Key Agreement in dynamic Peer Groups

von Dienersberger Maximilian

Motivation:

Globale Vernetzung nimmt immer mehr Einzug in unsere Welt und eine einfache eins zu eins Kommunikation ist bereits in vielen Bereichen längst überholt. Was den Sicherheitsaspekt betrifft wird der Focus meist nur auf Peer-to-Peer Beziehungen gesetzt, da Gruppen-Sicherheit als triviale Erweiterung des zwei-Personen-Falls gesehen wird. Dies lässt einerseits den Aspekt außer acht, dass bei großen Gruppen weitaus effizientere Protokolle benötigt werden und andererseits die Dynamik einer Gruppe eine Rolle spielt.

Zielsetzung:

Unser Ziel ist es ein Verfahren zu finden um eine sichere (und effiziente) Erstellung einer Gruppe zu ermöglichen und folgende Gruppen-Operationen sicher zu realisieren:

- Hinzufügen einer oder mehrerer Personen
- Entfernen eines oder mehrerer Mitglieder

Lösungsansatz des Initialisierungsproblems:

Um dieses Problem zu lösen erweitern wir den 2-Personen-Diffie-Hellmann Ansatz auf mehrere Personen:

Alle Teilnehmer $\mathbf{M_1,M_2,\ldots,M_n}$ stimmen einer Zyklischen Gruppe **G** zu. Bei dieser **α** der Generator ist. Nun wählt jedes Mitglied ein $\mathbf{N_i} \in \mathbb{Z}_q$. Der Gruppenschlüssel beträgt dann somit $\boldsymbol{\alpha}^{N_1 \ldots N_n}$.

CLIQUES: IKA.1

Das IKA(Initial Key Agreement) Protokoll ist für die Generierung der Gruppe zuständig. Der Focus liegt auf den Sicherheitsaspekt, da die Generierung nur einmal stattfindet.

Jeder M_i führt folgende Operationen aus:

- Generiert privaten Exponenten
- Setzt $INT(i,j) = (INT(i-1,j))^{N_i}$
- Setzt $INT(i,i) = CRD_{i-1}$
- Setzt $CRD_i = (CRD_{i-1})^{N_i}$

$M_i \qquad\qquad M_{i+1}$

$\{\alpha^{\Pi(N_p | p \in [1,i] \wedge p \neq j)} | \; j \in [1,i]\}, \alpha^{\Pi(N_p | p \in [1,i])}$ →

Stage 1 (Upflow): round i; $i \in [1, n-1]$

Das letzte Mitglied berechnet zuerst den Gruppenschlüssel und sendet alle Teilschlüssel an alle anderen Mitglieder, welche mit diesen und ihren geheimen Exponenten den Schlüssel errechnen können.

$M_i \qquad\qquad M_n$

← $\{\alpha^{\Pi(N_p | p \in [1,n] \wedge p \neq j)} | \; j \in [1,n]\}$

Stage 2 (Broadcast): round n

Ein Problem dieses Algorithmus ist die nicht-lineare Anzahl an Potenzen, was bedeutet, dass es bei einer enormen Anzahl an Gruppenmitgliedern zu lange Laufzeit benötigt um die Gruppe zu erstellen.

CLIQUES: IKA.2

Idee: Anfängliche Potenzen sparen und durch einen zusätzlichen Broadcast in der vorletzten Runde ersetzen.

Statt $\frac{(n+3)n}{2} - 1$ Potenzen nur noch $5n - 6$.

Jedoch n-1 Nachrichten die der vorletzte Teilnehmer zum gleichen Zeitpunkt erhält, was zu einem Stau führen kann.

$M_i \qquad\qquad M_{i+1}$

$\xrightarrow{\alpha^{\Pi\{N_p \mid p\in[1,i]\}}}$

Stage 1 (Upflow): Round i; $i \in [1, n-2]$

$M_i \qquad\qquad M_{n-1}$

$\xleftarrow{\alpha^{\Pi\{N_p \mid p\in[1,n-1]\}}}$

Stage 2 (Broadcast): Round $n-1$

$M_i \qquad\qquad M_n$

$\xrightarrow{\alpha^{\frac{\Pi\{N_p \mid p\in[1,n-1]\}}{N_i}\}}}$

Stage 3 (Response): Round n

$M_i \qquad\qquad M_n$

$\xleftarrow{\{\alpha^{\frac{\Pi(N_p \mid p\in[1,n]}{N_j}} \mid j \in [1,n]\}}$

Stage 4 (Broadcast): Round $n+1$

Sicherheit von IKA:

Die Sicherheit des IKA leitet sich von 2-Personen Diffie-Hellman Ansatz ab.

