Acordo de Chave com Autenticação sem Certificado Digital

Denise Goya, Cleber Okida, Routo Terada
Departmento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo
Email: {dhgoya, cleberok, rt}@ime.usp.br

***Resumo*—Sob o modelo de criptografia de chave pública sem certificado (*certificateless*), protocolos de acordo de chave estabelecem chaves de sessão, com garantia de autenticidade dos participantes, sem uso de certificados digitais. Propõe-se, neste artigo, novo protocolo que pode ser demonstrado seguro sob um forte modelo de segurança. Resultados experimentais de implementação de uma aplicação sobre dispositivos móveis atestam maior eficiência da proposta perante protocolos anteriores.**

I. Introdução

Protocolo de acordo de chaves é uma ferramenta primordial para o desenvolvimento de soluções de segurança que requerem sigilo das comunicações. Com protocolos desse tipo, é possível que participantes em uma rede de comunicação combinem chaves de sessão, usando canais públicos. Protocolos de acordo de chaves com autenticação fazem o mesmo, porém com garantia de autenticidade dos participantes.

Sob o modelo de criptografia de chave pública, protocolos de acordo de chaves com autenticação funcionam sob a premissa de que a chave pública é previamente certificada, o que em implementações tradicionais ocorre com a implantação de uma infraestrutura de chaves públicas (ICP), com uso de certificados digitais.

Em 2003, Al-Riyami e Paterson [1] propuseram um modelo alternativo de chave pública que dispensa a necessidade de certificados e ICP. Nesse modelo, conhecido como *certificateless*, cada usuário cria um par de chaves (pública e privada, tal qual no modelo convencional). Adicionalmente, a autoridade do sistema, chamada Centro de Geração de Chaves ou KGC (Key Generation Centre), fornece a cada usuário registrado uma chave secreta parcial, calculada a partir da chave secreta mestra do KGC e da identidade do usuário. Essa chave secreta parcial é um componente da chave secreta (completa, de longa duração) e estabelece um vínculo entre o usuário e o sistema. A certificação da chave pública ocorre implicitamente com a execução dos protocolos.

Comparado ao modelo convencional em que é requerida uma ICP para geração, distribuição, validação e revogação de certificados, o modelo de Al-Riyami e Paterson simplifica os processos, requer uma infraestrutura mais simples e potencialmente reduz custos operacionais e computacionais. É uma alternativa promissora para aplicações que envolvem o uso de dispositivos como *smart cards* ou *smartphones*, que possam ser preparados previamente para se comunicarem com servidores. Mais especificamente, os protocolos de acordo de chaves com autenticação no modelo de criptografia de chave pública sem certificados (CL-AKA) permitem que tais dispositivos e um servidor se autentiquem mutuamente e estabeleçam chaves de sessão não falsificáveis. Essas chaves de sessão podem ser adotadas como chaves secretas em cifras de bloco, para garantir sigilo nas comunicações.

Um problema que detectamos nessa classe de protocolos é a carência de opções eficientes, que levam em conta a reduzida capacidade de memória dos dispositivos móveis e que ao mesmo tempo ofereçam elevado nível de segurança.

Trabalhos Relacionados

Existe uma quantidade relativamente grande e recente de protocolos de acordo de chaves sob o modelo tradicional com ICP, dos quais podemos citar o MQV [2], que recebeu sucessivos aprimoramentos e posteriormente foi padronizado [3] e chegar à versão HMQV [4]. Uma compilação abrangente de protocolos dessa classe é encontrada em [5]. Para uma comparação grosseira de tempo de processamento do MQV (ou HMQV) com um protocolo CL-AKA seria necessário acrescentar ao MQV os procedimentos para verificação da assinatura do certificado (eventualmente, obter o certificado antes e verificar as assinaturas no caminho de certificação até a raiz e conferir se o certificado foi revogado). A assinatura e sua verificação tipicamente são realizadas com o RSA ou DSA [6].

Um outro modelo de criptografia de chave pública que dispensa o uso de certificados digitais é o baseado em identidade [7]. Chen, Cheng e Smart apresentaram uma análise de quase vinte protocolos de acordo de chaves no modelo baseado em identidade [8]. Há duas situações de risco para protocolos de acordo de chaves baseados em identidade:

1) quando a autoridade do sistema age mal intencionalmente e faz uso de seu conhecimento privilegiado das chaves secretas de todos os usuários para falsificar uma chave;
2) a chave mestra secreta do sistema é comprometida.

Nesses dois casos, o vazamento do segredo temporário de uma sessão é suficiente para a recuperação da chave de sessão. Uma fonte de vazamento de segredos temporários

pode ser, por exemplo, funções geradoras de números pseudo-aleatórios fracas ou armazenamento de forma insegura de valores aleatórios previamente calculados (este último caso é prática corrente especialmente em aplicações que envolvem dispositivos de menor capacidade de processamento).

Com o uso do modelo sem certificados, as consequências dos riscos acima são atenuadas, pois cada usuário possui três segredos (e não dois, como no caso baseado em identidade):

1) o valor secreto do usuário, o qual gera a sua chave pública;
2) a chave parcial secreta (calculada pelo KGC e compartilhada com o usuário);
3) o segredo temporário de sessão (um número pseudo-aleatório, sorteado para o cálculo da chave de sessão).

Um KGC mal intencionado atuando sobre um protocolo de acordo de chaves sem certificado, ou um adversário que tenha acesso à chave mestra secreta, precisa corromper outros dois segredos adicionais: um permanente e outro temporário, ambos em posse exclusiva do usuário. Sob um modelo de segurança adequado, um protocolo CL-AKA permanece seguro se no máximo dois dos três segredos de cada usuário for corrompido. Portanto, comparativamente aos protocolos de acordo de chaves baseados em identidade, um CL-AKA apresenta nível de segurança maior.

Diversos protocolos CL-AKA podem ser encontrados na literatura. Um estudo realizado por Swanson [9], [10] mostra que todos os protocolos propostos até então eram inseguros e, por esse motivo, a autora propôs um novo modelo de segurança adequado ao caso sem certificados.

