

(19)日本国特許庁(JP)

(12)公表特許公報(A)

(11)公表番号
特表2025-514754
(P2025-514754A)

(43)公表日 令和7年5月9日(2025.5.9)

(51)国際特許分類

F I

テーマコード (参考)

H O 4 N 21/236(2011.01) H O 4 N 21/236

H O 4 N 21/434(2011.01) H O 4 N 21/434

5 C 1 6 4

審査請求 未請求 予備審査請求 有 (全72頁)			
(21)出願番号	特願2024-561796(P2024-561796)	(71)出願人	507236292
(86)(22)出願日	令和5年4月13日(2023.4.13)		ドルビー ラボラトリーズ ライセンシン
(85)翻訳文提出日	令和6年10月18日(2024.10.18)		グ コーポレイション
(86)国際出願番号	PCT/US2023/018469		アメリカ合衆国 9 4 1 0 3 カリフォル
(87)国際公開番号	WO2023/205025		ニア州 サンフランシスコ マーケット
(87)国際公開日	令和5年10月26日(2023.10.26)		ストリート 1 2 7 5
(31)優先権主張番号	63/332,210	(74)代理人	110004381
(32)優先日	令和4年4月18日(2022.4.18)		弁理士法人I T O H
(33)優先権主張国・地域又は機関	米国(US)	(72)発明者	リードミラー, ジェフリー
(31)優先権主張番号	63/332,212		アメリカ合衆国 9 4 1 0 3 カリフォル
(32)優先日	令和4年4月18日(2022.4.18)		ニア州 サンフランシスコ マーケット
(33)優先権主張国・地域又は機関	米国(US)		ストリート 1 2 7 5 ドルビー ラボラ
(31)優先権主張番号	63/443,933	(72)発明者	トリーズ インコーポレイテッド 内
	最終頁に続く		サンチェス, フレディ
			アメリカ合衆国 9 4 1 0 3 カリフォル
			最終頁に続く

(54)【発明の名称】 コーディングされたメディアのためのマルチソース方法及びシステム

(57)【要約】

コーディングされたメディアのマルチソース方法及びシステムに関する実施形態が開示される。一部の実施形態において、第1のデバイスにおける方法は：メディア資産を表現するメディア・データを受信し；ビットストリーム識別データ、コンテンツ固有エンコード・データ、及びメディア・セグメント・データのうちの少なくとも1つを含む第1の複数のデータ要素を取得し；メディア・データの少なくとも一部分を、第1のコーディング・プロセスに従って符号化して、メディア資産に対応するコーディングされたデータを取得し；第1のコーディング・プロセスに関連する情報に基づいて、第1の複数のデータ要素とは異なる第2の複数のデータ要素を生成し；第1の複数のデータ要素と第2の複数のデータ要素とを、メディア資産を表現する1つ以上のコーディングされたビットストリームに結合し；1つ以上のコーディングされたビットストリームを1つ以上の第2のデバイスへ、1つ以上のネットワーク経路を用いて伝送するステップを含む。

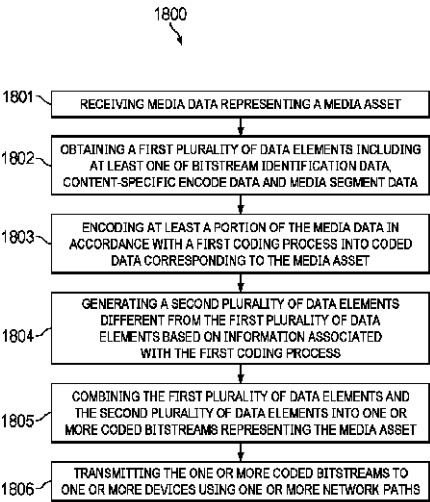


FIG. 18

【特許請求の範囲】**【請求項 1】**

第1のデバイスにおける方法であって：

少なくとも1つのプロセッサを使用して、メディア資産を表現するメディア・データを受信するステップ；

前記少なくとも1つのプロセッサを使用して、ビットストリーム識別データ、コンテンツ固有エンコード・データ、及びメディア・セグメント・データのうちの少なくとも1つを含む第1の複数のデータ要素を取得するステップ；

前記少なくとも1つのプロセッサを使用して、前記メディア・データの少なくとも一部分を、第1のコーディング・プロセスに従って符号化して、前記メディア資産に対応するコーディングされたデータを取得するステップ；

前記少なくとも1つのプロセッサを使用して、前記第1のコーディング・プロセスに関連する情報に基づいて、前記第1の複数のデータ要素とは異なる第2の複数のデータ要素を生成するステップ；

前記少なくとも1つのプロセッサを使用して、前記第1の複数のデータ要素と前記第2の複数のデータ要素とを、前記メディア資産を表現する1つ以上のコーディングされたビットストリームに結合するステップ；及び

前記少なくとも1つのプロセッサを使用して、前記1つ以上のコーディングされたビットストリームを1つ以上の第2のデバイスへ、1つ以上のネットワーク経路を用いて伝送するステップ；

