2ZD-03

DAG で表現される順序構造をもつアグリゲート MAC

田邊 和宗
千葉大学 大学院融合理工学府
数学情報科学専攻

多田 充
千葉大学 大学院理学研究院
m.tada@faculty.chiba-u.jp

概要

共通鍵を用いた認証方式であるメッセージ認証コード（MAC）におけるメッセージ認証に用いられるタグについて，複数の送信者からそれぞれ異なるメッセージに対して生成したタグを 1 つに集約し，検証者がその集約タグに対し，1 回検証するだけで，送信者全員のタグの検証を可能にするアグリゲート MAC（AMAC）は，タグサイズや検証にかかる計算コストを軽減できるという利便性をもつ。本論文では，有向非巡回グラフ（DAG）で表現される順序構造を可能とする AMAC（DAM）を提案し，その安全性について述べる。

1 はじめに

インターネットにおける通信において, データの認証が重要となる。データ認証には秘密の共有鍵を用いたメッセージ認証コード (MAC) や, 公開鍵暗号系に基づくデジタル署名が挙げられる。計算リソースの限られたデバイス間の通信において MAC が用いられることが多いが, 認証タグを生成するノードが多くなるにつれて, 認証にかかる計算コストは増大する。[3] が提案したアグリゲート MAC は複数の MAC タグを一括で検証できるものであり, タグを集約してトータルのタグサイズを 1 つ分のタグサイズと同程度とした。しかし, [3] の方式では, タグ生成順序を検証できず, [1] によって安全性の問題が指摘された。[1] は, 逐次集約可能なアグリゲート MAC を提案し, その安全性を証明した。[2] では, 直並列グラフで表現できるタグ生成順序を可能としたアグリゲート MAC を提案し, 利用されている擬似ランダム関数の識別不可能性にその安全性を帰着させた。本論文では, [2] のタグ生成順序をさらに拡張し, 有向非巡回グラフ (DAG) で表現できるタグ生成順序を可能とした。

2 メッセージ認証コード (MAC)

メッセージ認証コード（MAC）を以下のように定義する。

定義 1（MAC） MAC 方式 MAC=(KG, Tag, Ver) は以下のアルゴリズムからなる。

鍵生成： $sk \leftarrow \mathsf{KG}(1^\lambda)$. セキュリティーパラメータ λ を入力とし, 秘密鍵 sk を出力する。

タグ生成： $\sigma \leftarrow \mathsf{Tag}(sk, m)$. 秘密鍵 sk とメッセージ $m \in \mathcal{M}$ を入力とし, タグ σ を出力する。

タグ検証： $b \leftarrow \mathsf{Ver}(sk, m, \sigma)$. 秘密鍵 sk, メッセージ $m \in \mathcal{M}$, タグ σ を入力とし, タグの正当性の判定 b ($\in \{0,1\}$) を出力する。

3 有向非巡回グラフ（DAG）

DAG は閉路を含まない有向グラフであり, 本論文では, タグを生成する順序を表現するために用いる。DAG の各頂点はタグの生成者を表し, 2 つの頂点 u, v を結ぶ有向辺 (u, v) は, タグ生成者 u がタグ生成者 v に情報を送ることを示す。

3.1 DAG の合成

2 つの DAG $\psi_1 = (V_1, E_1)$ および $\psi_2 = (V_2, E_2)$ に対して, それらの合成を $\psi_1 \sqcup \psi_2$ を表記し, $\psi_1 \sqcup \psi_2 \stackrel{\text{def}}{=} (V_1 \cup V_2, E_1 \cup E_2)$ と定義する。

2 つの DAG の合成は, 必ずしも DAG になるとは限らない。本論文では, DAG をタグ生成の順序を表現するために用いるため, 合成する DAG の間に矛盾がないか, また, 合成した結果が DAG になるかを検証する必要がある。後者については, 深さ優先探索などを適用することにより, 容易に検証できるが, 前者については, DAG に対する最大後方部分グラフ（GBP）, および, 複数の DAG に対する無矛盾性を定義する必要がある。

3.2 DAG に対する GBP と History, および複数の DAG に対する無矛盾性

DAG ψ における, 頂点 u に対する GBP は, u に到達できる全ての頂点, および, 到達するまでに辿ることが

できる全ての辺からなる ψ の部分グラフであり, $\psi_B^{(u)}$ と表記する。また, DAG ψ に対する History（$\mathsf{H}(\psi)$）は,

$$\mathsf{H}(\psi) \stackrel{\text{def}}{=} \left\{ \psi_B^{(u)} \;\middle|\; u \in V(\psi) \right\}$$

と定義し, 2 つの DAG ψ_1, ψ_2 が無矛盾であるとは, $\mathsf{H}(\psi_1 \sqcup \psi_2) = \mathsf{H}(\psi_1) \cup \mathsf{H}(\psi_2)$ であることとする。

4 提案方式

ここでは, DAG アグリゲート MAC 方式（DAM 方式）を提案する。この方式において, 集約に参加するタグ生成者の集合を $\mathcal{ID}(:= \{ID_1, \ldots, ID_n\})$ と表記する。

定義 2（DAM） DAM 方式 $\mathsf{DAM} = (\mathsf{DKG}, \mathsf{Tag}, \mathsf{DTag}, \mathsf{DVer})$ は以下のアルゴリズムからなる。