Em [11], Lippold, Boyd e González Nieto (LBG) aprimoram o modelo de Swanson e apresentam o primeiro protocolo CL-AKA demonstrado seguro sob um modelo forte de segurança. O protocolo LBG se mantém seguro enquanto cada usuário detiver ao menos um segredo não comprometido. Em particular o KGC, que conhece as chaves parciais secretas de todos os usuários, mesmo se tiver acesso aos segredos temporários, ainda terá que comprometer o valor secreto que gerou a chave pública do usuário para ser capaz de recuperar uma chave de sessão. A maior desvantagem do protocolo LBG é sua complexidade, o que o torna pouco eficiente computacionalmente.

Outro protocolo CL-AKA de que temos conhecimento ter sido demonstrado seguro sob o modelo de [11] foi apresentado por Goya, Okida e Terada (GOT) em [12]. Trata-se de uma versão otimizada do protocolo antecessor LBG, conferindo maior velocidade, sob as mesmas condições de segurança. Julgamos, entretanto, que o protocolo GOT é ainda pouco eficiente para uso em aplicações que envolvem dispositivos com menor capacidade de armazenamento.

Nossas Contribuições

Neste artigo, apresentamos um novo protocolo de acordo de chaves sob o modelo de criptografia de chave pública sem certificado, que pode ser demonstrado seguro sob um forte modelo de segurança. Comparado aos protocolos LBG e GOT, nossa proposta é computacionalmente mais eficiente, em níveis elevados de segurança. E mostra-se como uma opção para cenários em que há maior restrição de uso de memória.

Também descrevemos um estudo de caso relacionado a uma aplicação de informática em saúde, envolvendo telefones celulares e tráfego de prontuários médicos. Mostramos como fizemos uso de protocolos do tipo CL-AKA para cobrir requisitos de autenticação de servidor e de celulares, e estabelecer chaves de sessão para trafegar mensagens com garantia de sigilo. Apresentamos resultados experimentais que atestam a maior eficiência de nossa proposta.

Organização do Trabalho

Na Seção II, apresentamos conceitos e notação necessários para a compreensão dos protocolos. Nas Seções III, IV e V descrevemos nosso protocolo, sua consistência, suas propriedades e demonstração de segurança. Na Seção VI, comparamos nossa proposta com protocolos relacionados. Em seguida, na Seção VII, mostramos um estudo de caso e resultados experimentais. Por fim, apresentamos conclusões e trabalhos futuros.

II. Conceitos Preliminares

Quando escrevemos $x \xleftarrow{\$} \mathbb{Z}_q^*$, significa que a variável x recebe o resultado de um sorteio aleatório e independente sobre o conjunto dos inteiros de $1, \cdots, q-1$.

1) Emparelhamento Bilinear: Os protocolos aqui discutidos fazem uso de uma função definida sobre curvas elípticas, com propriedades específicas. Sejam G e G_T grupos cíclicos de ordem prima q. Um emparelhamento bilinear admissível [13], é um mapeamento $e : G \times G \rightarrow G_T$, que satisfaz as seguintes condições:

- **Bilinearidade**: para todo $P, Q \in G$ e $a, b \in \mathbb{Z}_q$, $e(aP, bQ) = e(P, Q)^{ab} = e(abP, Q) = e(P, abQ)$
- **Não Degeneração**: não leva todos os pares de $G \times G$ à identidade em G_T
- **Computabilidade**: existe algoritmo eficiente que calcula $e(P, Q)$ para todo $P, Q \in G$

2) Problemas Computacionais Supostos Difíceis: Determinados problemas computacionais são considerados difíceis, isto é, não se conhece algoritmo eficiente (de complexidade de tempo polinomial) para resolvê-los. A segurança dos protocolos criptográficos aqui relacionados é fundamentada na suposição de dificuldade dos problemas definidos a seguir.

Seja $e : G \times G \rightarrow G_T$ um emparelhamento bilinear admissível, com $P \in G$ um gerador de G e valores aleatórios $a, b, c \in \mathbb{Z}_q$ e $Q \in G_T$. São considerados difíceis:

- **DL.** Problema do Logaritmo Discreto: dado $aP \in G$, calcular a.
- **CDH.** Problema Computacional Diffie-Hellman: dados $\langle P, aP, bP \rangle$, calcular abP.
- **BDH.** Problema Bilinear Diffie-Hellman: dados $\langle P, aP, bP, cP \rangle$, calcular $e(P, P)^{abc}$.
- **Gap BDH.** Problema Gap Bilinear Diffie-Hellman: dados $\langle P, aP, bP, cP \rangle$, calcular $e(P, P)^{abc}$, com a ajuda de um programa eficiente que responde se vale ou não a igualdade: $e(P, P)^{abc} = Q$

3) Mecanismo de Encapsulamento de Chave: Nossa proposta de protocolo embute um mecanismo de encapsulamento de chave (KEM), com o qual é possível cifrar um segredo (temporário), usando a chave pública do destinatário, e enviá-lo com garantia de sigilo e autenticidade [14]. Durante o processo de desencapsulamento, o segredo é decifrado com a chave secreta do receptor. É dito que um esquema KEM é CCA-seguro quando ele é resistente a ataques adaptativos de texto cifrado escolhido.

4) Propriedades de Segurança: As propriedades de segurança mais importantes e requeridas nos protocolos de acordo de chaves com autenticação são relacionadas a seguir. Os protocolos aqui mencionados contemplam todas elas.