を含む方法。

【請求項 2】

請求項1に記載の方法において、前記第1のコーディング・プロセスは、前記メディア資産に対応するコーディングされたデータを取得するための線型コーディング・プロセスである、方法。

【請求項 3】

請求項1に記載の方法において、前記第1のコーディング・プロセスは、非線型コーディング・プロセスであって、前記メディア・データを、別の線型コーディング・プロセスに合致するコーディングされたデータに変換するものである、方法。

【請求項 4】

請求項1に記載の方法において、前記第1の複数のデータ要素は、コンテンツ固有エンコード・データ又はメディア・セグメント・データのうちの少なくとも1つとビットストリーム識別データとを含む、方法。

【請求項 5】

請求項1に記載の方法において、前記データ要素はデータ構造である、方法。

【請求項 6】

請求項1ないし5のうちの何れか一項に記載の方法において、前記第1の複数のデータ要素は同期データ要素を含み、前記1つ以上のコーディングされたビットストリームの各々は同期データ要素で始まる、方法。

【請求項 7】

請求項1ないし6のうちの何れか一項に記載の方法において、前記同期データ要素は：

前記1つ以上のコーディングされたビットストリームのそれぞれのコーディングされたビットストリームが、共通のソース・メディア資産から生成されたことを識別する識別データ；

前記1つ以上のコーディングされたビットストリームのそれぞれのコーディングされたビットストリームを、前記共通のソース・メディア資産から生成された他のコーディングされたビットストリームと同期するための同期データ；及び

前記1つ以上のコーディングされたビットストリームのそれぞれのコーディングされたビットストリームの系譜データ；

のうちの少なくとも1つを含む、方法。

【請求項 8】

請求項7に記載の方法において、系譜データは：

前記それぞれのコーディングされたビットストリームに含まれるコーディングされたシンボルを生成するために使用された元の又はコーディングされたシンボルを識別するデータ；

前記それぞれのコーディングされたビットストリームに含まれるコーディングされたシンボルを生成するために、元の又はコーディングされたシンボルがどのように結合されたかを識別するデータ；及び

前記それぞれのコーディングされたビットストリームに含まれるコーディングされたシンボルを生成するために使用された符号化係数を識別するデータ；

を含む、方法。

10

【請求項 9】

請求項1ないし8のうちの何れか一項に記載の方法において、前記1つ以上のコーディングされたビットストリーム全体にわたって同期データ要素をインターリーブするステップを更に含み、前記同期データ要素は、前記1つ以上のコーディングされたビットストリームのうちの1つを受信する第2の電子デバイスが、前記1つ以上のコーディングされたビットストリームのうちの1つの中で、複数のポイントから復号化を開始できるようにする、方法。

【請求項 10】

請求項1ないし9のうちの何れか一項に記載の方法において、前記第1の複数のデータ要素又は前記第2の複数のデータ要素は、受信したメディア・データに関連するシステム・データに少なくとも部分的に基づいて生成される、方法。

20

【請求項 11】

請求項1ないし9のうちの何れか一項に記載の方法において、前記第1の複数のデータ要素を取得するステップは、受信したメディア・データ又は受信したメディア・データに関連するシステム・データから、前記第1の複数のデータ要素を生成するステップを含む、方法。

【請求項 12】

請求項1ないし11のうちの何れか一項に記載の方法において、前記メディア・データを第1のコーディング・プロセスに従って符号化することは、前記メディア・データを複数の元のシンボルに分け、前記複数の元のシンボルに基づいて複数のコーディングされたシンボルを生成することを含む、方法。

30

【請求項 13】

請求項12に記載の方法において、ネットワーク・コーディング技術に従って前記複数の元のシンボルのサブセットを線型結合することによって、コーディングされたシンボルが生成される、方法。

【請求項 14】

請求項1ないし13のうちの何れか一項に記載の方法において、前記メディア・データは線型コーディングされたビットストリームを含む、方法。

【請求項 15】

請求項1ないし14のうちの何れか一項に記載の方法において、第1の複数のデータ要素を取得するステップは：

前記メディア・データからメタデータを取り出すステップ；及び

取り出されたメタデータから前記第1の複数のデータ要素を演算するステップ；

を含む、方法。

40

【請求項 16】

請求項1ないし15のうちの何れか一項に記載の方法において、線型復号化を実行することなく、前記メディア・データからメタデータが取り出される、方法。

【請求項 17】

請求項1ないし16のうちの何れか一項に記載の方法において、前記メディア・データ

50

を第1のコーディング・プロセスに従って符号化することは、前記コーディングされたビットストリーム内の線型コーディングされたシンボルを、前記メディア資産を表現する新たな線型コーディングされたシンボルに再符号化することを含む、方法。

【請求項18】

請求項1ないし17のうちの何れか一項に記載の方法において、結合するステップの前に、第1のサイズを有する1つ以上のデータ要素を、前記第1のサイズより小さな第2のサイズを有するデータ要素に分けるステップを更に含む、方法。

【請求項19】

請求項1ないし18のうちの何れか一項に記載の方法において、ハッシュ・データを、前記第2の複数のデータ要素のうちの1つ以上のデータ要素に追加するステップを更に含み、これにより、前記1つ以上のビットストリームを受信するデバイスによる検証、認証、又は完全性のうちの少なくとも1つを可能にする、方法。

