Correção de Deficiências no Acordo de Chaves de Mandt

Vilc Queupe Rufino[1], **Routo Terada**[2]

[1] Centro de Estudos da Marinha em São Paulo – Marinha do Brasil
Av. Professor Lineu Prestes, 2468 - Cidade Universitária
São Paulo – SP – Brasil

[2]Instituto de Matemática e Estatística – Universidade de São Paulo (USP)
Rua do Matão, 1010 - Cidade Universitária
São Paulo – SP – Brasil

{vilc,rt}@ime.usp.br

Abstract. *Mandt and Tan proposed an efficient certificateless key agreement schema based on intractability of the computational Diffie-Hellman problem. They still proposed an extended version to use with different trusted authorities. This paper shows that the extended version has a deficiency and presents a possible solution. It shows another possible correction to known vulnerability. Although our solution has complexity higher than original protocol, it is useful to develop hierarchical key agreement certificateless schemas.*

Resumo. *Mandt e Tan propuseram um esquema eficiente para acordo de chaves sem certificados, baseando-se na intratabilidade do Problema Computacional de Diffie-Hellman. Apresentaram no mesmo trabalho uma versão estendida para utilização com diferentes autoridades de confiança. Neste artigo identificamos uma deficiência na versão estendida e apresentamos uma possível solução. Também sugerimos uma correção para outra vulnerabilidade já conhecida. Embora nossas soluções levam a um protocolo final com maior complexidade computacional, o resultado é útil para o desenvolvimento esquemas de acordos de chaves hierárquicos sem certificados (certificateless).*

1. Introdução

"A distribuição de chaves criptográficas é o principal problema em sistemas criptográficos" [Blundo et al. 1993]; e "Protocolos de Acordo de Chaves são fundamentais para assegurar comunicação autêntica e privada entre duas entidades sobre uma rede insegura" . [Strangio 2006]

Há uma grande variedade no uso de protocolos para acordos de chaves, em especial, acordos de chaves baseados em sistemas sem certificados possuem a vantagem de dispensar o uso de uma infraestrutura de chaves públicas para certificação de chaves; e ainda não possuem a custódia da chave privada de suas entidades participantes, tal como nos sistemas baseados na identidade.

Sistemas sem certificados usam a identidade como parte da chave pública, e permitem que as entidades participantes distribuam suas chaves públicas antes de obterem valores secretos da Autoridade de Confiança (KGC - *Key Generation Center*).

Usamos neste texto o termo vulnerabilidade para descrever uma falha não prevista durante o projeto, que se explorada pode comprometer diretamente os atributos de segurança do esquema ou protocolo. O termo deficiência usamos para descrever um aspecto não desejável, que isoladamente não afeta os atributos de segurança, mas junto com outras características poderá se tornar uma vulnerabilidade.

1.1. Motivação

[Mandt e Tan 2006] propuseram um esquema para acordo de chaves sem certificados e com segurança baseada no problema bilinear computacional de Diffie e Hellman. Em sua proposta inicial descreveram um protocolo que minimizava em pelo menos dois emparelhamentos bilineares em relação ao modelo inicial proposto em [Al-Riyami e Paterson 2003]. Contudo após um estudo mais detalhado observou-se que havia falhas, e provavelmente contribuíram para a não evolução do esquema e o não uso nas aplicações sugeridas pelo autor.

Deficiências não identificadas podem ser mais nocivas que as conhecidas, pois um usuário que conheça os riscos poderá evitá-los com normas e procedimentos, porém caso não desconfie poderá permitir o comprometimento de informações sensíveis, destinadas exclusivamente a usuários legítimos. Tal como um consumidor que faz compras na internet usando seu número de cartão de crédito e código de segurança, se o consumidor desconfiar que as informações podem estar sendo enviadas para um usuário ilegítimo ele pode optar por outra forma de pagamento, ou ainda, a operadora de cartão pode orientar ao comprador a realizar compras somente em sites previamente cadastrados. Em [Singh 2000] mostra no primeiro capítulo um fato histórico onde a vulnerabilidade de uma cifra culminou na condenação e morte da rainha da Escócia.

1.2. Contribuição

Este trabalho identifica uma nova deficiência no protocolo proposto por Mandt em sua versão para duas autoridades de confiança (KGCs), e propõe possíveis soluções para esta deficiência e para uma vulnerabilidade descrita em [Swanson 2008]. Ainda faz uma breve descrição da utilização do protocolo corrigido para uso em acordos de chaves hierárquicos.

2. Conceitos Básicos

2.1. Grupos

Um grupo $\mathbb{G}$ é um conjunto não vazio dotado de uma operação binária $\circ$, que satisfaz as seguintes propriedades [Koblitz 1994]:

- Possui um elemento identidade: $\exists i \in \mathbb{G} : \forall a \in \mathbb{G} : i \circ a = a \circ i = a$;
- Possui o elemento inverso: $\forall a \in \mathbb{G} : \exists \bar{a} \in \mathbb{G} : a \circ \bar{a} = i$
- Associatividade: $\forall Q, R, S \in \mathbb{G} : (Q \circ R) \circ S = Q \circ (R \circ S)$;
- Fechamento: $\forall Q, R \in \mathbb{G} : Q \circ R \in \mathbb{G}$.

Quando um grupo é definido para operações de adição podemos dizer que é um grupo aditivo, e iremos representá-lo por $\mathbb{G}_1$; quando um grupo é definido para operações de multiplicação dizemos que é um grupo multiplicativo e iremos representá-lo por $\mathbb{G}_2$.

Quando o número de elementos de um grupo é finito, este número e chamado de ordem do grupo.

Podemos aplicar a operação $\circ$ sobre o mesmo elemento $Q \in \mathbb{G}$ do grupo várias vezes, tal como $\underbrace{Q \circ Q \circ Q \circ \cdots \circ Q}_{a\,vezes}$. Para $Q \in \mathbb{G}_1$ representamos por aQ, para $Q \in \mathbb{G}_2$ representamos por Q^a, onde $a \in \mathbb{N}$.

$Q \in \mathbb{G}_1$ é um elemento gerador do grupo se $\exists a \in \mathbb{N} : \forall R \in \mathbb{G}_1 : R = aQ$ ou se $Q \in \mathbb{G}_2$, $\exists a \in \mathbb{N} : \forall R \in \mathbb{G} : R = Q^a$.