- Resistência a ataques de personificação básicos: um adversário não deve ser capaz de personificar um usuário se não conhecer sua chave secreta.
- Resistência a ataques de compartilhamento desconhecido de chave (UKS, ou *Unknown Key-Share*): deve ser inviável convencer um participante A de que ele está compartilhando uma chave com B, quando na realidade está compartilhando com outro usuário C (honestamente registrado no sistema), enquanto C pensa (corretamente) estar compartilhando com A.
- Segurança de chave conhecida: cada execução do protocolo deve produzir uma chave de sessão única. O protocolo deve permanecer seguro mesmo que um adversário descubra algumas chaves de sessão já negociadas.
- Resistência a ataques de personificação pelo comprometimento de chave secreta (KCI, ou *Key-Compromise Impersonation*): se a chave secreta de longa duração de um usuário A é comprometida, um atacante não deve ser capaz de personificar um usuário B perante A.
- Segurança no futuro completa-fraca (wPFS, ou *Weak Perfect Forward Secrecy*): deve ser inviável para um atacante recuperar uma chave de sessão mesmo que, no futuro, venha a corromper as chaves secretas de longa duração de *todos* os participantes envolvidos naquela sessão, porém sob a condição de que ele não esteve ativamente envolvido na escolha dos segredos temporários para o cálculo daquela chave de sessão. Esta última condição é o que caracteriza a segurança PFS como *fraca*. Em [4] foi demonstrado que em protocolos com apenas duas passagens de mensagens (como é o caso de todos os aqui discutidos) wPFS é o melhor que se pode alcançar com relação à segurança no futuro.
- Resistência ao vazamento de segredos temporários: o vazamento de um valor secreto temporário não deve comprometer a segurança de sessões que não o tenham usado.

No caso especial em que não são necessários certificados digitais para as chaves públicas, duas propriedades adicionais são desejáveis:

- Segurança no futuro perante o KGC (*KGC Forward Secrecy*): o KGC deve ser incapaz de recuperar chaves de sessão, mesmo que monitore o tráfego durante o estabelecimento das chaves e que, portanto, tenha acesso a todos os dados públicos.
- Resistência ao vazamento de segredos temporários para o KGC: nos protocolos CL-AKA, o KGC ainda que seja capaz de aprender os valores secretos temporários de qualquer sessão (e fazendo uso de seu conhecimento de todas as chaves secretas parciais) deve ser incapaz de recuperar a chave de sessão.

III. Descrição do Protocolo Proposto

Nossa proposta segue a conversão genérica de um mecanismo KEM para um protocolo de acordo de chaves com autenticação, apresentado por Boyd, Cliff, González Nieto e Paterson [15]. Os autores demonstraram que, dado um esquema KEM baseado em identidade CCA-seguro, então a conversão produz um protocolo de acordo de chaves seguro sob o modelo de Canetti-Krawczyk [16], com a propriedade adicional de resistência a ataques KCI.

Como entrada da conversão de [15], usamos o mecanismo de encapsulamento de chave sem certificado (CL-KEM) gerado a partir da construção de Fiore, Gennaro e Smart [17], que é mais eficiente que os esquemas de [14]. O esquema CL-KEM empregado foi gerado a partir do protocolo SCK-2 de acordo de chaves baseado em identidade [8], modificado para que a conversão de [17] ocorra de forma consistente. As modificações que realizamos no protocolo SCK-2 mantêm-se seguras, conforme demonstramos na Seção V.

Portanto, o protocolo apresentado abaixo é uma concretização da conversão genérica de [15], com adaptações necessárias para o caso sem certificado. Ele é composto das seguintes fases:

- Inicialização do Sistema
- Geração de Chaves de Usuário
- Acordo de Chave
  1) Inicia
  2) Responde
  3) DerivaI
  4) DerivaR

A fase de *Acordo de Chave* é executada entre dois participantes, sempre que desejarem estabelecer novo segredo compartilhado. O iniciador da comunicação executa os passos *Inicia* e *DerivaI*, enquanto o receptor executa *Responde* e *DerivaR*. Descrevemos, a seguir, cada uma das etapas do protocolo.

Inicialização do Sistema

Sejam G e G_T grupos de ordem prima q com $P \in G$ um gerador de G. Seja $e : G \times G \rightarrow G_T$ um emparelhamento bilinear admissível.

O KGC escolhe $s \xleftarrow{\$} \mathbb{Z}_q^*$ como sua chave secreta mestra e calcula sua chave mestra pública como $mpk = sP$.

Para um parâmetro de segurança k, o KGC seleciona três funções de hash criptográficas:
$H : \{0,1\}^* \rightarrow \{0,1\}^k$
$H_1 : \{0,1\}^* \rightarrow G$
$H_2 : G \times G_T \times \{0,1\}^* \times G \times G \rightarrow \{0,1\}^k$

O KGC publica como parâmetros públicos do sistema $\langle q, G, G_T, e, k, P, sP, H, H_1, H_2 \rangle$.

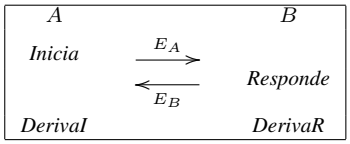
Geração de Chaves de Usuário

Cada usuário U:

- possui uma identidade única $ID_U \in \{0,1\}^*$ e um valor público $Q_U \in G$, onde $Q_U = H_1(ID_U)$
- escolhe $x_U \xleftarrow{\$} \mathbb{Z}_q^*$
- calcula sua chave pública $PK_U = x_U P$
- recebe do KGC a chave parcial secreta $d_U = sQ_U$, entregue de forma segura.

Acordo de Chave

Um participante A que deseja computar uma chave de sessão com B, inicia o protocolo executando os passos *Inicia* e *DerivaI*, abaixo. O participante B, por sua vez, executa os passos *Responde* e *DerivaR*.



