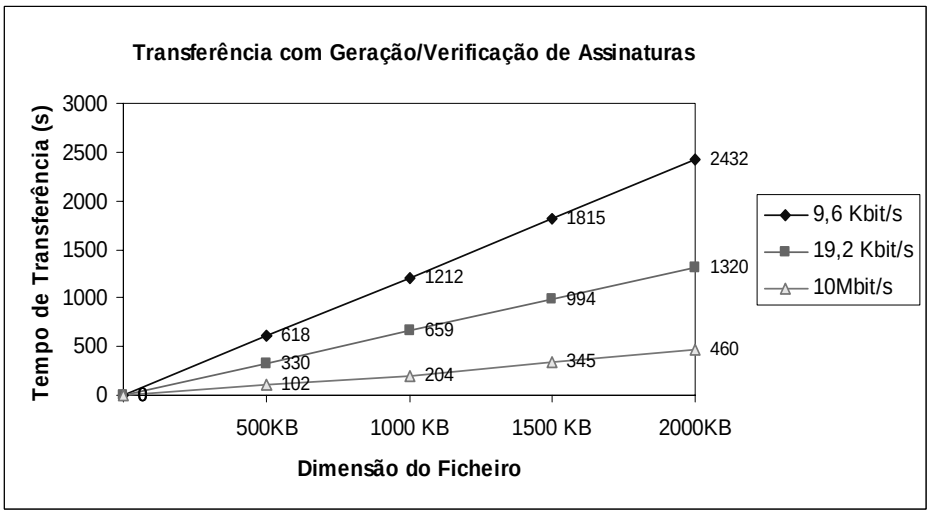

Figura 5 : Transferência com Geração/Verificação de Assinatura

Dentro da mesma velocidade a diminuição de desempenho é constante, como se mostra na Figura 6. Este facto reforça a ideia de que a degradação decorrente se deve à mesma perturbação, a geração e verificação de assinaturas.

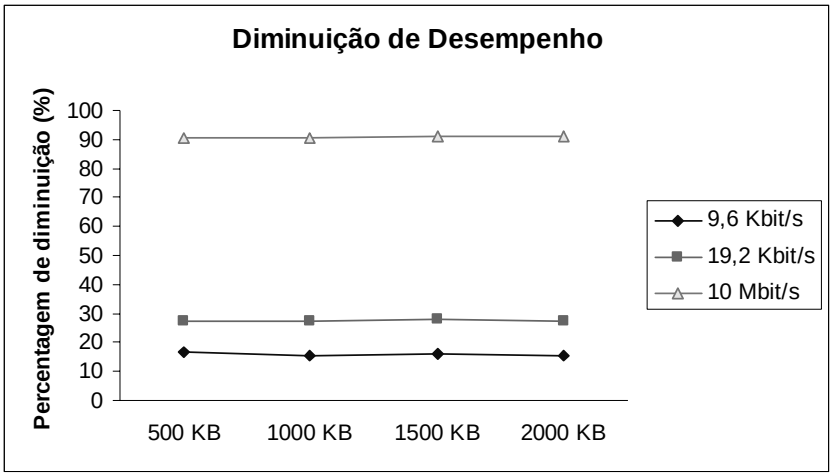

Figura 6 : Diminuição de Desempenho do Processo de Transferência

Considerando o pior caso nos resultados obtidos, conclui-se que a uma velocidade de 9,6 Kbit/s a transmissão da aplicação demoraria cerca de 40 minutos a concluir.

Apesar da obviedade das conclusões, estas assumem uma verdadeira importância quando observadas do ponto de vista da viabilidade da implementação da solução, pois os valores apresentados permitem constatar que:

- Durante o normal funcionamento do POS, haverá em muitos casos, períodos de inactividade da ordem de grandeza dos valores apresentados. Pelo que a actualização pode ser efectuada sem perturbar o funcionamento do estabelecimento onde o POS está instalado. O custo de comunicações associado a este intervalo de tempo é suportável e muito mais reduzido do que uma intervenção técnica realizada com a presença humana;
- Durante o tempo de transmissão é provável que existam interrupções derivados por um lado a erros de transmissão, por outro à necessidade de realizar operações de cliente. Estes dois factos reforçam a necessidade do protocolo de transferência possuir mecanismos de recuperação que permitam construir a aplicação em blocos aproveitando os blocos de dados já transferidos e os intervalos de inactividade.