Se o grupo possui um elemento gerador é chamado de cíclico.

2.2. Emparelhamento Bilinear

O protocolo proposto por Mandt usa emparelhamentos bilineares admissíveis, que foi proposto inicialmente para fins criptográficos em [Sakai et al. 2000], mas foi em [Boneh e Franklin 2001] onde propuseram o primeiro esquema cifragem baseado na identidade considerado eficiente e demonstrado seguro. A definição a seguir segue esses artigos:

Definição 2.1. Sejam $\mathbb{G}_1$ e $\mathbb{G}_2$ grupos cíclicos de ordem prima q. Um emparelhamento bilinear admissível é um mapeamento $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, que satisfaz as seguintes propriedades:

1. **Bilinear**: Para qualquer $P, Q, R \in \mathbb{G}_1$, temos:
   $e(P + Q, R) = e(P, R) \cdot e(Q, R)$
   $e(P, Q + R) = e(P, Q) \cdot e(P, R)$
   em particular para $a, b \in \mathbb{Z}_q$, temos:
   $e(aQ, bQ) = e(Q, Q)^{ab} = e(abQ, Q) = e(Q, abQ)$
2. **Não Degeneração**: Não leva todos os pares $\mathbb{G}_1 \times \mathbb{G}_1$ à identidade em $\mathbb{G}_2$
3. **Computável**: Existe um algoritmo eficiente que calcula $e(P, Q)$ para todos $P, Q \in \mathbb{G}_1$

2.3. Problemas Computacionais de Interesse

Em [Castro et al. 2007] é apresentado uma introdução de noções fortes de segurança, e descrita a metodologia para demonstrar a segurança de um algoritmo criptográfico assimétrico, em suma, é feita uma redução de algum problema difícil à quebra do protocolo alvo. O protocolo de Mandt está baseado nas dificuldades dos problemas do logaritmo discreto sobre curvas elípticas (PLD-CE), problema de Diffie-Hellman bilinear (BDH) e o problema de decisão de Diffie-Hellman bilinear (DBDH).

O Problema do logaritmo discreto sobre curvas elípticas provavelmente é mais difícil que o problema equivalente em $\mathbb{Z}_q$; apresentamos abaixo a definição deste problema tal como em [Terada 2008]:

Definição 2.2. "Problema do Logaritmo Discreto sobre Curvas Elípticas" *(PLD-CE)*:

- Seja $\mathbb{G}$ um grupo finito com a operação $\circ$, $Q \in \mathbb{G}$ gerador do subgrupo $\mathbb{J} \subseteq \mathbb{G}$
- Dado $R \in \mathbb{J}$, onde $R \neq Q$
- Encontrar $a \in \mathbb{Z}_q^*$, tal que $R = \underbrace{Q \circ Q \circ \ldots Q}_{a\,vezes}$, onde $1 \leq a \leq (|\mathbb{J}| - 1)$

Se $\mathbb{G}$ é o conjunto finito de pontos sobre uma curva elíptica e a operação $\circ$ é a soma de dois pontos, então encontrar o valor a, tal que $R = aQ$.

Seguem as definições do problemas de Diffie-Hellman Bilinear como apresentado em [Boneh e Franklin 2001]:

Definição 2.3. "Problema de Diffie-Hellman Bilinear" *(BDH)*:

- Sejam $\mathbb{G}_1$ e $\mathbb{G}_2$ grupos cíclicos de ordem prima q, o valor Q um gerador do grupo $\mathbb{G}_1$, valores $a, b, c \in \mathbb{Z}_q^*$ e a função $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ um emparelhamento bilinear admissível;
- Dados $Q, aQ, bQ, cQ \in \mathbb{G}_1$;
- Encontrar $e(Q, Q)^{abc} \in \mathbb{G}_2$.

Definição 2.4. "Problema de Decisão de Diffie-Hellman Bilinear" *(DBDH)*:

- Sejam $\mathbb{G}_1$ e $\mathbb{G}_2$ grupos cíclicos de ordem prima q, o valor Q um gerador do grupo $\mathbb{G}_1$, valores $a, b, c \in \mathbb{Z}_q^*$ e a função $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ um emparelhamento bilinear admissível;
- Dados $Q, aQ, bQ, cQ \in \mathbb{G}_1$ e $z \in \mathbb{G}_2$;
- Decidir se $z = e(Q, Q)^{abc}$.

Usaremos os problemas acima para evidenciarmos as correções de vulnerabilidades e deficiências identificadas.

2.4. Acordo de Chaves

Acordos de chaves são esquemas que permitem o estabelecimento de uma chave simétrica compartilhada, por duas ou mais entidades, através de um canal inseguro usando-se de informações públicas.

Os protocolos para acordo de chaves normalmente se baseiam em valores públicos e secretos de longa duração, mas também podem possuir valores públicos e secretos de curta duração (valores efêmeros). Os valores efêmeros são criados a cada novo acordo realizado, normalmente são valores escolhidos aleatoriamente.

Valores públicos efêmeros são trocados durante o início do acordo de chaves, e o momento em que ocorre a troca destes valores é chamado fase de sessão.

Quando o protocolo não possui valores efêmeros dizemos que é um protocolo não interativo, as informações disponíveis para a troca de chaves são valores de longa duração. Os valores públicos podem estar disponíveis em algum repositório, ou entregues a cada usuário no momento de sua adesão ao sistema.

Para corrigir uma deficiência no protocolo para acordo de chaves apresentado em [Smart 2002] foi proposta em [Chen e Kudla 2003] a aplicação de uma função hash ao final do cálculo do valor comum (era a própria chave de sessão), tal função também é conhecida como "função de derivação da chave de sessão", que é adotada por Mandt em seu protocolo.

2.5. Acordo de Chaves sem Certificados

Em [Al-Riyami e Paterson 2003] foram apresentados os conceitos de criptografia de chave pública sem certificado, em seu trabalho estendido é apresentado um protocolo para acordo de chaves descrito sob este conceito. Todas as características de sistemas sem certificados poderá ser vista no texto em referência, porém destacamos algumas características gerais que fazem parte do protocolo proposto por Mandt:

- Dispensa a necessidade da infraestrutura de chaves públicas;
- A chave privada possui duas partes, uma gerada pela autoridade de confiança (KGC) e outra pelo próprio usuário;
- Não há custódia de chaves;
- A chave pública é gerada pelo usuário, a partir de parâmetros públicos;
- A identidade faz parte da chave pública e também é usada para garantir a autenticidade.