1) *Inicia*:
   $t_A \xleftarrow{\$} \mathbb{Z}_q^*$
   $C_A \leftarrow t_A P$
   $y_A \xleftarrow{\$} \mathbb{Z}_q^*$
   $Y_A \leftarrow y_A P$
   $E_A \leftarrow (A, PK_A, C_A, Y_A)$
   envia E_A para B
   recebe $E_B = (B, PK_B, C_B, Y_B)$ de B
   verifica se C_B, Y_B estão em G e se PK_B está correto
   aborta em caso de erro
2) *Responde*:
   recebe $E_A = (A, PK_A, C_A, Y_A)$ de A
   verifica se C_A, Y_A estão em G e se PK_A está correto
   aborta em caso de erro
   $t_B \xleftarrow{\$} \mathbb{Z}_q^*$
   $C_B \leftarrow t_B P$
   $y_B \xleftarrow{\$} \mathbb{Z}_q^*$
   $Y_B \leftarrow y_B P$
   $E_B \leftarrow (B, PK_B, C_B, Y_B)$
   envia E_B para A
3) *DerivaI*:
   a) Encapsula temporário:
      $Z_{A_1} \leftarrow t_A PK_B$
      $Z_{A_2} \leftarrow e(t_A \cdot mpk, Q_B)$
      $K_A \leftarrow H_2(Z_{A_1}, Z_{A_2}, ID_B, PK_B, C_A)$
   b) Desencapsula temporário:
      $Z_{B_1} \leftarrow x_A C_B$
      $Z_{B_2} \leftarrow e(C_B, d_A)$
      $K_B \leftarrow H_2(Z_{B_1}, Z_{B_2}, ID_A, PK_A, C_B)$
   c) Calcula chave:
      $K_{AB} \leftarrow y_A Y_B$
      $M_1 \leftarrow (x_A PK_B, x_A Y_B, y_A PK_B)$
      $M_2 \leftarrow ((t_A + y_A)(C_B + Y_B), Z_{A_1}, Z_{B_1})$
      $sid \leftarrow (E_A, E_B)$
      $pid \leftarrow (ID_A, ID_B, app_{id})$
      $SK \leftarrow H(K_A, K_B, K_{AB}, M_1, M_2, sid, pid)$
4) *DerivaR*:
   a) Encapsula temporário:
      $Z_{B_1} \leftarrow t_B PK_A$
      $Z_{B_2} \leftarrow e(t_B \cdot mpk, Q_A)$
      $K_B \leftarrow H_2(Z_{B_1}, Z_{B_2}, ID_A, PK_A, C_B)$
   b) Desencapsula temporário:
      $Z_{A_1} \leftarrow x_B C_A$
      $Z_{A_2} \leftarrow e(C_A, d_B)$
      $K_A \leftarrow H_2(Z_{A_1}, Z_{A_2}, ID_B, PK_B, C_A)$
   c) Calcula chave:
      $K_{AB} \leftarrow y_B Y_A$
      $M_1 \leftarrow (x_B PK_A, y_B PK_A, x_B Y_A)$
      $M_2 \leftarrow ((t_B + y_B)(C_A + Y_A), Z_{A_1}, Z_{B_1})$
      $sid \leftarrow (E_A, E_B)$
      $pid \leftarrow (ID_A, ID_B, app_{id})$
      $SK \leftarrow H(K_A, K_B, K_{AB}, M_1, M_2, sid, pid)$

Uma comparação entre este protocolo e outros dois correlatos será feita posteriormente na Seção VI.

IV. Demonstração de Consistência

Vamos mostrar que a chave calculada por A é a mesma calculada por B, ou seja, que *DerivaI* e *DerivaR* produzem o mesmo valor SK. Aplicando-se a propriedade de comutatividade em $\mathbb{Z}_q$ e a propriedade de bilinearidade do emparelhamento e, tem-se as seguintes igualdades:

$$\begin{aligned} Z_{A_1} &= t_A PK_B = t_A x_B P = x_B t_A P = x_B C_A \\ Z_{A_2} &= e(t_A.mpk, Q_B) = e(t_A sP, Q_B) \\ &= e(t_A P, sQ_B) = e(C_A, d_B) \\ Z_{B_1} &= x_A C_B = x_A t_B P = t_B x_A P = t_B PK_A \\ Z_{B_2} &= e(C_B, d_A) = e(t_B P, sQ_A) \\ &= e(t_B sP, Q_A) = e(t_B.mpk, Q_A) \end{aligned}$$

De forma análoga, é possível verificar que os valores K_{AB}, M_1 e M_2 são igualmente calculados por A e B, de modo que ambos chegam à mesma chave compartilhada SK.

V. Demonstração de Segurança

Descrevemos nesta seção as ideias gerais da demonstração de segurança de nossa proposta de protocolo, que é seguro sob a hipótese de dificuldade do problema Gap BDH, no modelo de segurança de Canetti-Krawczyk (CK) [16] adaptado ao caso sem certificado, e sob o modelo de oráculos aleatórios [18].

Como São Garantidas as Propriedades de Segurança

O modelo CK de segurança para protocolos de acordo de chaves modela as propriedades de segurança descritas na Seção II-4, com exceção de KCI, KGC *Forward Secrecy* e Resistência ao vazamento de segredos temporários para o KGC. Descreveremos, portanto, nos parágrafos a seguir, como nosso protocolo alcança essas últimas propriedades.

A propriedade KGC *Forward Secrecy* é alcançada por conta do segundo segredo temporário sorteado durante a sessão (y_A). Se o KGC não recuperar os valores secretos temporários y_A ou y_B, não será capaz de recuperar a chave de sessão acordada entre A e B, conhecendo apenas os dados públicos E_A e E_B (mais especificamente, Y_A e Y_B), pois em caso contrário estaria resolvendo o problema do Logaritmo Discreto. Formalmente, esse aspecto é demonstrado no Teorema 2 de [15].