10

【請求項20】

請求項1ないし19のうちの何れか一項に記載の方法において、前記第2の複数のデータ要素の1つ以上のデータ要素であって、前記メディア資産に対応するファイル、前記メディア資産のブロック、前記メディア資産のセグメント、及び前記メディア資産のパケットのうちの1つ以上に関連付けられるものに、ハッシュ・データを追加するステップを更に含む方法。

【請求項21】

請求項1ないし20のうちの何れか一項に記載の方法において、結合するステップは、ディレクトリ・データ要素を、前記1つ以上のコーディングされたビットストリームに挿入するステップを含む方法。

20

【請求項22】

請求項21に記載の方法において、前記ディレクトリ・データ要素は、各々のビットストリーム内のサブアトムを識別する際に受信デバイスを支援するために使用される、方法。

【請求項23】

請求項1ないし22のうちの何れか一項に記載の方法において、前記1つ以上のコーディングされたビットストリームは、第1のタイプのデータ要素を含む第1のコーディングされたビットストリームと、前記第1のタイプのデータ要素を含まない第2のコーディングされたビットストリームとを含む、方法。

30

【請求項24】

請求項1ないし23のうちの何れか一項に記載の方法において、クライアントにおけるデコーダを初期化するために使用される前記データ要素又はサブアトムは、遅延に敏感なデータではない、方法。

【請求項25】

請求項1ないし24のうちの何れか一項に記載の方法において、ビットストリームは、`a_bitstream_header()`、`block_header()`、`encoder_content_info()`又は`media_segment_info()` `subatoms`のうちの少なくとも1つと、`sync()` `subatom`とを含む、方法。

40

【請求項26】

請求項1ないし25のうちの何れか一項に記載の方法において、前記データ要素又はサブアトムは、符号化されたデータ又は遅延に敏感なデータを搬送している、方法。

【請求項27】

請求項1ないし26のうちの何れか一項に記載の方法において、ビットストリームは、`packet()` `subatoms`と`sync()` `subatoms`を含むが、`bitstream_header()`、`block_header()`、`encoder_content_info()`又は`media_segment_info()` `subatoms`のうちの少なくとも1つを含んでいない、方法。

【請求項28】

50

請求項1ないし27のうちの何れか一項に記載の方法において、伝送するステップは：

第1のタイプの通信チャネルを用いて前記第1のコーディングされたビットストリームを伝送するステップ；及び

前記第1のタイプとは異なる第2のタイプの通信チャネルを用いて前記第2のコーディングされたビットストリームを伝送するステップ；

を含む、方法。

【請求項29】

請求項28に記載の方法において、前記第2のタイプの通信チャネルは、高遅延チャネル、低帯域幅チャネル、信頼性のあるチャネル、又は、第1のタイプの伝送プロトコルとは異なる第2のタイプの伝送プロトコルを使用するチャネルである、方法。

10

【請求項30】

第1のデバイスにおける方法であって：

少なくとも1つのプロセッサを使用して、メディア資産に関連する1つ以上のコーディングされたビットストリームを受信するステップ；

前記少なくとも1つのプロセッサを使用して、前記1つ以上のコーディングされたビットストリームをバッファに格納するステップ；

前記バッファに格納された前記1つ以上のコーディングされたビットストリームを処理するステップであって：

前記少なくとも1つのプロセッサを使用して、ビットストリーム識別データ、コンテンツ固有エンコード・データ、及びメディア・セグメント・データのうちの少なくとも1つを含む第1の複数のデータ要素を、前記1つ以上のコーディングされたビットストリームから抽出し；

20

前記少なくとも1つのプロセッサを使用して、前記1つ以上のコーディングされたビットストリームを生成するために使用されるそれぞれのコーディング・プロセスに関連する情報を含む第2の複数のデータ要素を、前記1つ以上のコーディングされたビットストリームから取り出すことを含むステップ；及び

前記少なくとも1つのプロセッサを使用して、前記第2の複数のデータ要素に基づいて前記メディア資産を表現する復号化されたデータを生成すること、及び、前記少なくとも1つのプロセッサを使用して、前記第2の複数のデータ要素に基づいて前記メディア資産を表現する新たにコーディングされたデータを生成すること、のうちの一方を実行するステップ；

30

を含む方法。

【請求項31】

請求項30に記載の方法において、復号化されたデータを生成することは、コーディングされたデータを生成するために使用された符号化タイプの決定に従って、複数のデコーダのうちのそれぞれのデコーダを使用して、コーディングされたデータを復号化することを含む、方法。

【請求項32】

請求項31に記載の方法において、前記符号化タイプは、取り出された第1の複数のデータ要素から決定される、方法。

40

【請求項33】

請求項31に記載の方法において、前記符号化タイプは、取り出された第2の複数のデータ要素から決定される、方法。

【請求項34】

請求項31に記載の方法において、前記符号化タイプは、取り出された第2の複数のデータ要素を使用することなく、前記第1の複数のデータ要素から決定される、方法。

【請求項35】

請求項31に記載の方法において、前記符号化タイプは、取り出された第1の複数のデータ要素を使用することなく、前記第2の複数のデータ要素から決定される、方法。

【請求項36】

50

請求項31ないし35のうちの何れか一項に記載の方法において、復号化されたデータ及び前記第1の複数のデータ要素から、前記メディア資産に対応する消費可能データを生成するステップを更に含む、方法。