2.6. Adversários em Acordos de Chaves sem Certificados

Faz parte da definição do protocolo a modelagem de adversário. A definição dos tipos de adversários feita para o protocolo de Mandt é apresentada abaixo usando-se a nomenclatura em [Swanson 2008]:

Adversários externos
: Não possuem acesso à chave mestra, porém são capazes de substituir chaves públicas;

Adversários internos
: Possuem acesso à chave mestra, mas não são capazes de substituir chaves públicas;

KGC mal intencionado
: Autoridade de confiança que não estabelece honestamente os parâmetros do sistema, e neste caso supomos que seja um adversário interno, e não possa substituir as chaves públicas.

2.7. Segurança em Acordos de Chaves

Além de definir os adversários, Mandt também define alguns atributos de segurança, os quais formalizam as características do protocolo e sugerem um modelo de segurança. Neste texto citamos apenas o atributo que foi usado em [Swanson 2008] para a demonstração da vulnerabilidade do protocolo para um usuário externo:

- Segurança quanto à personificação quando há comprometimento da chave:
  - Se a chave da entidade A foi comprometida, um atacante C não pode se passar por um outro usuário B perante A;
  - Ressalta-se que todos os sistemas não interativos estão sujeitos a este ataque, pois o atacante tem o mesmo poder que A para calcular a chave compartilhada.

Este trabalho mostra como um adversário externo é capaz de personificar o usuário B perante o usuário A, utilizando-se de valores públicos e eventualmente substituindo chaves públicas.

3. Acordo de Chaves de Mandt

A seguir apresentamos os acordos de chaves iniciando pelas definições gerais, valores definidos pela autoridade de confiança, seguidos pelas definições específicas das entidades participantes, cálculo do valor comum entre duas entidades A e B (codinomes para Alice e Beto) e terminamos com a verificação da consistência das chaves.

Definições gerais

- q um valor primo grande;
- $\mathbb{G}_1$ e $\mathbb{G}_2$ grupos cíclicos de ordem q;
- $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ um emparelhamento bilinear admissível.

Definições da autoridade de confiança (KGC)

- Define $\mathbb{G}_1$ e $\mathbb{G}_2$;
- Escolhe $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$;
- Escolhe um valor Q gerador do grupo $\mathbb{G}_1$;
- Calcula o valor público $Q_o = sQ$;
- Escolhe um valor aleatório secreto $s \in \mathbb{Z}_q^*$;
- Escolhe uma função hash $H : \{0,1\}^* \rightarrow \mathbb{G}_1$;
- Escolhe a função de derivação da chave de sessão $f : \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \{0,1\}^n$, onde $n \in \mathbb{N}$.

Definições para uma entidade qualquer (por exemplo $Alice$)

- Define-se o valor público $R_{Alice} = H(ID_{Alice})$
- Recebe de KGC o valor secreto $d_{Alice} = sR_{Alice}$;
- Escolhe um valor aleatório secreto $x_{Alice} \in \mathbb{Z}_q^*$;
- Chave secreta completa $s_{Alice} = \langle x_{Alice}, d_{Alice} \rangle$;
- Chave pública $P_{Alice} = x_{Alice}Q$
- Durante a fase de sessão:
  - Escolhe um valor aleatório $a \in \mathbb{Z}_q^*$ (Beto escolhe um valor b)
  - Calcula e envia o valor público $T_{Alice} = aQ$

Cálculo do valor comum v_{AB} entre as entidades $Alice$ e $Beto$

$$v_{AB} = e(R_{Beto}, P_{Beto} + Q_o)^a \cdot e(x_{Alice}R_{Alice} + d_{Alice}, T_{Beto})$$

Cálculo da chave de sessão k_{AB} entre as entidades $Alice$ e $Beto$

$$\begin{aligned} k_{AB} &= f(v_{AB}, aT_{Beto}, x_{Alice}P_{Beto}) \\ &= f(v_{AB}, abQ, x_{Alice}x_{Beto}Q) \end{aligned}$$

Consistência das chaves de sessão

Para verificar que a chave k_{AB}, calculada pela $Alice$, é igual à chave k_{BA}, calculada pelo $Beto$, basta observar que os valores comuns são iguais:

$$\begin{aligned} v_{AB} &= e(R_{Beto}, P_{Beto} + Q_o)^a && \cdot e(x_{Alice}R_{Alice} + d_{Alice}, T_{Beto}) \\ &= e(R_{Beto}, x_{Beto}Q + sQ)^a && \cdot e(x_{Alice}R_{Alice} + sR_{Alice}, bQ) \\ &= e(R_{Beto}, (x_{Beto} + s)Q)^a && \cdot e((x_{Alice} + s)R_{Alice}, bQ) \\ &= e(R_{Beto}, Q)^{a(x_{Beto}+s)} && \cdot e(R_{Alice}, Q)^{b(x_{Alice}+s)} \\ &= e((x_{Beto} + s)R_{Beto}, aQ) && \cdot e(R_{Alice}, (x_{Alice} + s)Q)^b \\ &= e(x_{Beto}R_{Beto} + sR_{Beto}, aQ) && \cdot e(R_{Alice}, x_{Alice}Q + sQ)^b \\ v_{BA} &= e(x_{Beto}R_{Beto} + d_{Beto}, T_{Alice}) && \cdot e(R_{Alice}, P_{Alice} + Q_o)^b \end{aligned}$$

Com esta igualdade comprova-se que a chave k_{AB} é consistente com a chave k_{BA}.

As considerações de segurança podem ser vistas no artigo original em [Mandt e Tan 2006].