Também em [15] é demonstrado que a conversão alcança a propriedade KCI, no caso de protocolos baseados em identidade. Para o caso sem certificado, introduzimos os valores M_1 e M_2, para que, mesmo que um adversário substitua chaves públicas por falsos valores à sua escolha, ou simplesmente corrompa os valores secretos x_A e x_B, não realize ataques de personificação do tipo KCI. Os valores M_1 e M_2 refletem as combinações possíveis dos valores secretos dos participantes, x_A e x_B, com os efêmeros t_A, t_B, y_A, y_B. Assim, se o adversário substituir uma chave pública ou corromper o x associado e obtiver vantagem sobre o criptossistema, há dois casos a considerar: (1) o adversário corrompeu os valores efêmeros e (2) corrompeu a chave secreta parcial (ambos os casos, simultaneamente com o comprometimento de x não é possível, pois nesse caso o adversário não alcançaria vantagem na sessão de teste e perderia o jogo). No caso (1), a vantagem do adversário é usada para se construir um algoritmo eficiente que resolve o problema Diffie-Hellman bilinear; em (2) de forma análoga se resolve o Diffie-Hellman computacional.

A resistência ao vazamento de segredos temporários para o KGC é obtida por causa das definições de M_1 e Z_{B_1}, que requerem o conhecimento do valor secreto x_A que gerou a chave pública, pois em caso contrário seria possível resolver o problema computacional Diffie-Hellman.

Modelo de Segurança

Os protocolos LBG e GOT foram demonstrados seguros segundo o modelo de [11], que é uma extensão do modelo CK, conhecida por eCK e definida em [19]. Swanson [9] e Lippold et al. estenderam eCK para incluir adaptações para o caso sem certificado. Os modelos CK e eCK diferem fundamentalmente no modo como são tratados os vazamentos dos segredos temporários. Enquanto em CK *todos* os segredos temporários de uma sessão podem ser revelados de uma única vez (por meio do oráculo *SessionStateReveal*), em eCK os temporários podem ser comprometidos um a um, à escolha do adversário.

Na demonstração de segurança de nossa proposta, reescrevemos o modelo CK com as mesmas adaptações realizadas em [9] e [11] e definimos o oráculo *SessionStateReveal* para revelar simultaneamente os valores efêmeros t_U e y_U do participante U, para uma dada sessão.

Construção de Fiore, Gennaro e Smart

Fiore, Gennaro e Smart [17] desenvolveram uma construção genérica de mecanismos de encapsulamento de chave sem certificado (CL-KEM) a partir de protocolos de acordo de chave com autenticação baseados em identidade.

Em uma versão preliminar de [17], colocada no *Cryptology ePrint Archive*, os autores apresentaram dois exemplos de protocolos aplicando a conversão proposta por eles e afirmaram que ambos exemplos eram CCA-seguros. O primeiro exemplo foi criado a partir de SCK-2 [8] e o segundo, a partir de FG [20]. No entanto, Yan e Tan em [21] mostraram um ataque para o segundo exemplo de CL-KEM, construído a partir do protocolo FG; também apresentaram uma versão corrigida.

Nosso protocolo usa como ponto de partida o protocolo SCK-2. Para que nossa proposta seja inteiramente consistente e segura, propomos uma variante que chamaremos de SCK2-OWA. Na subseção a seguir detalhamos a variante e mostramos que ela é segura, nas condições requeridas por [17].

Por fim, aplicando-se a conversão de Fiore, Gennaro e Smart ao SCK2-OWA, de acordo com os Teoremas 2, 3 e 4 de [17] obtemos um CL-KEM CCA-seguro que satisfaz as condições necessárias à aplicação da conversão seguinte, de [15].

Protocolo SCK2-OWA e Sua Segurança

O protocolo de acordo de chaves baseado em identidade com autenticação mútua SCK-2 foi originalmente proposto por Smart e melhorado por Chen e Kudla [8]. A variante SCK2-OWA garante autenticação apenas do participante que inicia a comunicação (*one-way authenticated*). Protocolos de acordo de chave com autenticação unilateral têm como exemplo clássico o SSL/TLS que, nas implementações mais corriqueiras, diferentemente dos clientes, o fornecedor de serviços é previamente autenticado por meio de certificado digital. Na conversão de [17], a autenticação unilateral é requerida para aumentar a eficiência do CL-KEM resultante.

O SCK2-OWA que propomos é composto pelas fases:

Inicialização do Sistema: a autoridade do sistema escolhe um emparelhamento bilinear admissível $e : G \times G \to G_T$ e um gerador P de G. Escolhe sua chave secreta mestra s; calcula sua chave mestra pública como $mpk = sP$ e seleciona as funções de hash $H_1 : \{0,1\}^* \to G$ e $H : G \times G_T \times \{0,1\}^* \times G \times G \to \{0,1\}^k$

Geração de Chaves de Usuário: cada usuário com identidade ID possui o valor público $Q_{ID} = H_1(ID)$ e sua chave secreta é $d_{ID} = sQ_{ID}$.

Acordo de Chave: os participantes A e B executam os passos do quadro a seguir, sempre que desejarem compartilhar uma chave de sessão:

A		B
$t_A \xleftarrow{\$} \mathbb{Z}_q^*$		$t_B \xleftarrow{\$} \mathbb{Z}_q^*$
$T_A \leftarrow t_A P$	$\xrightarrow{T_A}$	$T_B \leftarrow t_B P$
	$\xleftarrow{T_B}$	
$x \leftarrow t_A T_B$		$x \leftarrow t_B T_A$
$y \leftarrow e(T_B, d_A)$		$y \leftarrow e(t_B \cdot mpk, Q_A)$
$SK \leftarrow H(x, y, ID_A, T_A, T_B)$		$SK \leftarrow H(x, y, ID_A, T_A, T_B)$

Para que a conversão de Fiore, Gennaro e Smart funcione, o protocolo de entrada deve ser seguro sob um modelo mais forte que o convencional, no qual existe um oráculo chamado *Reveal**, que permite que o adversário descubra o valor de uma sessão com base apenas nas mensagens trocadas entre os participantes. Ainda assim, é possível seguir os passos principais da demonstração dos Teoremas 1 e 2 de [8] e fazer uso da vantagem do adversário para capturar a solução para uma instância de BDH em $e(caP, bP)$, inserida na variável y.

VI. Comparação com Protocolos Relacionados

Nossa proposta é comparável aos protocolos LBG [11] e GOT [12], pois todos possuem as mesmas propriedades de segurança listadas na Seção II-4 e tratam adversários com poderes equivalentes.