【請求項37】

請求項33に記載の方法において、前記消費可能データは、前記第1の電子デバイスで実行するアプリケーション、又はアプリケーションによってサポートされるメディア・ファイル・タイプと互換性がある、方法。

【請求項38】

請求項30ないし37のうちの何れか一項に記載の方法において、復号化されたデータを生成することは：

10

線型コーディングを実行することなく、第1のタイプのコーディングされたデータを第2のタイプのコーディングされたデータに処理し；且つ

前記第2のタイプのコーディングされたデータを復号化することが可能なデコーダを使用して、前記第2のタイプのコーディングされたデータを復号化すること；

を含む、方法。

【請求項39】

請求項38に記載の方法において、前記コーディングされたデータを処理することは、係数を整合させるか、又は前記コーディングされたデータを共通の有限体に射影するコーディング統合装置によって実行される、方法。

【請求項40】

20

請求項30ないし39のうちの何れか一項に記載の方法において、新たにコーディングされたデータを生成することは、前記第2の複数のデータ要素に含まれるコーディングされたデータを、コーディングされたデータを生成ために使用された符号化タイプの決定に従って、複数の再エンコーダのうちのそれぞれの再エンコーダを使用して処理することを含む、方法。

【請求項41】

請求項40に記載の方法において、前記符号化タイプは、取り出された第1の複数のデータ要素から決定される、方法。

【請求項42】

請求項40に記載の方法において、前記符号化タイプは、取り出された第2の複数のデータ要素から決定される、方法。

30

【請求項43】

請求項40に記載の方法において、前記符号化タイプは、取り出された第2の複数のデータ要素を使用することなく、前記第1の複数のデータ要素から決定される、方法。

【請求項44】

請求項40に記載の方法において、前記符号化タイプは、取り出された第1の複数のデータ要素を使用することなく、前記第2の複数のデータ要素から決定される、方法。

【請求項45】

請求項40に記載の方法において、新たにコーディングされたデータを生成することは：線型復号化を実行することなく、第1のタイプのコーディングされたデータを、第2タイプのコーディングされたデータに処理すること；及び

40

第2タイプのコーディングされたデータを符号化することが可能な再エンコーダを用いて、第2タイプのコーディングされたデータを再符号化すること；

を含む、方法。

【請求項46】

請求項30ないし45のうちの何れか一項に記載の方法において、コーディングされたデータを処理することは、係数を整合させるか、又は前記コーディングされたデータを共通の有限体に射影するコーディング統合装置によって実行される、方法。

【請求項47】

請求項30ないし46のうちの何れか一項に記載の方法において、新たにコーディングさ

50

れたデータを格納するか、又は前記新たにコーディングされたデータから1つ以上のビットストリームを生成するステップ；

を更に含む方法。

【請求項48】

請求項30ないし47のうちの何れか一項に記載の方法において、前記バッファに格納された前記1つ以上の符号化ビットストリームを処理するステップの前に、第1サイズを各々が有する前記第1の複数のデータ要素又は前記第2の複数のデータ要素のうちの第1の数量のデータ要素を、前記第1のサイズより大きな第2サイズを有する、より少ない数量のデータ要素に結合するステップを更に含む、方法。

【請求項49】

請求項48に記載の方法において、結合するステップは、CHUNKED_SUBATOMサブアトム内のデータに基づいている、方法。

【請求項50】

請求項30ないし48のうちの何れか一項に記載の方法において：

復号化又は再符号化の前に、第2の複数のデータ要素から完全性データを取り出すステップ；

それぞれのデータ要素がそれぞれの完全性データに基づいて検証の成功を判定したことに応じて、復号化又は再符号化のうちの少なくとも一方を実行するステップ；及び

それぞれのデータ要素がそれぞれの完全性データに基づいて検証の不成功を判定したことに応じて、復号化又は再符号化のうちの少なくとも一方の実行を控えるステップ；

を更に含む方法。

【請求項51】

請求項50に記載の方法において、前記完全性データはハッシュ・データを含み、前記ハッシュ・データの検証は鍵情報に基づいている、方法。

【請求項52】

請求項51に記載の方法において、前記鍵情報は、公開鍵又は秘密鍵のうちの少なくとも1つを含む、方法。

【請求項53】

請求項51に記載の方法において、前記鍵情報は、認証サーバーから受信される、方法。

【請求項54】

請求項51ないし53のうちの何れか一項に記載の方法において：

復号化又は再符号化の前に、前記第2の複数のデータ要素がそれぞれ暗号化されたデータを含んでいる旨の判定に従って、それぞれ暗号化されたデータを復号化鍵を使用して復号化するステップ；