3.1. Vulnerabilidade de Swanson

Em [Swanson 2008] é mostrado como um adversário externo E pode assumir a identidade de um usuário legítimo B perante o usuário A, conhecendo apenas o valor secreto x_A, como apresentado a seguir:

- O adversário E substitui a chave pública de B por $P_B^* = -Q_o + \beta Q$, onde β é um valor aleatório escolhido;
- Envia para o usuário A a chave pública P_B^* e o valor $T_B = bQ$;
- No protocolo original a chave pública P_B é somente o valor $x_B Q$ (que corresponde a um ponto na curva), e portanto A somente verificará se $P_B^* \in \mathbb{G}_1$;
- O usuário A calculará normalmente o valor comum v_{AB}:

$$\begin{aligned}
v_{AB} &= e(R_B, P_B^* + Q_o)^a && \cdot e(x_A R_A + d_A, T_B) \\
&= e(R_B, -Q_o + \beta Q + Q_o)^a && \cdot e(x_A R_A + s R_A, bQ) \\
&= e(R_B, \beta Q)^a && \cdot e((x_A + s) R_A, bQ) \\
&= e(R_B, Q)^{a\beta} && \cdot e(R_A, Q)^{b(x_A + s)} \\
&= e(\beta R_B, aQ) && \cdot e(R_A, (x_A + s)Q)^b \\
&= e(\beta R_B, aQ) && \cdot e(R_A, x_A Q + sQ)^b \\
v_{BA}^* &= e(\beta R_B, T_A) && \cdot e(R_A, P_A + Q_o)^b
\end{aligned}$$

- O valor de x_A é necessário na função de derivação da chave de sessão f, pois o adversário precisa do valor $x_A X_B = x_A x_B Q$.

Esta vulnerabilidade ocorre porque o valor da chave pública do protocolo original de Mandt não é assegurado quanto a sua legitimidade.

3.2. Correção da Vulnerabilidade de Swanson no protocolo de Mandt

Sugerimos como solução para este problema estabeler a chave pública tal qual [Al-Riyami e Paterson 2003], idealizadores do modelo sem certificados:

$$P_A = \langle X_A, Y_A \rangle = \langle x_A Q, x_A Q_o \rangle$$

Então cada usuário deve conferir o emparelhamento a seguir, antes de calcular o valor comum:

$$\begin{array}{ccccc}
e(Q, Y_A) & \stackrel{?}{=} & e(Q_o, X_A) & \text{pois} & \\
e(Q, x_A sQ) & = & e(sQ, x_A Q) & = & e(Q, Q)^{x_A s}
\end{array}$$

Para que o adversário tenha êxito em substituir $P_B = \langle X_B, Y_B \rangle$, ele poderia substituir X_B por $X_B^* = -Q_o + \beta Q$, mas precisaria calcular o valor:

$$Y_B^* = x_B^* sQ = (\beta - s) sQ$$

Para este cálculo o adversário precisa do valor s, contudo este valor está protegido pelo PLD-CE.

Esta vulnerabilidade provavelmente fez com que pesquisadores optassem por usar outros esquemas, e por isto uma deficiência específica para KGCs diferentes tenha ficado oculta. Além disso o emparelhamento bilinear é a operação mais complexa nos acordos de chaves de Mandt e Al-Riyami, e provavelmente a principal vantagem do acordo de chaves proposto em [Mandt e Tan 2006] é ter um número menor de emparelhamentos do que a proposta original de [Al-Riyami e Paterson 2003]. Como a correção implica necessariamente no uso de mais emparelhamentos, não houve interesse pela proposta de Mandt.

3.3. Acordo de Chaves de Mandt para KGCs Diferentes sem Vulnerabilidade de Swanson

Para usuários pertencentes a KGCs diferentes o protocolo é bem similar, o protocolo mostrado a seguir foi modificado da versão original para evitar a vulnerabilidade descrita em [Swanson 2008]:

Definições gerais

- q um valor primo grande;
- $\mathbb{G}_1$ e $\mathbb{G}_2$ grupos cíclicos de ordem q;
- $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ um emparelhamento bilinear admissível.

Definições da autoridade de confiança principal

Não é claro, na definição de Mandt, quem estabelece os parâmetros para o sistema, mas podemos supor que exista uma autoridade de confiança principal que define os seguintes parâmetros:

- Define $\mathbb{G}_1$ e $\mathbb{G}_2$;
- Escolhe $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$;
- Escolhe um valor Q gerador do grupo $\mathbb{G}_1$;
- Escolhe um valor aleatório secreto $s \in \mathbb{Z}_q^*$;
- Calcula o valor público $Q_o = sQ$;
- Escolhe uma função hash $H : \{0,1\}^* \rightarrow \mathbb{G}_1$;
- Escolhe a função de derivação da chave de sessão $f : \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \{0,1\}^n$, onde $n \in \mathbb{N}$.

Definições da autoridade de confiança (KGC$_i$)

- Escolhe um valor aleatório secreto $s_i \in \mathbb{Z}_q^*$;
- Calcula o valor público $Q_i = s_i Q$;
- Calcula o valor público $Q_{o_i} = s_i Q_o$

Definições para uma entidade A pertencente ao KGC$_i$

- Define-se o valor público $R_A = H(ID_A)$
- Recebe de KGC$_i$ o valor secreto $d_{A_i} = s_i R_A$;
- Escolhe um valor aleatório secreto $x_A \in \mathbb{Z}_q^*$;
- Chave secreta completa $s_A = \langle x_A, d_{A_i} \rangle$;
- Chave pública $P_A = \langle X_A, Y_A \rangle = \langle x_A Q, x_A Q_o \rangle$
- Durante a fase de sessão:
  - Escolhe um valor aleatório $a \in \mathbb{Z}_q^*$
  - Calcula e envia o valor público $T_A = aQ$

Cálculo do valor comum v_{AB} pela entidade A do KGC_1 para com B do KGC_2

- Verifica a validade da chave pública de B e KGC$_2$:
  $e(Q, Y_B) \stackrel{?}{=} e(Q_o, X_B)$
  $e(Q, Q_{o_2}) \stackrel{?}{=} e(Q_o, Q_2)$
- Se os emparelhamento forem iguais, calcula o valor comum:

$$v_{AB} = e(R_B, X_B + Q_2)^a \cdot e(x_A R_A + d_{A_1}, T_B)$$

Cálculo da chave de sessão k_{AB} entre as entidades A e B

$$\begin{aligned} k_{AB} &= f(v_{AB}, aT_B, x_A X_B) \\ &= f(v_{AB}, abQ, x_A x_B Q) \end{aligned}$$