鍵生成（DKG）： セキュリティパラメータ λ を入力とし, 秘密鍵 sk を出力する。これを $sk \leftarrow \mathsf{DKG}(1^\lambda)$ と表記する。

シングルタグ生成（Tag）： ID とメッセージ m, ID に対応する秘密鍵 sk_{ID} を入力とし, タグ t を出力する。これを $t \leftarrow \mathsf{Tag}(sk_{ID}, ID, m)$ と表記する。

集約タグ生成（DTag）： $\mathcal{I} \subset \mathcal{ID}$, メッセージ m_i, 各 ID_i までの順序構造 ψ_i とタグ τ_i（ただし, i は $ID_i \in \mathcal{I}$ となるものとする）, および, 集約を行う者の ID, メッセージ m と秘密鍵 sk を入力とし, タグ生成順序 ψ および集約タグ τ を出力する。これを

$$(\tau, \psi) \leftarrow \mathsf{DTag}(ID^{(\mathcal{I})} \cup \{ID\}, sk, m^{(\mathcal{I})} \cup \{m\}, \tau^{(\mathcal{I})}, \psi^{(\mathcal{I})})$$

と表記する。ただし, $ID^{(\mathcal{I})} \stackrel{\text{def}}{=} \bigcup_{ID_i \in \mathcal{I}} \{ID_i\}$ とし, $sk^{(\mathcal{I})}$ などについても同様とする。

集約タグ検証（DVer）： $\mathcal{I} \subset \mathcal{ID}$, メッセージの集合 $m^{(\mathcal{I})}$, 秘密鍵の集合 $sk^{(\mathcal{I})}$, タグ生成の順序構造 $\psi_{\mathcal{I}}$ と集約タグ $\tau_{\mathcal{I}}$ を入力とし, 集約タグの正当性の判定 b（$\in \{0,1\}$）を出力する。これを,

$$b \leftarrow \mathsf{DVer}(ID^{(\mathcal{I})}, sk^{(\mathcal{I})}, m^{(\mathcal{I})}, \tau_{\mathcal{I}}, \psi_{\mathcal{I}})$$

と表記する。

4.1 擬似ランダム関数を用いた DAM の構成法

$f : \{0,1\}^L \times \{0,1\}^\lambda \to \{0,1\}^k$ を擬似ランダム関数（PRF）とする。ただし, L は十分大きいものとし，本来の入力サイズが L に満たない場合は，$00\cdots0$ などのビット列をパディングするものとする。また，ここでは, $\alpha \in \{0,1\}^\lambda$ に対して, $f(*, \alpha)$ を $f_\alpha(*)$ と表記する。

下記に, PRF を用いた DAM 方式における, 各タグ生成者 P が行う手続きを述べる。

P は, ID と秘密鍵 (ID, sk) を持ち, 他のタグ生成者 $P_1, \ldots, P_\ell$ から, ID とメッセージの集合 $(ID^{\psi_1}, m^{(\psi_i)})$, 集約タグ τ_{ψ_i}, 集約順序 ψ_i を受け取る。

P は受け取った順序構造 $\psi_1, \ldots, \psi_\ell$ に矛盾がないか確かめ, $\psi \stackrel{\text{def}}{=} \bigsqcup_{i=1}^{\ell} \psi_i \wedge ID$ を求める。そして,

$$\tau \stackrel{\text{def}}{=} f_{sk}(ID^{(\ell)} \cup \{ID\}, m^{(\psi)}, \bigoplus_{i=1}^{\ell} \tau_{\psi_i}, \psi)$$

を計算し, $ID, m^{(\psi)}, \tau, \psi$ を次のタグ生成者に送る。

P が最初のタグ生成者となる場合は, すべての ψ_i を空グラフ, $ID^{(\psi_i)}$ などは全て $\varnothing$ として上記を適用する。

4.2 PRF を用いた DAM の安全性

前節で構成した DAM 方式の安全性は, [2] の方式に対する安全性と同様に示すことができる。すなわち, 提案 DAM 方式において集約タグを偽造できる多項式時間アルゴリズム A が存在すると仮定すると, 使用している PRF $f_{sk_1}, \ldots, f_{sk_n}$ と, 真にランダムな関数 $\gamma_1, \ldots, \gamma_n$ を識別できる多項式時間アルゴリズムが存在することを示すことができる。

5 まとめ

本論文では擬似ランダム関数（PRF）を用いて, DAG で表せるタグ生成順序構造をもつアグリゲート MAC 方式を提案した。提案方式では, タグ生成者の順序構造を動的に構築することが可能であるため, より自由度の高い順序での集約タグの生成が期待できる。

参考文献

[1] O. Eikemeier, M. Fischlin, J. F. Götzmann, A. Lehmann, D. Schröder, P. Schröder and D. Wagner: "*History-free aggregate message authentication codes*", Proceedings of SCN2010, Lecture Notes in Computer Science, vol.6280, pp.309-328, Springer-Verlag, 2010.

[2] Y. Ishii and M. Tada: "*Structurally aggregate message authentication codes*", Proceedings of The International Symposium on Information Theory and Its Applications (ISITA) 2020, pp.339-343, 2020.

[3] J. Katz and A. Lindell: "*Aggregate message authentication codes*", Proceedings of CT-RSA 2008, Lecture Notes in Computer Science, vol.4964, pp.155-169, Springer-Verlag, 2008.