を更に含む方法。

【請求項55】

請求項51に記載の方法において、前記鍵情報又はハッシュ・データは、bitstream_encryption_key_idに関連している、方法。

【請求項56】

請求項30に記載の方法において、無許可のアクセス、消費、復号化、又は再符号化を防止するために、メディア資産に関連付けられた受信した1つ以上のコーディングされたビットストリームに対して、前記メディア資産の所有者又は作成者によって暗号化が適用される、方法。

【請求項57】

請求項30に記載の方法において、

第1の電子デバイスに関連付けられた識別情報を、前記第1の電子デバイスとは異なる第2の電子デバイスへ伝送する送信するステップ；及び

メディア資産に関連付けられた1つ以上のコーディングされたビットストリームへのアクセス、消費、復号化、又は再符号化を可能にする復号鍵を、前記第2の電子デバイスから受信するステップ；

を更に含む方法。

【請求項 58】

第1のデバイスにおける方法であって：

少なくとも1つのプロセッサを使用して、第1のタイプの第1のコーディングされたビットストリームを取得するステップ；

前記少なくとも1つのプロセッサを使用して、前記第1のタイプとは異なる第2のタイプの第2のコーディングされたビットストリームを生成するステップ；及び

前記少なくとも1つのプロセッサを使用して、前記第2のコーディングされたビットストリームを1つ以上の第2の電子デバイスへ、1つ以上のネットワーク経路を介して伝送するステップ；

を含む方法。

【請求項 59】

請求項58に記載の方法において、前記第1のタイプの第1のビットストリームは、第1のタイプの線型コーディング・プロトコル又は標準に従って生成される、方法。

【請求項 60】

請求項58又は59に記載の方法において、前記第1又は第2のビットストリームは、データ・セグメントに関連する系統的パケットと、前記データ・セグメントに関連する非系統的パケットとを含む、方法。

【請求項 61】

請求項58ないし60のうちの何れか一項に記載の方法において、前記第1又は第2のビットストリームは、前記データ・セグメントに関連する系統的パケットを含み、前記データ・セグメントに関連する非系統的パケットを含まない、方法。

【請求項 62】

請求項58ないし61のうちの何れか一項に記載の方法において、前記第1又は第2のビットストリームは、前記データ・セグメントに関連する非系統的パケットを含み、前記データ・セグメントに関連する系統的パケットを含まない、方法。

【請求項 63】

請求項59ないし62のうちの何れか一項に記載の方法において、前記第2のタイプの第2のビットストリームは、第2のタイプの線型コーディング・プロトコル又は標準に従って生成される、方法。

【請求項 64】

請求項59ないし63のうちの何れか一項に記載の方法において、前記1つ以上のネットワーク経路は、1つより多いネットワーク、1つより多いネットワーク・タイプ、1つより多いネットワーク伝送媒体、又は、異なるパフォーマンス特性を有する通信チャネルを含む、方法。

【請求項 65】

第1のデバイスにおける方法であって：

少なくとも1つのプロセッサを使用して、第1のタイプのコーディングを使用して符号化されたメディア資産に関連付けられた第1のビットストリームを受信するステップ；

前記第1のビットストリームが第2のタイプのコーディングに関連付けられたデコーダで復号化されるべきである旨の判定に従って、前記第1のビットストリームを、第2のタイプの線型コーディングの第2のビットストリームに処理するステップであって：

前記少なくとも1つのプロセッサを使用して、前記第1のビットストリームの一群のフィールドからデータを取り出し、そのデータを、前記第2のビットストリームの第2の対応する一群のフィールドにマッピングし；

前記少なくとも1つのプロセッサを使用して、前記メディア資産に関連するメタデータを、前記第1のビットストリームから取り出し；

前記少なくとも1つのプロセッサを使用して、前記第2のビットストリームを、前記メディア資産のコーディングされていない表現に、前記第2のタイプのコーディングに関連付けられたデコーダを用いて復号化することを含むステップ；

前記第1のビットストリームが第2のタイプのコーディングに関連付けられたデコーダで復号化されるべきではない旨の判定に従って、前記第1のビットストリームを処理するステップであって：

前記少なくとも1つのプロセッサを使用して、前記メディア資産に関連付けられたメタデータ、及び第1のタイプのコーディングに関連付けられたデコーダを使用して、前記メディア資産のコーディングされていない表現に、前記第1のビットストリームを復号化し；

前記少なくとも1つのプロセッサを使用して、前記メディア資産のコーディングされていない表現を、前記メディア資産に関連する取り出されたメタデータとともにアプリケーションに配分することを含むステップ；

を含む、方法。

【請求項66】

命令を記憶する非一時的なコンピュータ読み取り可能な記憶媒体であって、前記命令は、演算装置により実行されると、請求項1ないし65のうちの何れか一項に記載の方法を前記演算装置に実行させる、記憶媒体。

【請求項67】

少なくとも1つのプロセッサ；及び

命令を記憶するメモリ；

を含む演算装置であって、前記命令は、前記少なくとも1つのプロセッサにより実行されると、請求項1ないし65のうちの何れか一項に記載の方法を前記演算装置に実行させる、通信装置。