Consistência das chaves de sessão

Não há alterações na função de derivação da chave de sessão, por isso só precisamos verificar que o valor comum calculado pela entidade A é igual ao valor comum calculado pela entidade B:

$$\begin{aligned} v_{AB} &= e(R_B, X_B + Q_2)^a && \cdot e(x_A R_A + d_{A_1}, T_B) \\ &= e(R_B, x_B Q + s_2 Q)^a && \cdot e(x_A R_A + s_1 R_A, bQ) \\ &= e(R_B, (x_B + s_2)Q)^a && \cdot e((x_A + s_1)R_A, bQ) \\ &= e(R_B, Q)^{a(x_B + s_2)} && \cdot e(R_A, Q)^{b(x_A + s_1)} \\ &= e((x_B + s_2)R_B, aQ) && \cdot e(R_A, (x_A + s_1)Q)^b \\ &= e(x_B R_B + s_2 R_B, aQ) && \cdot e(R_A, x_A Q + s_1 Q)^b \\ v_{BA} &= e(x_B R_B + d_{B_2}, T_A) && \cdot e(R_A, X_A + Q_1)^b \end{aligned}$$

3.4. Deficiência no Protocolo de Mandt para KGCs Diferentes

Apresentamos como um adversário externo E pode assumir a identidade de um usuário legítimo B perante o usuário A, desde que estejam sob KGCs diferentes:

- O adversário E escolhe aleatoriamente um valor $s_E \in \mathbb{Z}_q^*$;
- Constrói um falso KGC$_E$ e lhe atribui o valor de $Q_E = s_E Q$ e $Q_{oE} = s_E Q_o$, substituindo a chave pública do verdadeiro KGC da entidade B;
- Calcula sua chave pública como $P_B^* = \langle X_B^*, Y_B^* \rangle = \langle x_B^* Q, x_B^* Q_o \rangle$
- Envia normalmente para a entidade A a chave pública P_B^* e o valor $T_B^* = bQ$
- A entidade A irá verificar a chave pública com os emparelhamentos:
  $e(Q, Y_B^*) \stackrel{?}{=} e(Q_o, X_B^*)$
  $e(Q, Q_{o_E}) \stackrel{?}{=} e(Q_o, Q_E)$;
- A verificação é válida;
- A entidade A do KGC$_1$ irá calcular a chave de sessão normalmente com a entidade E, sem perceber que o KGC usado pelo adversário é falso, como segue:

$$\begin{aligned} v_{AB} &= e(R_B, X_B^* + Q_E)^a && \cdot e(x_A R_A + d_{A_1}, T_B^*) \\ &= e(R_B, x_B^* Q + s_E Q)^a && \cdot e(x_A R_A + s_1 R_A, bQ) \\ &= e(R_B, (x_B^* + s_E)Q)^a && \cdot e((x_A + s_1)R_A, bQ) \\ &= e(R_B, Q)^{a(x_B^* + s_E)} && \cdot e(R_A, Q)^{b(x_A + s_1)} \\ &= e((x_B^* + s_E)R_B, aQ) && \cdot e(R_A, (x_A + s_1)Q)^b \\ &= e(x_B^* R_B + s_E R_B, aQ) && \cdot e(R_A, x_A Q + s_1 Q)^b \\ v_{BA} &= e(x_B^* R_B + s_E R_B, T_A) && \cdot e(R_A, X_A + Q_1)^b \end{aligned}$$

A solução trivial para evitar esta deficiência é entregar para todas as entidades as chaves públicas de todos os KGCs. Contudo isso deve ser feito através de um canal autêntico. Para garantir autenticidade normalmente se usa algum tipo de certificação, mas se for usada a certificação tradicional, seriam perdidas as vantagens de um sistema sem certificado. Daí surge a questão, por que não usar o próprio algoritmo para garantir a certificação dos KGCs? A resposta a este questionamento é a base da nossa solução proposta.

Esta deficiência ocorre porque não existem relações privadas entre as chaves secretas dos KGCs, isto permite que qualquer participante do sistema gere seu próprio KGC.

4. Correção da Deficiência para KGCs Diferentes

Nossa proposta para corrigir esta deficiência é construir uma relação privada entre cada novo KGC do sistema e uma autoridade de confiança comum a todos os usuários, e todos os KGCs deverão ser construídos a partir desta autoridade de confiança. É desejável que após criado um novo KGC, este tenha independência da autoridade de confiança que o criou, podendo gerar novas entidades participantes do sistema, as quais serão autenticadas através de valores públicos. Vejamos como isto pode ser feito:

Definições gerais

- q um valor primo grande;
- $\mathbb{G}_1$ e $\mathbb{G}_2$ grupos cíclicos de ordem q;
- $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ um emparelhamento bilinear admissível.

Definições da autoridade de confiança comum (KGC$_o$)

- Define $\mathbb{G}_1$ e $\mathbb{G}_2$;
- Escolhe $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$;
- Escolhe um valor Q gerador do grupo $\mathbb{G}_1$;
- Escolhe um valor aleatório secreto $s_o \in \mathbb{Z}_q^*$;
- Calcula o valor público $Q_o = s_o Q$;
- Escolhe uma função hash $H : \{0,1\}^* \rightarrow \mathbb{G}_1$;
- Escolhe a função de derivação da chave de sessão $f : \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \{0,1\}^n$, onde $n \in \mathbb{N}$.

