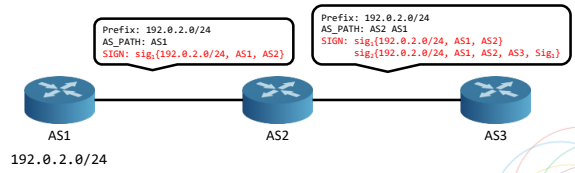
アグリゲート署名を用いたBGPsec AS_PATH検証手法の提案と実装評価

大阪大学 情報科学研究科
矢内 直人，OUYANG Junjie

BGPsec

- 電子署名を用いたBGPのセキュリティ拡張 [LS11]
  - 経路情報の生成元の正しさを検証するOrigin Validation機能
  - 経路情報が通過してきたパスを検証するPath Validation機能
- 経路情報の増大によってBGPルータのメモリ容量が不足し，既存のBGPルータへの適用が困難



アグリゲート署名

複数の署名を一つに集約(aggregate)可能な署名[BGLS03]（署名のサイズを$O(n)$から$O(1)$にできる）

- 利点：署名サイズの大幅な削減が可能
  - 各ルータから受信したAS_PATHへの署名を集約
  - IETF でも一部で議論
    (BGPsec Design Choices and Summary of Supporting Discussions)
- 欠点：署名の構成が不適切で，思いのほか役に立たない
  - 署名の構成がBGPsecの仕様に合っておらず，安全性が下がる(?)
  - 仕様に合わせた構成にすると，集約の性能が落ちる

[BGLS03] D. Boneh, C. Gentry, B. Lynn, H. Shacham. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. EUROCRYPT 2003.

アグリゲート署名の“お約束”

署名の集約可否は署名の作りに依存
→アルゴリズムが決まった時点で
「できる」署名か「できない」署名か決まる

- ECDSA … 「できない」署名の代表
- GDH署名[BLS01] … 「できる」署名の代表
  - 署名：$\sigma_i = x_i \cdot H(m_i)$
  - 検証：$e(\sigma_i, g_2) = e(H(m_i), x_i \cdot g_2)$

署名の検証には双線形写像$e: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$を利用：
$e(H(m_i), x_i \cdot g_2) = e(H(m_i), g_2)^{x_i} = e(x_i \cdot H(m_i), g_2)$

[BLS01] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. ASIACRYPT 2001.

アグリゲート署名概略

- GDH署名[BLS01]
  - 署名：$\sigma_i = x_i \cdot H(m_i)$
  - 検証：$e(\sigma_i, g_2) = e(H(m_i), x_i \cdot g_2)$
- アグリゲート署名[BGLS03]：GDH署名から構成
  - 集約：$\sigma = \sum_{i=1}^{n} \sigma_i$
  - 検証：$e(\sigma, g_2) = \prod_{i=1}^{n} e(H(m_i), x_i \cdot g_2)$

集約では足し算で集約，検証では掛け算により検証
（双線形写像を通じてn個分を一括計算）

アグリゲート署名の問題点

署名の構成が不適切で，思いのほか役に立たない

- BGPsecではAS_PATHだけではなく，「前のASが作った署名」にも署名する（署名のチェーン）
- アグリゲート署名でチェーンを作ろうとしたら…？
  - 署名と集約：$\sigma_i = x_i \cdot H(m_i, \sigma_{i-1}) + \sigma_{i-1}$
  - 検証：$e(\sigma, g_2) = \prod_{i=1}^{n} e(H(m_i, \sigma_{i-1}), x_i \cdot g_2)$

すべてのAS の署名が必要になり，集約の意味がない

この課題を解決するアグリゲート署名を作る

併存型アグリゲート署名

署名の「検証式」に対して署名することで，チェーンを作る

- GDH署名 [BLS01]
  - 署名：$\sigma_i = x_i \cdot H(m_i)$
  - 検証：$e(\sigma_i, g_2) = e(H(m_i), x_i \cdot g_2)$
- 併存型アグリゲート署名：GDH署名から構成
  - 署名と集約：$\sigma_i = x_i \cdot H(m_i, e(H(m_{i-1}), x_{i-1} \cdot g_2)) + \sigma_{i-1}$

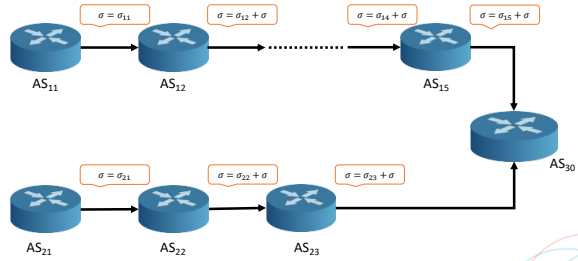
ひとつ前の署名の検証式をハッシュに入れる

  - 検証：$e(\sigma, g_2) = \prod_{i=1}^{n} e(H(m_i, e(H(m_{i-1}), x_{i-1} \cdot g_2)), x_i \cdot g_2)$

検証処理の中でも前のASの署名は不要

7

併存型アグリゲート署名 による署名集約 その一

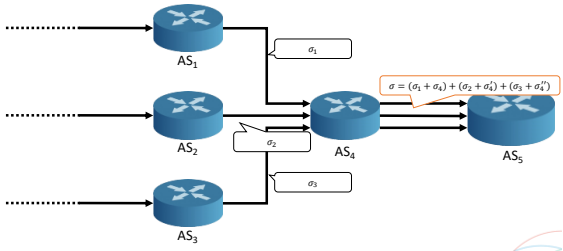


“AS数”相当の署名を受信
AS_{30}の署名受信数：8

“経路数”相当の署名を受信
AS_{30}の署名受信数：2

8

併存型アグリゲート署名 による署名集約 その二



“経路数”相当の署名を受信
AS_5の署名受信数：3

1個の署名を受信
AS_5の署名受信数：1

9

実験方針

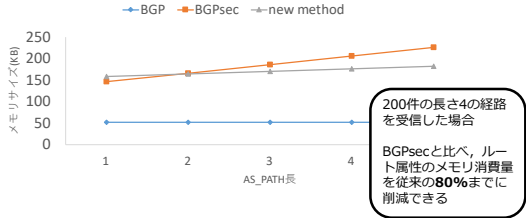
- シミュレーションツール
  - **BIRD Internet Routing Daemon**
  - 経路情報のメモリ消費量を測定
  - 仮想ネットワーク環境
    - 6台の仮想BGPルータにより線状ネットワークを構築
- アグリゲート署名の実装
  - 署名および検証の演算用ライブラリはTEPLA2.0を利用
  - BIRDのソースコードを編集し、BGPsecの仕様を変更

10

メモリ消費量の計測結果

単位：KB

AS_PATH長	1	2	3	4	5
ﾙｰﾃｨﾝｸﾞﾃｰﾌﾞﾙ	42	42	42	42	42
ﾙｰﾄ属性	116	122	128	134	140
合計	158	164	170	176	182



200件の長さ4の経路を受信した場合

BGPsecと比べ，ルート属性のメモリ消費量を従来の**80%**までに削減できる

まとめ

- BGPsecの標準化は進んでいるが，既存のBGPルータへの適用が難しくインターネット全体への導入が困難
- アグリゲート署名を用いたメモリ消費量の削減方法に注目し，新たな検証手法を提案
- BIRDで新手法をプロトタイプ実装し計測した結果，長さ4の経路を200件ずつ格納する場合のメモリ消費量は，従来のBGPsecと比べ，80%まで削減可能
- 計算量の増大により，検証時間が長くなる

12