CDH (Computitional-Diffie-Hellman):
Der Angreifer besitzt $\boldsymbol{\alpha}^{N_1}$, $\boldsymbol{\alpha}^{N_2}$ und möchte $\boldsymbol{\alpha}^{N_1 N_2}$ berechnen.

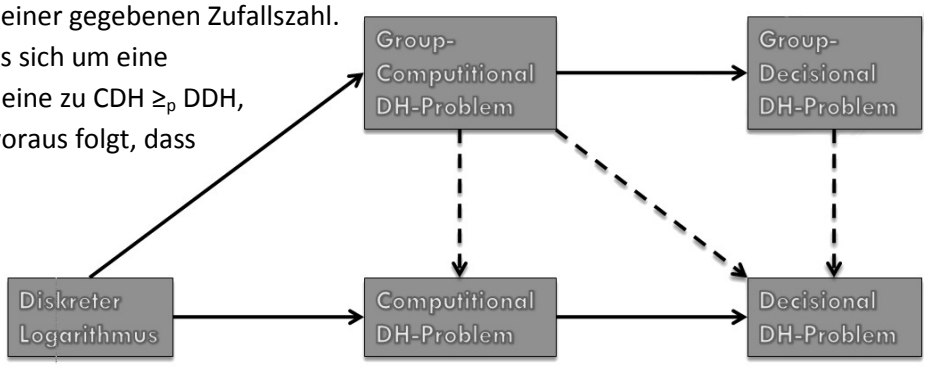
Reduzierbar auf den diskreten Logarithmus, d.h.
CDH ist über den Sicherheitsparameter der die Größe der Gruppe darstellt hart.
Dies bedeutet CDH ist semantisch sicher unter der Voraussetzung des Random-Oracle-Modells.

DDH(Decisional-Diffie-Hellman):
Der Angreifer besitzt $\boldsymbol{\alpha}^{N_1}$, $\boldsymbol{\alpha}^{N_2}$ und möchte $\boldsymbol{\alpha}^{N_1 N_2}$ von $\boldsymbol{\alpha}^{X}$ unterscheiden wobei X eine Zufallszahl ist.
Dazu benötigt man das Konzept der polynomiellen Ununterscheidbarkeit. Wonach ein Verfahren sicher ist, wenn es keinen Algorithmus gibt, der in polynomieller Laufzeit zwischen einem Diffie-Hellman Schlüssel und einer Zufallszahl mit einer Wahrscheinlichkeit signifikant größer als ½ unterscheiden kann. Dies trifft für DDH zu und ist eine stärkere Bedingung als bei CDH woraus diese Beziehung folgt:
DL $\geq_p$ CDH $\geq_p$ DDH

GCDH(Generalized-Computitional-Diffie-Hellman):
Berechne den letzten Wert in dem Set, der dem Schlüssel entspricht.
Führt wieder auf DL zurück, woraus folgt, dass GCDH hart ist.

$\boldsymbol{\alpha}$, $\boldsymbol{\alpha}^{N_1}$
$\boldsymbol{\alpha}^{N_1}$, $\boldsymbol{\alpha}^{N_2}$, $\boldsymbol{\alpha}^{N_1 N_2}$
$\boldsymbol{\alpha}^{N_1 N_2}$, $\boldsymbol{\alpha}^{N_2 N_3}$, $\boldsymbol{\alpha}^{N_1 N_3}$, $\boldsymbol{\alpha}^{N_1 N_2 N_3}$
$\boldsymbol{\alpha}^{N_1 N_2 N_3}$, $\boldsymbol{\alpha}^{N_2 N_3 N_4}$, $\boldsymbol{\alpha}^{N_1 N_3 N_4}$, $\boldsymbol{\alpha}^{N_2 N_3 N_4}$, $\boldsymbol{\alpha}^{N_1 N_2 N_3 N_4}$

GDDH(Generalized-Decisional-Diffie-Hellman auch natural-Extension of the Diffie-Hellman-Problem):
Unterscheide den letzten Wert von einer gegebenen Zufallszahl.
Führt wieder auf GCDH zurück, da es sich um eine stärkere Bedingung handelt, womit eine zu CDH $\geq_p$ DDH, parallele Reduzierung stattfindet, woraus folgt, dass GDDH hart ist.