Definições da autoridade de confiança (KGC$_i$)

- Define-se o valor público $R_{KGC_i} = H(ID_{KGC_i})$
- Recebe de KGC_o o valor secreto $d_{_{KGC_i}} = s_o R_{KGC_i}$;
- Escolhe um valor aleatório secreto $s_i \in \mathbb{Z}_q^*$;
- Chave secreta completa $s_{_{KGC_i}} = \langle s_i, d_{_{KGC_i}} \rangle$;
- Chave pública $P_{KGC_i} = \langle X_{KGC_i}, Y_{KGC_i} \rangle = \langle s_i Q, s_i Q_o \rangle$

Definições para uma entidade A pertencente ao KGC$_i$

- Define-se o valor público $R_A = H(ID_A)$
- Recebe de KGC$_i$ o valor secreto $d_{_{A_i}} = s_i R_A + d_{_{KGC_i}}$;
- Escolhe um valor aleatório secreto $x_{_A} \in \mathbb{Z}_q^*$;
- Chave secreta completa $s_{_A} = \langle x_{_A}, d_{_{A_i}} \rangle$;
- Chave pública $P_A = \langle X_A, Y_A \rangle = \langle x_{_A} Q, x_{_A} Q_o \rangle$
- Durante a fase de sessão:
  - Escolhe um valor aleatório $a \in \mathbb{Z}_q^*$
  - Calcula e envia o valor público $T_A = aQ$

Cálculo do valor comum $v_{_{AB}}$ pela entidade A do KGC_1 para com B do KGC_2

- Verifica a chave pública de B e KGC$_2$:
  $e(Q, Y_B) \stackrel{?}{=} e(Q_o, X_B)$
  $e(Q, Y_{KGC_2}) \stackrel{?}{=} e(Q_o, X_{KGC_2})$
- Se a verificação for válida, calcula o valor comum:
  $$v_{_{AB}} = [e\,(R_B, X_B + X_{KGC_2}) \cdot e\,(R_{KGC_2}, Q_o)]^a \cdot e(x_{_A} R_A + d_{_{A_1}}, T_B)$$

O cálculo da chave comum utiliza um emparelhamento a mais, que é a conferência da relação do KGC_2 e o KGC_o, neste novo emparelhamento permanece a exponenciação do valor efêmero a, que continua protegido pela dificuldade dos problemas BDH e DBDH.

Cálculo da chave de sessão k_{AB} entre as entidades A e B

$$\begin{aligned} k_{AB} &= f(v_{AB}, aT_B, x_A X_B) \\ &= f(v_{AB}, abQ, x_A x_B Q) \end{aligned}$$

Consistência das chaves de sessão

De igual forma não há alterações na função de derivação da chave de sessão, por isso só precisamos verificar que o valor comum calculado pela entidade A é igual ao valor comum calculado pela entidade B:

$$\begin{aligned} K'_{AB} &= [e(R_B, X_B + X_{KGC_2}) \cdot e(R_{KGC_2}, Q_o)]^a \cdot e(x_A R_A + d_{A_1}, T_B) \\ &= e(R_B, x_B Q + s_2 Q)^a \cdot e(R_{KGC_2}, s_o Q)^a \cdot e(x_A R_A + s_1 R_A + d_{KGC_1}, bQ) \\ &= e(R_B, (x_B + s_2)Q)^a \cdot e(R_{KGC_2}, s_o Q)^a \cdot e((x_A + s_1)R_A + s_o R_{KGC_1}, bQ) \\ &= e((x_B + s_2)R_B, aQ) \cdot e(s_o R_{KGC_2}, aQ) \cdot e((x_A + s_1)R_A, bQ) \cdot e(s_o R_{KGC_1}, bQ) \\ &= e((x_B + s_2)R_B + s_o R_{KGC_2}, aQ) \cdot e((x_A + s_1)R_A, Q)^b \cdot e(s_o R_{KGC_1}, Q)^b \\ &= e(x_B R_B + s_2 R_B + s_o R_{KGC_2}, aQ) \cdot e(R_A, (x_A + s_1)Q)^b \cdot e(R_{KGC_1}, s_o Q)^b \\ &= e(x_B R_B + (s_2 R_B + s_o R_{KGC_2}), aQ) \cdot e(R_A, x_A Q + s_1 Q)^b \cdot e(R_{KGC_1}, s_o Q)^b \\ K'_{BA} &= e(x_B R_B + d_{B_2}, T_A) \cdot [e(R_A, X_A + X_{KGC_1}) \cdot e(R_{KGC_1}, Q_o)]^b \end{aligned}$$

Verificamos que a nova proposta continua sendo consistente. E ainda, conseguimos proteger nosso sistema contra a criação de um falso KGC^*, pois os valores d_{KGC_i} ou d_A são secretos, e os valores a e s_o estão protegidos no emparelhamento $e(R_{KGC_2}, Q_o)^a$ pelo BDH.

4.1. Uso em Esquemas Hierárquicos

A relação criada entre os KGC_o e os KGC_is pode ser estendida para níveis hierárquicos, onde o KGC_o define os parâmetros do sistema, e cada $KGC_i^{(l)}$ no nível l passa para o $KGC_j^{(l+1)}$ no nível $l+1$ uma chave parcial secreta $d_{KGC_j^{(l+1)}} = s_{KGC_i^{(l)}} H(ID_{KGC_i^{l+1}}) + d_{KGC_i^l}$. A chave comum será calculada usando-se um novo emparelhamento para cada nível, a demonstração pode ser vista no apêndice.

Embora nossa solução use quatro novos emparelhamentos, tornando o esquema original mais complexo que o esquema original de Al-Riyami; esta nova abordagem permitirá a construção de acordos de chaves hierárquicos onde a complexidade aumenta linearmente com a profundidade da entidade de nível mais baixo. Além disso os cálculos dos emparelhamentos são independentes, podendo-se realizá-los em paralelo, tornando-o eficientes em sistemas multiprocessados.

5. Conclusão

Neste trabalho apresentamos uma deficiência no protocolo de acordo de chaves apresentado em [Mandt e Tan 2006] para operação com dois KGCs. Mostramos uma possível solução para esta deficiência, possíveis correção para a vulnerabilidade descrita em [Swanson 2008], baseadas na dificuldade de resolver o problema do logaritmo discreto sobre curvas elípticas e do problema de Diffie-Hellman bilinear, mantendo o algoritmo com os mesmos atributos de segurança descritos pelo autor.

O uso em esquemas hierárquicos pode ser ainda explorado para uso com maior eficiência, visto que não apresenta custódia de chaves, é um esquema bastante seguro contra comprometimento de nós.

As correções ainda precisam de uma formalização com um modelo de segurança forte, tal como o modelo de [Swanson 2008].

É comum os autores de esquemas de acordo de chaves demonstrar a segurança para o protocolo base, sugerir a utilização para múltiplas autoridades de confiança e não demonstrar a segurança, a identificação desta deficiência também contribui para motivar os autores a desenvolverem demonstrações de segurança para seus protocolos estendidos.