Ambos os protocolos LBG e GOT possuem duas versões, para dois níveis de segurança teórica. Isso é devido ao fato de que os dois são demonstrados seguros sob a hipótese de dificuldade do problema BDH e admitem uma variante aproximadamente 50% mais veloz que pode ser demonstrada segura sob a hipótese de dificuldade do problema Gap BDH. Este último problema computacional é inferior ao BDH, pois sempre que o Gap BDH valer em um determinado grupo algébrico, também valerá o BDH para o mesmo grupo (mas não é garantido que o Gap BDH é difícil onde o BDH o for).

Nosso protocolo proposto baseia-se na dificuldade do Gap BDH e não possui demonstração de segurança para o problema mais forte BDH. Portanto as comparações de desempenho (em tempo de processamento) são realizadas com as versões simplificadas de LBG e GOT, de forma que todos são nivelados ao mesmo patamar de segurança teórica.

Os protocolos LBG e GOT são demonstrados seguros sob um modelo de segurança variante de eCK, descrito na Seção V, enquanto nossa proposta é segura sob uma variante de CK. Embora os modelos CK e eCK não sejam comparáveis formalmente (por serem modelos distintos), em termos práticos a comparação dos protocolos é justa, pois os adversários contra cada um desses protocolos agem de maneira similar.

Para os três protocolos é possível pré-computar alguns valores, quando o usuário se comunicar sempre com os mesmos parceiros. Nesse caso, em nosso protocolo pode-se calcular previamente, por exemplo, $e(mpk, Q_B)$ e durante a execução do protocolo é realizada uma exponenciação por t_A; o usuário A pode calcular e armazenar em memória segura o valor $x_A PK_B$.

Na Tabela I, apresentamos a quantidade de operações mais relevantes que cada participante precisa calcular em uma sessão com os protocolos LBG, GOT e nossa proposta (Prop). Dispusemos nas primeiras linhas as operações que, em média, são mais lentas.

Para o cenário sem pré-computação de valores, o protocolo que aqui propomos economiza nas operações mais caras (cálculo de emparelhamentos bilineares e multiplicações no grupo resultante do emparelhamento G_T), enquanto dispende mais operações que custam menos. As tomadas de desempenho apresentadas na Seção VII mostram que nosso protocolo é mais veloz que seus antecessores. Sob o contexto de pré-cálculo e armazenamento, nosso protocolo é apenas o segundo mais eficiente.

VII. Aplicação para Dispositivos Móveis

Nesta seção, apresentamos um estudo de caso com aplicação do protocolo proposto em um sistema envolvendo dispositivos móveis. Descrevemos a implantação, escolha de parâmetros e resultados experimentais.

Estudo de Caso

Os Centros de Saúde desempenham o papel de órgão provedor da atenção primária à saúde no âmbito do Sistema Único de Saúde (SUS) brasileiro. Apesar da tecnologia contribuir para novas descobertas e em novas formas de oferecer uma assistência à saúde de melhor qualidade à população, ainda não existe um sistema computacional aberto para o gerenciamento de um Centro de Saúde que seja integrado, flexível e ágil.

O SUS brasileiro atribui aos Centros de Saúde, também denominados de unidades básicas de saúde, o papel de órgão provedor da atenção primária à saúde. Para que este papel seja cumprido com responsabilidade e eficácia, se mostrou fundamental a condução de programas públicos de atenção domiciliar, tais como o Programa Saúde da Família e o Programa de Saúde Bucal.

O objetivo destes programas é o de melhorar a qualidade do serviço de saúde prestado à população por meio da aproximação entre equipes de saúde e a comunidade. A estratégia permite, sobretudo, uma mudança do paradigma de tratamento de doenças para o de promoção da saúde. Apesar de sua importância para a organização e articulação do sistema de atenção primária, os programas de atenção domiciliar são normalmente conduzidos com pouco – ou nenhum – suporte de Tecnologia da Informação (TI).

Com o objetivo de prover melhores ferramentas de TI para os profissionais de saúde, em 2004 foi iniciado o desenvolvimento de um sistema de captura de dados para dispositivos móveis, por exemplo, *smartphones* e PDAs, capaz de auxiliar o trabalho dos profissionais de saúde nos atendimentos domiciliares.

O sistema Borboleta está no processo de evolução de uma ferramenta móvel de apoio à coleta de informações para um Sistema Móvel de Prontuário Eletrônico, que integra novas tecnologias e está fortemente baseado no conceito de acompanhamento da situação clínica dos pacientes.

Tabela I
Comparação dos Protocolos

	Sem Pré-computação			Com Pré-computação		
	LBG	GOT	Prop	LBG	GOT	Prop
Emparelhamentos	5	4	2	1	1	1
Exponenciações em G_T	0	0	0	1	0	1
Multiplicações em G_T	2	1	0	1	0	0
Multiplicações em G	5	5	10	4	5	8
Adições em G	0	2	1	0	2	1

O sistema está atualmente sendo validado pelos profissionais de saúde do Centro de Saúde Escola Butantã, parceiros no desenvolvimento do projeto e responsáveis por conduzir um Programa de Atenção Primária Domiciliar [22].

O conceito de sistema de Prontuário Eletrônico (PE) é mais amplo do que o de coleta de informação, pois implica armazenar os registros médicos (prontuários) dos pacientes em meios digitais [23]. Estes sistemas agregam valor ao serviço de saúde e, em larga escala, podem reduzir custos [24]. No Brasil, a Sociedade Brasileira de Informática em Saúde (SBIS) elaborou um conjunto de normas técnicas que devem ser seguidas para a construção de um PE.

No Projeto Borboleta, a segurança da informação que trafega entre dispositivos móveis conectados sem fio se torna inerentemente necessária. Os PDAs com baixo poder de processamento e conexões de baixa largura de banda exigem algoritmos especiais em velocidade e nível de segurança alta.