【発明の詳細な説明】

【技術分野】

【0001】

【0001】 関連出願の相互参照

本件出願は、2022年4月18日付で出願された米国仮特許出願第63/332,210号、2022年4月18日付で出願された米国仮特許出願第63/332,212号、及び2023年2月7日付けで出願された米国仮特許出願第63/443,933号の優先権の利益を主張しており、これら各々の全体は参照により援用される。

【0002】

【0002】 技術分野

本開示は、一般に、メディアを符号化及び復号化することに関連する。

【背景技術】

【0003】

【0003】 アプリケーション層コーディングは、1つ以上のソースから1つ以上のレシーバへのデータ伝送の全体的な品質に対して下位の層のプロトコル、ネットワーク及びチャネルがもたらす影響を軽減するために使用されることが可能なアプローチである。ネットワークを介したデータ転送レートは、ネットワークの特性（例えば、輻輳、パケット損失、遅延など）に大きく依存する。アプリケーション層コーディングは、ネットワークの特性（例えば、ネットワーク容量、遅延など）、特定のユース・ケース（例えば、オンライン・ストリーミング）及び全体的な目的（例えば、レイテンシの最小化）に関する知識を与えられている際に、適切なコーディング・タイプを選択することによって、データ転送を最適化しようとする。

【発明の概要】

【0004】

【0004】 コーディングされたメディアのためのマルチソース方法及びシステムの実施形態が開示される。

【0005】

【0005】 一部の実施形態において、方法は、第1のデバイスにおいて：メディア資産（media asset）を表現するメディア・データを受信し；ビットストリーム識別データ

10

20

30

40

50

、コンテンツ固有符号化データ、及びメディア・セグメント・データのうちの少なくとも1つを含む第1の複数のデータ要素を取得し；メディア・データの少なくとも一部分を、第1のコーディング・プロセスに従って符号化して、メディア資産に対応するコーディングされたデータを取得し；第1のコーディング・プロセスに関連する情報に基づいて、第1の複数のデータ要素とは異なる第2の複数のデータ要素を生成し；第1の複数のデータ要素と第2の複数のデータ要素とを、メディア資産を表現する1つ以上のコーディングされたビットストリームに結合し；1つ以上のコーディングされたビットストリームを1つ以上の第2のデバイスへ、1つ以上のネットワーク経路を用いて伝送するステップを含む。

【0006】

【0006】 一部の実施形態において、第1のコーディング・プロセスは、メディア資産に対応するコーディングされたデータを取得するための線型コーディング・プロセス (linear coding process) である、

10

【0007】 一部の実施形態において、第1のコーディング・プロセスは、非線型コーディング・プロセス (non-linear coding process) であって、メディア・データを、別の線型コーディング・プロセスに合致するコーディングされたデータに変換するものである。

【0007】

【0008】 一部の実施形態において、データ要素はデータ構造である。

【0008】

【0009】 一部の実施形態において、第1の複数のデータ要素は、同期データ要素を含み、1つ以上のコーディングされたビットストリームの各々は同期データ要素で始まる。

20

【0009】

【0010】 一部の実施形態において、同期データ要素は：

1つ以上のコーディングされたビットストリームのそれぞれのコーディングされたビットストリームが、共通のソース・メディア資産から生成されたことを識別する識別データ；

1つ以上のコーディングされたビットストリームのそれぞれのコーディングされたビットストリームを、共通のソース・メディア資産から生成された他のコーディングされたビットストリームと同期するための同期データ；及び

1つ以上のコーディングされたビットストリームのそれぞれのコーディングされたビットストリームの系譜データ (lineage data) ；

30

のうちの少なくとも1つを含む。

【0010】

【0011】 一部の実施形態において、系譜データは：

それぞれのコーディングされたビットストリームに含まれるコーディングされたシンボルを生成するために使用された元の又はコーディングされたシンボル (original or coded symbols) を識別するデータ；

それぞれのコーディングされたビットストリームに含まれるコーディングされたシンボルを生成するために、元の又はコーディングされたシンボルがどのように結合されたかを識別するデータ；及び

40

それぞれのコーディングされたビットストリームに含まれるコーディングされたシンボルを生成するために使用された符号化係数を識別するデータを含む。

【0011】

【0012】 一部の実施形態において、方法は、1つ以上のコーディングされたビットストリーム全体にわたって同期データ要素をインターリーブするステップを更に含み、同期データ要素は、1つ以上のコーディングされたビットストリームのうちの1つを受信する第2の電子デバイスが、1つ以上のコーディングされたビットストリームのうちの1つの中で、複数のポイントから復号化を開始できるようにする。

【0012】

【0013】 一部の実施形態において、第1の複数のデータ要素又は第2の複数のデータ要

50

素は、受信したメディア・データに関連するシステム・データに少なくとも部分的に基づいて生成される。

【0013】

【0014】 一部の実施形態において方法は次の事項を含む：第1の複数のデータ要素を取得するステップは、受信したメディア・データ又は受信したメディア・データに関連するシステム・データから、第1の複数のデータ要素を生成するステップを含む。