AKA (Auxiliary Key Agreement):

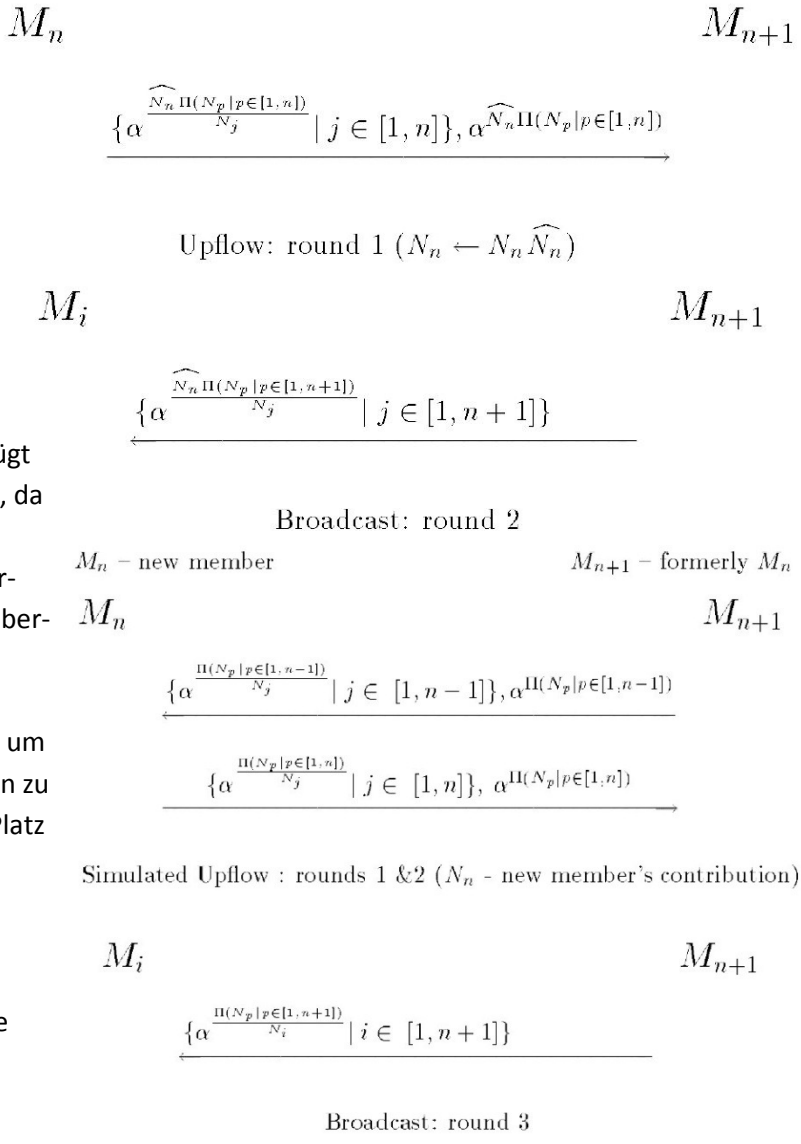
Unter AKA sind alle Operationen vereint, welche die Dynamik einer Gruppe regeln. Alle Operationen müssen beliebig oft während der Gruppenexistenz ausgeführt werden können. Jede Operation muss gewährleisten, dass Mitglieder, die aufgenommen werden, kein Vorwissen in Bezug auf den Schlüssel besitzen und Mitglieder, welche ausgeschlossen werden, mit ihren Informationen keine Möglichkeit haben den neuen Schlüssel einfach zu berechnen.

AKA: Member Addition

Um diese Operationen auszuführen benötigt man das Konzept des Gruppenleiters, der festgelegt wird als das Mitglied, das den Broadcast ausführt.
Es gibt zwei Möglichkeiten für das hinzufügen eines neuen Mitglieds:

a) Fließender Gruppenleiter:
   Das neue Mitglied wird am Ende angefügt und wird somit der neue Gruppenleiter, da er nun den Broadcast ausführt.
   Jedoch wird hier dem am wenigsten vertrauenswürdigen Mitglied die Leitung übertragen.

b) Fester Gruppenleiter:
   Der alte Leiter erhält den letzten Index, um auch weiterhin den Broadcast ausführen zu können. Das neue Mitglied erhält den Platz des vorherigen Gruppenleiters.



AKA: Mass Join

Durch die Mass Join Operation werden mehrere Mitglieder zugleich hinzugefügt. Eine Möglichkeit der Lösung ist eine aneinander Kettung mehrerer Member Addition Operationen.
Jedoch werden hierzu viele Broadcasts durchgeführt, deshalb kettet man den Upflow Teil mehrerer Member Additions aneinander und führt zuletzt nur einen Broadcast aus.

AKA: Group Fusion

Eine Group Fusion ist prinzipiell ein Mass Join, wobei man in dem Fall abwiegen kann ob es nicht sogar sinnvoll ist einen neuen IKA-Durchlauf auszuführen, da alle Schlüsselanteile erneuert werden. Für welche Methode man sich entscheidet hängt von dem Größenverhältnis der beiden Gruppen ab. Ist die einzugliedernde Gruppe klein so eignet sich ein Mass Join, ist sie sehr groß kann man einen neuen IKA-Durchlauf erlauben.