Implantação

No estudo de caso do projeto Borboleta, nós utilizamos o acordo de chaves para combinar uma chave secreta entre um PDA, em posse de um agente de saúde em atendimento domiciliar, e um servidor. Esta chave combinada é usada para trocar as informações com garantia de confidencialidade.

Antes da utilização do protocolo de acordo de chaves, é necessária uma etapa de preparação do sistema, no caso o PDA e o servidor de dados dos pacientes.

Inicialmente, o KGC é instalado em uma máquina de acesso restrito. Ele sorteia a chave mestra s armazenando-a em memória segura e calcula a chave pública do sistema mpk.

Em seguida, o servidor e o celular, também em um ambiente controlado, executam os passos descritos no subitem *Geração de Chaves de Usuário* na Seção III.

Um PDA A calcula Q_A a partir da identificação do aparelho. Ele sorteia um valor secreto x_A, e o armazena em memória segura do PDA, para gerar a sua chave pública PK_A. O servidor B executa a geração de chaves analogamente, gerando x_B e armazenando-o em memória segura do servidor.

O KGC calcula e entrega as respectivas chaves parciais secretas d_A do PDA e d_B do servidor, concluindo a etapa de preparação do sistema. Todos os PDAs, utilizados pelos agentes de saúde, serão inicializados desta forma.

Após esta etapa de inicialização do sistema, o PDA A se encarrega de iniciar as sessões, executando os passos *Inicia* e *DerivaI*. Por sua vez, o servidor B executa os passos *Responde* e *DerivaR*.

Estabelecida uma chave de sessão com o servidor, o PDA torna-se apto a enviar e receber informações cifradas, com o auxílio de uma cifra de bloco padronizada, como o AES.

Escolha de Parâmetros e Ambiente de Testes

Escolhemos as curvas supersingulares B-271 e B-1223, sobre os corpos binários com grau de mergulho $k = 4$, padronizadas respectivamente em [25] e na RFC 5349 [26]. A primeira curva oferece nível de segurança de 70 bits, enquanto a segunda, de 128 bits. Adotamos o emparelhamento ηT [27] na implementação dos testes comparativos.

O tamanho das chaves é escolhido de modo a oferecer segurança criptográfica suficiente. Os autores de [25] fornecem uma comparação dos tamanhos das chaves aproximados para criptossistemas simétricos e assimétricos, com base nos algoritmos mais eficientes para atacá-los.

O Projeto Borboleta requer a implementação em Java devido à fácil portabilidade entre os dispositivos de baixo poder de processamento. Os resultados apresentados a seguir foram obtidos, entretanto, com a biblioteca criptográfica Relic 0.2.3 [28] escrita em C ANSI, para uma avaliação preliminar.

No experimento, foi empregado um computador com processador Intel de 1,6 GHz para simulação de execução dos protocolos entre um celular e o servidor.

Resultados Experimentais

Na Tabela II, apresentamos os tempos em segundos que cada participante precisa, em média, para executar uma sessão dos protocolos. Também mostramos o ganho relativo de nossa proposta em relação aos outros protocolos; valores negativos representam o quanto nossa proposta foi mais lenta que o primeiro colocado na respectiva classe de comparação.

A metade superior da Tabela II se refere à simulação das versões que são demonstradas seguras sob a hipótese de dificuldade do problema computacional Gap BDH, sem pré-computação de valores. Essa configuração confere maior nível de segurança.

A metade inferior reflete o cenário com valores computados e armazenados previamente em memória segura. Nesse caso, os protocolos são executados em menor tempo, porém ao custo de maior uso de memória e possível redução do nível de segurança, que passa a depender do grau de proteção dado a esse armazenamento auxiliar.

Observarmos que nosso protocolo foi significativamente mais rápido que os demais, na maioria das comparações. Em particular, foi mais eficiente sob a configuração com o maior nível de segurança, com curvas para o patamar de segurança de 128 bits e sem gasto de memória adicional. Nesse contexto, nossa proposta foi cerca de 53% mais veloz que GOT e 65% mais rápido que LBG.

Para aplicações em que não há restrições severas para o uso de memória nem impedimento para armazenamento seguro dos valores pré-computados a partir das chaves secretas, o

Tabela II
DESEMPENHO DOS PROTOCOLOS, EM SEGUNDOS

		B-271: 70 bits			B-1223: 128 bits	
sem pré-computação	*Proposto*	**0,13**			**7,20**	
	GOT	0,22	(41%)	$(*)$	15,41	(53%)
	LBG	0,29	(55%)		20,74	(65%)
com pré-computação	*Proposto*	**0,06**			**4,15**	
	GOT	0,05	(-20%)	$(**)$	3,75	(-11%)
	LBG	0,09	(33%)		5,96	(30%)

$(*)$ *Proposto* é mais rápido ou $(**)$ mais lento que o protocolo referido.

protocolo GOT é mais indicado, por oferecer melhor desempenho.

No caso da aplicação alvo de estudo, os prontuários eletrônicos podem consumir considerável quantidade de memória quando, por exemplo, há imagens médicas para diagnóstico. Em situações como essa, certos modelos de PDA podem não comportar a implementação que exige armazenamento de valores pré-calculados. E recomenda-se, nesse caso, o uso do protocolo aqui proposto por oferecer o segundo melhor tempo.

VIII. Conclusões e Trabalhos Futuros

Propusemos um novo protocolo de acordo de chaves com autenticação mútua no modelo de criptografia de chave pública sem certificado, que pode ser demonstrado seguro sob um forte modelo de segurança. Apresentamos resultados experimentais de implementação de uma aplicação sobre dispositivos móveis, atestando maior eficiência de nossa proposta perante protocolos anteriores. Mostramos que o protocolo é indicado para aplicações que requeiram maior nível de segurança ou em que haja restrição de uso de memória dos dispositivos móveis.