【0014】

【0015】 一部の実施形態において、メディア・データを第1のコーディング・プロセスに従って符号化することは、メディア・データを複数の元のシンボルに分け、複数の元のシンボルに基づいて複数のコーディングされたシンボルを生成することを含む。

【0015】

【0016】 一部の実施形態において、ネットワーク・コーディング技術に従って複数の元のシンボルのサブセットを線型に結合することによって、コーディングされたシンボルが生成される。

【0016】

【0017】 一部の実施形態において、メディア・データは線型コーディングされたビットストリームを含む。

【0017】

【0018】 一部の実施形態において、第1の複数のデータ要素を取得するステップは：メディア・データからメタデータを取り出すステップ；及び

取り出されたメタデータから第1の複数のデータ要素を演算するステップを含む、

【0019】 一部の実施形態において、線型復号化を実行することなく、メディア・データからメタデータが取り出される。

【0018】

【0020】 一部の実施形態において、メディア・データを第1のコーディング・プロセスに従って符号化することは、コーディングされたビットストリーム内の線型コーディングされたシンボルを、メディア資産を表現する新たな線型コーディングされたシンボルに再符号化することを含む。

【0019】

【0021】 一部の実施形態において、方法は、結合するステップの前に、第1のサイズを有する1つ以上のデータ要素を、第1のサイズより小さな第2のサイズを有するデータ要素に分けるステップを更に含む。

【0020】

【0022】 一部の実施形態において、方法は、ハッシュ・データを、第2の複数のデータ要素のうちの1つ以上のデータ要素に追加するステップを更に含み、これにより、1つ以上のビットストリームを受信するデバイスによる検証、認証、又は整合性 (validation, authentication, or integrity) のうちの少なくとも1つを可能にする。

【0021】

【0023】 一部の実施形態において、方法は、第2の複数のデータ要素の1つ以上のデータ要素であって、メディア資産に対応するファイル、メディア資産のブロック、メディア資産のセグメント、及びメディア資産のパケットのうちの1つ以上に関連付けられるものに、ハッシュ・データを追加するステップを更に含む。

【0022】

【0024】 一部の実施形態において、結合するステップは、ディレクトリ・データ要素を、1つ以上のコーディングされたビットストリームに挿入するステップを含む。

【0023】

【0025】 一部の実施形態において、ディレクトリ・データ要素は、各々のビットストリーム内のサブアトム (subatoms) を識別する際に受信デバイスを支援するために使用される。

10

20

30

40

50

【0024】

【0026】 一部の実施形態において、1つ以上のコーディングされたビットストリームは、第1のタイプのデータ要素を含む第1のコーディングされたビットストリームと、第1のタイプのデータ要素を含まない第2のコーディングされたビットストリームとを含む。

【0025】

【0027】 一部の実施形態において、クライアントにおけるデコーダを初期化するために使用されるデータ要素又はサブアトムは、遅延に敏感なデータ (latency sensitive data) ではない、

【0028】 一部の実施形態において、ビットストリームは、a bitstream_header(), block_header(), encoder_content_info() 又は media_segment_info() subatoms のうちの少なくとも1つと、sync() subatom とを含む。

10

【0026】

【0029】 一部の実施形態において、データ要素又はサブアトムは、符号化されたデータ又は遅延に敏感なデータを搬送している。

【0027】

【0030】 一部の実施形態において、ビットストリームは、packet() subatoms と sync() subatoms を含むが、 bitstream_header(), block_header(), encoder_content_info() 又は media_segment_info() subatoms のうちの少なくとも1つを含んでいない。

20

【0028】

【0031】 一部の実施形態において、伝送するステップは：

第1のタイプの通信チャネルを用いて第1のコーディングされたビットストリームを伝送するステップ；及び

第1のタイプとは異なる第2のタイプの通信チャネルを用いて第2のコーディングされたビットストリームを伝送するステップを含む。

【0029】

【0032】 一部の実施形態において、第2のタイプの通信チャネルは、高遅延チャネル、低帯域幅チャネル、信頼性のあるチャネル、又は、第1のタイプの伝送プロトコルとは異なる第2のタイプの伝送プロトコルを使用するチャネルである。

30

【0030】

【0033】 一部の実施形態において、方法は、第1のデバイスにおいて：

メディア資産に関連する1つ以上のコーディングされたビットストリームを受信するステップ；

1つ以上のコーディングされたビットストリームをバッファに格納するステップ；

バッファに格納された1つ以上のコーディングされたビットストリームを処理するステップであって：ビットストリーム識別データ、コンテンツ固有符号化データ (content-specific encode data)、及びメディア・セグメント・データのうちの少なくとも1つを含む第1の複数のデータ要素を、1つ以上のコーディングされたビットストリームから抽出し；1つ以上のコーディングされたビットストリームを生成するために使用されるそれぞれのコーディング・プロセスに関連する情報を含む第2の複数のデータ要素を、1つ以上のコーディングされたビットストリームから取り出すことを含むステップ；及び

40

第2の複数のデータ要素に基づいてメディア資産を表現する復号化されたデータを生成すること、及び、第2の複数のデータ要素に基づいてメディア資産を表現する新たにコーディングされたデータを生成すること、のうちの一方を実行するステップを含む。