AKA: Member Exclusion

Das entfernen eines Mitglieds muss unter der Voraussetzung erfolgen, dass er mit seinen bekannten Anteilen und dem Schlüssel nicht mehr an der Gruppenkommunikation teilnehmen kann.

Der Gruppenleiter ändert seinen Schlüsselanteil N_n zu $N_n\widehat{N_n}$ wobei $\widehat{N_n}$ eine neue Zufallszahl aus $\mathbb{Z}_q$ ist. Anschließend wird ein neuer Broadcast ausgeführt, so dass der Ausgeschlossene keinen Teil-schlüssel erhält. Falls der entfernte Mitglieder der Gruppenleiter war, führt der vorletzte Index die oben genannten Operationen aus.

M_n M_i

$$\{\alpha^{\frac{\widehat{N_n}\,\Pi(N_p \mid p\in[1,n])}{N_j}} \mid j \in [1,n] \wedge j \neq d\} \longrightarrow$$

$$\text{Broadcast: round 1 } (N_n \leftarrow N_n\widehat{N_n})$$

AKA: Subgroup Exclusion

Anstatt nur einem Mitglieder den Schlüssel nicht mehr zu schicken, wird dies bei mehreren Mitgliedern vollzogen. Anschließend legt man bei den entfernten Teilnehmern einen neuen Gruppenleiter fest ,der ebenfalls seinen Anteil erneuert und einen Broadcast an alle Ausgeschlossenen sendet.

AKA: Key Refresh

Ein regelmäßiges Erneuern des Schlüsselanteils, eines jeden Teilnehmers, schützt erstens besser vor Plain- bzw. Ciphertextangriffen und zweitens wird so ein geknackter Schlüssel erneuert um den Angreifer Zugriff auf nur wenige Daten zu ermöglichen.

Das Mitglied h, welches derjenige ist der den Anteil am längsten nicht aktualisiert hat, erneuert diesen ebenso wie bei Member Exclusion. Danach führt h den Broadcast aus.

M_h M_i

$$\{\alpha^{\frac{\widehat{N_h}\,\Pi(N_p \mid p\in[1,n])}{N_j}} \mid j \in [1,n]\} \longrightarrow$$

$$\text{Broadcast: round 1 } (N_h \leftarrow N_h\widehat{N_h})$$

Sicherheit der AKA-Operationen:

Bei der AKA-Sicherheitsbetrachtung müssen, erstens alle momentanen Teilnehmer zweitens alle ausgeschlossenen Mitglieder und alle zukünftigen Mitglieder in den Focus genommen werden. Die momentanen Teilnehmer werden aus der Betrachtung ausgenommen, da wir von einer erfolgreichen Identitätsprüfung ausgehen. Die anderen beiden Gruppen können zu einer Menge an möglichen Angreifern EVE zusammengefasst werden.

Man nehme an EVE hat die Möglichkeit alle Nachrichten zwischen den Teilnehmern abzufangen. Sie kann davon ausgehen, dass der Letzte benötigte Teilschlüssel nur in dem Broadcast einer Member-Addition oder Exclusion Operation enthalten ist (Dessen Anteil sich zu vorher geändert hat). So erlangt man wieder die letzte Zeile des unter IKA aufgeführten Sets. Daraus folgt, dass alle AKA-Operationen eine Instanz des GDDH-Problems sind.

Weitere Ansätze:

Der Ansatz von Steer auch als Str referenziert, basiert nicht auf GCDH oder GDDH ist aber auch sicher, jedoch aufgrund seiner Struktur ungeeignet für dynamische Gruppen.

Um eine andere Ausführung des GCDH handelt es sich bei dem Protokoll von Burmester und Desmedt. IKA wird effizienter ausgeführt bei AKA jedoch ist es aufgrund eines höheren Synchronisationsaufwand weniger effizient als CLIQUES.

Das am häufigsten angewendete Protokoll ist das TGDH (Tree-based generalized Diffie-Hellman), da es durch Ausnutzung einer Baumstruktur effizientere AKA Operationen ermöglicht, jedoch einen höheren Speicheraufwand aufweist.

Literaturverweise:

Michael Steiner, Gene Tsudik, Michael Waidner „Key Agreement in Dynamic Peer Groups“ 1999

Michael Steiner „Secure Group Key Agreement“ 2002

Prof. Dr. Schnorr „Vorlesung Kryptographie I und II“ 2001