Referências

Al-Riyami, S., Paterson, K. G. (2003). Certificateless Public Key Cryptography. In *Asiacrypt'03 - LCNS*, pages 452–473, Taipei - Taiwan. Springer Berlin. Extensão em `http://eprint.iacr.org/2003/126`.

Blundo, C., Santis, D. A., Herzberg, A., Kutten, S., Vaccaro, U.,, Yung, M. (1993). Perfectly secure key distribution for dynamic conferences. In *LNCS - Crypto'92*, pages 471–486. Springer-Berlin. v.740.

Boneh, D., Franklin, M. (2001). Identity Based Encryption from Weil Pairing. In *Crypto'01 - LNSC*, volume 2139, pages 213–229, Santa Barbara, California, USA. Springer Berlin.

Castro, R., Dahab, R. D.,, Devegili, A. J. (2007). *VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais: Minicursos do SBSeg 2007*, chapter Introdução à Segurança Demonstrável, pages 103–152. UFRJ/NCE, Rio de Janeiro.

Chen, L., Kudla, C. (2003). Identity-based authenticated key agreement protocols from pairings. In *Proceedings 16th IEEE Security Foundations Workshop*, pages 219–233.

Koblitz, N. (1994). *A course in number theory and cryptography*. Springer-Verlag, New York - NY - USA, 2 edition.

Mandt, T. K., Tan, C. H. (2006). Certificateless Authenticated Two-Party Key Agreement Protocols. In *11th Asian Computing Science Conference'06*, volume 4435, pages 37–44, Tokyo - Japan. Springer Berlin.

Sakai, R., Ohgishi, K.,, Kasahara, M. (2000). Cryptosystems based on pairing. In *Symposium on Cryptography and Information Security, SCIS2000*, pages 26–28, Okinawa - Japan.

Singh, S. (2000). *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. Anchor, New York, rev. edition.

Smart, N. P. (2002). An identity based authenticated key agreement protocol based on the weil pairing. In *Electronics Letters*, pages 630–632. Springer Berlin. v.38.

Strangio, M. A. (2006). On the resilience of key agreement protocols to key compromise impersonation. In *LNCS - EuroPKI'06*, pages 233–247. Springer-Berlin. v.4043.

Swanson, C. M. (2008). Security in key agreement: Two-party certificateless schemes. Master's thesis, University of Waterloo - Canadá. `http://hdl.handle.net/10012/4156`.

Terada, R. (2008). *Segurança de Dados - Criptografia em redes de computador*. Blucher, 2 edition.

Apêndice - Construção do Esquema Hierárquico

Definições gerais

- q um valor primo grande;
- $\mathbb{G}_1$ e $\mathbb{G}_2$ grupos cíclicos de ordem q;
- $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ um emparelhamento bilinear admissível;
- Convencionamos que a entidade $A_{(l-1)}$ no nível $(l-1)$ é a entidade geradora da entidade $A_{(l)}$ no nível l.

Definições da autoridade de confiança raíz (KGC$_o$ ou $A_{(0)}$)

- Define $\mathbb{G}_1$ e $\mathbb{G}_2$;
- Escolhe $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$;
- Escolhe um valor Q gerador do grupo $\mathbb{G}_1$;
- Escolhe um valor aleatório secreto $s_o \in \mathbb{Z}_q^*$;
- Calcula o valor público $Q_o = s_o Q = X_{A_{(0)}}$;
- Escolhe uma função hash $H : \{0,1\}^* \rightarrow \mathbb{G}_1$;
- Escolhe a função de derivação da chave de sessão $f : \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \{0,1\}^n$, onde $n \in \mathbb{N}$.

Definições para uma entidade $A_{(l)}$ no nível l

- Define-se o valor público $R_{A_{(l)}} = H(ID_{A_{(l)}})$
- Recebe de $A_{(l-1)}$ o valor secreto $d_{A_{(l)}} = s_{A_{(l-1)}} R_{A_{(l)}} + d_{A_{(l-1)}} = \sum_{m=1}^{l} s_{A_{(m-1)}} R_{A_{(m)}}$;
- Escolhe um valor aleatório secreto $s_{A_{(l)}} \in \mathbb{Z}_q^*$;
- Chave secreta completa $\xi_{A_{(l)}} = \langle s_{A_{(l)}}, d_{A_{(l)}} \rangle$;
- Chave pública $P_{A_{(l)}} = \langle X_{A_{(l)}}, Y_{A_{(l)}} \rangle = \langle s_{A_{(l)}} Q, s_{A_{(l)}} Q_o \rangle$
- Durante a fase de sessão:
  - Escolhe um valor aleatório $a \in \mathbb{Z}_q^*$
  - Calcula e envia o valor público $T_{A_{(l)}} = aQ$

Cálculo do valor comum $v_{A_{(l)}B_{(u)}}$ pela entidade $A_{(l)}$ para com $B_{(u)}$

- Verifica todos emparelhamentos de $t = 1$ até $t = u$:
  $e(Q, Y_{B_{(t)}}) = e(Q_o, X_{B_{(t)}})$
- Se todos emparelhamento forem válidos, calcula o valor comum:

$$v_{A_{(l)}B_{(u)}} = \left[e\left(R_{B_{(u)}}, X_{B_{(u)}} + X_{B_{(u-1)}}\right) \cdot \left(\prod_{z=1}^{u-1} e\left(R_{B_{(z)}}, X_{B_{(z-1)}}\right) \right) \right]^a \cdot e(s_{A_{(l)}} R_{A_{(l)}} + d_{A_{(l)}}, T_{B_{(u)}})$$

Cálculo da chave de sessão $k_{A_{(l)}B_{(u)}}$ entre as entidades $A_{(l)}$ e $B_{(u)}$

$$\begin{aligned} k_{A_l B_u} &= f(v_{A_{(l)}B_{(u)}}, aT_{B_{(u)}}, x_{A_{(l)}} X_{B_{(u)}}) \\ &= f(v_{A_{(l)}B_{(u)}}, abQ, x_{A_{(l)}} x_{B_{(u)}} Q) \end{aligned}$$