Estes resultados também são importantes para dimensionar o sistema nas suas várias componentes, ajudando a analisar outras variáveis que influenciam o processo de actualização como um todo, só para dar alguns exemplos:

- O número de SDAs necessários para um processo de actualização de grande escala e o número de transferências que podem decorrer em simultâneo.
- O processo de gestão de actualizações a implementar no SSP face às restrições anteriores.

7 CONCLUSÕES

Este artigo descreve uma solução para a actualização segura e automática de aplicações EFT nos terminais POS. Esta solução requer a introdução de alguns componentes na arquitectura actual do sistema de pagamentos automático, em particular, um servidor do sistema de certificação onde os produtores de aplicações podem enviar e validar as aplicações para actualização, e um conjunto de servidores de distribuição que interagem com os POS durante a actualização. Foi disponibilizado um protocolo para gerir os diversos passos da actualização da aplicação, desde o momento

em que é produzida até que é instalada no POS. Esta solução foi implementada e avaliada numa rede de PCs.

Bibliografia

[1] K. Bohle, M. Rader, U. Riehm, *"Electronic Payment Systems in European Countries, Country Synthesis Report"*, Istitut fur Technikfolgenabschatzung und Systemankyse, 1999.

[2] American National Institute, *"Proposed American National Standard X4.16, Magnetic Stripe Encoding for Financial Transaction Cards"*, 1980.

[3] C. Koç, D. Naccache, C. Paar, *"Cryptographic Hardware and Embedded Systems – CHES"*, Springer-Verlag, 2001.

[4] U. Black, *"X.25 and Related Protocols"*, IEEE Computer Society Press, 1991.

[5] EMVCo, LLC ("EMVCo"), *"EMV2000 Integrated Circuit Card Specification for Payment Systems, Book 1 Application Independent ICC to Terminal Interface Requirements"*, EMVCo 2000.

[6] EMVCo, LLC ("EMVCo"), *"EMV2000 Integrated Circuit Card Specification for Payment Systems, Book 2 Security and Key Management"* , EMVCo 2000.

[7] EMVCo, LLC ("EMVCo"), *"EMV2000 Integrated Circuit Card Specification for Pt Systems, Book 4 Cardholder, Attendant, and Acquirer Interface Requirements"*, EMVCo 2000.

[8] U.S Department of Commerce, *"FIPS PUB 180-1 Secure Hash Standard"*, 1995.

[9] RSA Laboratories, *"PKCS#1: RSA Encryption Standard. Version v2.1"*, 2002.

[10] RSA Laboratories, *"PKCS#7: Cryptographic Message Syntax Standard. Version 1.5"*, 1993.

[11] National Institute of Standards and Technology, *"FIPS PUB 180-1 Secure Hash Standard"*, 1995.

[12] International Standard, *"ISO/IEC 9797 Information technology - Security techniques - Message Authentication Codes (MACs)"*, 1999.

[13] International Standard, *"ISO/IEC 9807 Banking and related financial services - Requirements for message authentication (retail)"*, 1991.

[14] American National Standard, *"ANSI X9.8 American National Standard for Pin Management and Security"*, 1982.

[15] American National Standard for Financial Services, *"ANSI X9.31 Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)"*, 1998.

[16] International Standard, *"ISO 8583, Financial transaction card originated messages – Interchange message specification"*, 2003.

[17] International Standard, *"ISO 15668, Banking Secure File Transfer (retail)"*, 1997.

[18] European Telecommunication Standards Institute, *"GSM Technical Specification"*, 1995.