Em outro trabalho, apresentaremos a demonstração formal de segurança do protocolo proposto, sob o modelo de oráculos aleatórios e hipótese de dificuldade computacional do problema Gap Bilinear Diffie-Hellman, sob o modelo de Canneti-Krawczyk, adaptado para o contexto de criptografia de chave pública sem certificados.

Agradecimentos

Este trabalho foi financiado por Fapesp, sob os projetos de números 2008/06189-0 2008/50412-5.

Referências

[1] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *ASIACRYPT 2003*, ser. Lecture Notes in Computer Science, vol. 2894. Springer, 2003, cryptology ePrint Archive, Report 2003/126, http://eprint.iacr.org/.

[2] A. Menezes, M. Qu, and S. A. Vanstone, "Some key agreement protocols providing implicit authentication," in *2nd Workshop Selected Areas in Cryptography, SAC'95*, 1995, pp. 22–32.

[3] *P1363 Standard Specifications for Public-Key Cryptography*, IEEE, 2000.

[4] H. Krawczyk, "Hmqv: a high-performance secure diffie-hellman protocol," in *Advances in Cryptology â CRYPTO 2005, LNCS 3621*, 2005, p. 546566.

[5] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*. Berlin, Germany: Springer, 2003.

[6] R. Terada, *Segurança de Dados - Criptografia em Redes de Computador*, 2nd ed. São Paulo, SP: Editora Edgard Blücher, 2008.

[7] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of CRYPTO 84 on Advances in cryptology*, ser. Lecture Notes in Computer Science, vol. 196/1985. New York, NY, USA: Springer-Verlag New York, Inc., 1984, pp. 47–53.

[8] L. Chen, Z. Cheng, and N. P. Smart, "Identity-based key agreement protocols from pairings," *Int. J. Inf. Secur.*, vol. 6, no. 4, pp. 213–241, 2007.

[9] C. M. Swanson, "Security in key agreement: Two-party certificateless schemes," Master's thesis, University of Waterloo, 2008, http://hdl.handle.net/10012/4156.

[10] C. Swanson and D. Jao, "A study of two-party certificateless authenticated key-agreement protocols," in *INDOCRYPT '09: Proceedings of the 10th International Conference on Cryptology in India*, ser. Lecture Notes in Computer Science, vol. 5922. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 57–71.

[11] G. Lippold, C. Boyd, and J. González Nieto, "Strongly secure certificateless key agreement," in *Pairing '09: Proceedings of the 3rd International Conference Palo Alto on Pairing-Based Cryptography*, ser. Lecture Notes in Computer Science, vol. 5671. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 206–230, cryptology ePrint Archive, Report 2009/219, http://eprint.iacr.org/.

[12] D. Goya, C. Okida, and R. Terada, "A two-party certificateless authenticated key agreement protocol," in *SBSeg 2010 X Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*. Sociedade Brasileira de Computação, 2010.

[13] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.

[14] K. Bentahar, P. Farshim, J. Malone-Lee, and N. P. Smart, "Generic constructions of identity-based and certificateless kems," *J. Cryptol.*, vol. 21, no. 2, pp. 178–199, 2008.

[15] C. Boyd, Y. Cliff, J. M. G. Nieto, and K. G. Paterson, "One-round key exchange in the standard model," *Int. J. Appl. Cryptol.*, vol. 1, no. 3, pp. 181–199, 2009.

[16] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *EUROCRYPT '01: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*. London, UK: Springer-Verlag, 2001, pp. 453–474.

[17] D. Fiore, R. Gennaro, and N. Smart, "Constructing certificateless encryption and id-based encryption from id-based key agreement," in *Pairing 2010*. Springer, 2010, no prelo. Disponível em Cryptology ePrint Archive, Report 2009/600.

[18] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *First ACM Conference on Computer and Communications Security*. Fairfax, Virginia, USA: ACM, 1993, pp. 62–73.

[19] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *ProvSec'07: Proceedings of the 1st international conference on Provable security*, ser. Lecture Notes in Computer Science, vol. 4784. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 1–16.

[20] D. Fiore and R. Gennaro, "Making the diffie-hellman protocol identity-based," in *CT-RSA*, ser. Lecture Notes in Computer Science, vol. 5985. Springer, 2010, pp. 165–178.

[21] C. T. G. Yang, "Certificateless public key encryption: A new generic construction and two pairing-free schemes," *Theoretical Computer Science*, 2010, no prelo.

[22] G. L. Duarte, R. Correia, P. Leal, H. Domingues, F. Kon, R. Kon, and J. E. Ferreira, "Borboleta and saguisaúde - open source mobile telehealth for public home healthcare," in *Med-e-Tel 2010, VIII International eHealth*, ser. Telemedicine and Health ICT Forum, 2010.

[23] D. w. Bates, M. Ebell, E. Gotlieb, J. Zapp, and H. Mullins, "A proposal for electronic medical records in u.s. primary care," *Journal of the American Medical Informatics Association*, vol. 10, no. 1, pp. 1–10, Jan 2003.

[24] S. J. Wang, B. Middleton, L. A. Prosser, C. G. Bardon, C. D. Spurr, P. J. Carchidi, A. F. Kittler, R. C. G. andDavid G. Fairchild, A. J. Sussman, G. J. Kuperman, and D. W. Bates, "A cost-benefit analysis of eletronic medical records in primary care," *The American Journal of Medicine*, vol. 114, no. 5, pp. 397–403, April 2003.

[25] J. N. Nist, B. Kaliski, and R. Security, "Comments received on sp 800-57 recommendation for key management, part 1: General guideline."

[26] L. Zhu, K. Jaganathan, and K. Lauter, "Elliptic Curve Cryptography (ECC) Support for Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)," RFC 5349 (Informational), Internet Engineering Task Force, Sep. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5349.txt

[27] P. Barreto, S. Galbraith, C. hÉigeartaigh, and M. Scott, "Efficient pairing computation on supersingular abelian varieties," *Designs, Codes and Cryptography*, vol. 42, pp. 239–271, 2007.

[28] Relic, "0.2.3," http://code.google.com/p/relic-toolkit/.