Consistência das chaves de sessão

De igual forma não há alterações na função de derivação da chave de sessão, por isso só precisamos verificar se o valor comum calculado pela entidade $A_{(l)}$ é igual ao valor calculado pela entidade $B_{(u)}$:

$$
\begin{aligned}
v_{A_{(l)}B_{(u)}} &= \left[e\left(R_{B_{(u)}}, X_{B_{(u)}} + X_{B_{(u-1)}}\right) \cdot \prod_{z=1}^{u-1} e\left(R_{B_{(z)}}, X_{B_{(z-1)}}\right) \right]^{a} \cdot e\left(s_{A_{(l)}} R_{A_{(l)}} + d_{A_{(l)}}, T_{B_{(u)}}\right) \\
&= \left[e\left(R_{B_{(u)}}, s_{B_{(u)}} Q + s_{B_{(u-1)}} Q\right) \cdot \prod_{z=1}^{u-1} e\left(R_{B_{(z)}}, s_{B_{(z-1)}} Q\right) \right]^{a} \cdot e\left(s_{A_{(l)}} R_{A_{(l)}} + d_{A_{(l)}}, T_{B_{(u)}}\right) \\
&= \left[e\left(R_{B_{(u)}}, (s_{B_{(u)}} + s_{B_{(u-1)}})Q\right) \cdot \prod_{z=1}^{u-1} e\left(R_{B_{(z)}}, s_{B_{(z-1)}} Q\right) \right]^{a} \cdot e\left(s_{A_{(l)}} R_{A_{(l)}} + d_{A_{(l)}}, T_{B_{(u)}}\right) \\
&= \left[e\left((s_{B_{(u)}} + s_{B_{(u-1)}})R_{B_{(u)}}, Q\right) \cdot \prod_{z=1}^{u-1} e\left(s_{B_{(z-1)}} R_{B_{(z)}}, Q\right) \right]^{a} \cdot e\left(s_{A_{(l)}} R_{A_{(l)}} + d_{A_{(l)}}, T_{B_{(u)}}\right) \\
&= e\left((s_{B_{(u)}} + s_{B_{(u-1)}})R_{B_{(u)}} + \sum_{z=1}^{u-1} s_{B_{(z-1)}} R_{B_{(z)}}, Q\right)^{a} \cdot e\left(s_{A_{(l)}} R_{A_{(l)}} + d_{A_{(l)}}, T_{B_{(u)}}\right) \\
&= e\left(s_{B_{(u)}} R_{B_{(u)}} + s_{B_{(u-1)}} R_{B_{(u)}} + \sum_{z=1}^{u-1} s_{B_{(z-1)}} R_{B_{(z)}}, aQ\right) \cdot e\left(s_{A_{(l)}} R_{A_{(l)}} + d_{A_{(l)}}, T_{B_{(u)}}\right) \\
&= e\left(s_{B_{(u)}} R_{B_{(u)}} + \sum_{z=1}^{u} s_{B_{(z-1)}} R_{B_{(z)}}, T_{A_{(l)}}\right) \cdot e\left(s_{A_{(l)}} R_{A_{(l)}} + d_{A_{(l)}}, T_{B_{(u)}}\right) \\
&= e\left(s_{B_{(u)}} R_{B_{(u)}} + d_{B_{(u)}}, T_{A_{(l)}}\right) \cdot e\left(s_{A_{(l)}} R_{A_{(l)}} + d_{A_{(l)}}, T_{B_{(u)}}\right) \\
&= e\left(s_{B_{(u)}} R_{B_{(u)}} + d_{B_{(u)}}, T_{A_{(l)}}\right) \cdot e\left(s_{A_{(l)}} R_{A_{(l)}} + \sum_{m=1}^{l} s_{A_{(m-1)}} R_{A_{(m)}}, bQ\right) \\
&= e\left(s_{B_{(u)}} R_{B_{(u)}} + d_{B_{(u)}}, T_{A_{(l)}}\right) \cdot e\left(s_{A_{(l)}} R_{A_{(l)}} + s_{A_{(l-1)}} R_{A_{(l)}} + \sum_{m=1}^{l-1} s_{A_{(m-1)}} R_{A_{(m)}}, Q\right)^{b} \\
&= e\left(s_{B_{(u)}} R_{B_{(u)}} + d_{B_{(u)}}, T_{A_{(l)}}\right) \cdot e\left((s_{A_{(l)}} + s_{A_{(l-1)}})R_{A_{(l)}} + \sum_{m=1}^{l-1} s_{A_{(m-1)}} R_{A_{(m)}}, Q\right)^{b} \\
&= e\left(s_{B_{(u)}} R_{B_{(u)}} + d_{B_{(u)}}, T_{A_{(l)}}\right) \cdot \left[e\left((s_{A_{(l)}} + s_{A_{(l-1)}})R_{A_{(l)}}, Q\right) \cdot \prod_{m=1}^{l-1} e\left(s_{A_{(m-1)}} R_{A_{(m)}}, Q\right) \right]^{b} \\
&= e\left(s_{B_{(u)}} R_{B_{(u)}} + d_{B_{(u)}}, T_{A_{(l)}}\right) \cdot \left[e\left(R_{A_{(l)}}, (s_{A_{(l)}} + s_{A_{(l-1)}})Q\right) \cdot \prod_{m=1}^{l-1} e\left(R_{A_{(m)}}, s_{A_{(m-1)}} Q\right) \right]^{b} \\
&= e\left(s_{B_{(u)}} R_{B_{(u)}} + d_{B_{(u)}}, T_{A_{(l)}}\right) \cdot \left[e\left(R_{A_{(l)}}, s_{A_{(l)}} Q + s_{A_{(l-1)}} Q\right) \cdot \prod_{m=1}^{l-1} e\left(R_{A_{(m)}}, s_{A_{(m-1)}} Q\right) \right]^{b} \\
v_{B_{(u)}A_{(l)}} &= e\left(s_{B_{(u)}} R_{B_{(u)}} + d_{B_{(u)}}, T_{A_{(l)}}\right) \cdot \left[e\left(R_{A_{(l)}}, X_{A_{(l)}} + X_{A_{(l-1)}}\right) \cdot \prod_{m=1}^{l-1} e\left(R_{A_{(m)}}, X_{A_{(m-1)}}\right) \right]^{b}
\end{aligned}
$$