【0031】

【0034】 一部の実施形態において、復号化されたデータを生成することは、コーディングされたデータを生成するために使用された符号化タイプの決定に従って、複数のデコーダのうちのそれぞれのデコーダを使用して、コーディングされたデータを復号化することを含む。

50

【0032】

【0035】 一部の実施形態において、符号化タイプは、取り出された第1の複数のデータ要素から決定される。

【0033】

【0036】 一部の実施形態において、符号化タイプは、取り出された第2の複数のデータ要素から決定される。

【0034】

【0037】 一部の実施形態において、符号化タイプは、取り出された第2の複数のデータ要素を使用することなく、第1の複数のデータ要素から決定される。

【0035】

【0038】 一部の実施形態において、符号化タイプは、取り出された第1の複数のデータ要素を使用することなく、第2の複数のデータ要素から決定される。

【0036】

【0039】 一部の実施形態において、方法は、復号化されたデータ及び第1の複数のデータ要素から、メディア資産に対応する消費可能データ (consumable data) を生成するステップを更に含む。

【0037】

【0040】 一部の実施形態において、消費可能データは、第1の電子デバイスで実行するアプリケーション、又はアプリケーションによってサポートされるメディア・ファイル・タイプと互換性がある。

【0038】

【0041】 一部の実施形態において、復号化されたデータを生成することは：

線型コーディングを実行することなく、第1のタイプのコーディングされたデータを第2のタイプのコーディングされたデータに処理し；且つ

第2のタイプのコーディングされたデータを復号化することが可能なデコーダを使用して、第2のタイプのコーディングされたデータを復号化することを含む。

【0039】

【0042】 一部の実施形態において、コーディングされたデータを処理することは、係数を整合させるか、又はコーディングされたデータを共通の有限体 (common finite field) に射影するコーディング統合装置によって実行される。

【0040】

【0043】 一部の実施形態において、新たにコーディングされたデータを生成することは、第2の複数のデータ要素に含まれるコーディングされたデータを、コーディングされたデータを生成ために使用された符号化タイプの決定に従って、複数の再エンコーダのうちのそれぞれの再エンコーダを使用して処理することを含む。

【0041】

【0044】 一部の実施形態において、符号化タイプは、取り出された第1の複数のデータ要素から決定される。

【0042】

【0045】 一部の実施形態において、符号化タイプは、取り出された第2の複数のデータ要素から決定される。

【0043】

【0046】 一部の実施形態において、符号化タイプは、取り出された第2の複数のデータ要素を使用することなく、第1の複数のデータ要素から決定される。

【0044】

【0047】 一部の実施形態において、符号化タイプは、取り出された第1の複数のデータ要素を使用することなく、第2の複数のデータ要素から決定される。

【0045】

【0048】 一部の実施形態において、新たにコーディングされたデータを生成することは：

10

20

30

40

50

線型復号化を実行することなく、第1のタイプのコーディングされたデータを、第2タイプのコーディングされたデータに処理すること；及び

第2タイプのコーディングされたデータを符号化することが可能な再エンコーダを用いて、第2タイプのコーディングされたデータを再符号化することを含む。

【0046】

【0049】 一部の実施形態において、コーディングされたデータを処理することは、係数を整合させるか、又はコーディングされたデータを共通の有限体に射影するコーディング統合装置によって実行される。

【0047】

【0050】 一部の実施形態において、方法は、新たにコーディングされたデータを格納するか、又は新たにコーディングされたデータから1つ以上のビットストリームを生成するステップを更に含む。

10

【0048】

【0051】 一部の実施形態において、方法は、バッファに格納された1つ以上の符号化ビットストリームを処理するステップの前に、第1サイズを各々が有する第1の複数のデータ要素又は第2の複数のデータ要素のうちの第1の数量のデータ要素を、第1のサイズより大きな第2サイズを有する、より少ない数量のデータ要素に結合するステップを更に含む。

【0049】

【0052】 一部の実施形態において、結合するステップは、CHUNKED_SUBATOMサブアトム内のデータに基づいている。

20

【0050】

【0053】 一部の実施形態において、方法は：

復号化又は再符号化の前に、第2の複数のデータ要素から整合性データ (integrity data) を取り出すステップ；

それぞれのデータ要素がそれぞれの整合性データに基づいて検証の成功を判定したことに応じて、復号化又は再符号化のうちの少なくとも一方を実行するステップ；及び

それぞれのデータ要素がそれぞれの整合性データに基づいて検証の不成功を判定したことに応じて、復号化又は再符号化のうちの少なくとも一方の実行を控えるステップを更に含む。

30

【0051】

【0054】 一部の実施形態において、整合性データはハッシュ・データを含み、ハッシュ・データの検証は鍵情報に基づいている。

【0052】

一部の実施形態において、鍵情報は、公開鍵又は秘密鍵のうちの少なくとも1つを含む。

【0053】

【0055】 一部の実施形態において、鍵情報は、認証サーバーから受信される。

【0054】

【0056】 一部の実施形態において、方法は、復号化又は再符号化の前に、第2の複数のデータ要素がそれぞれ暗号化されたデータを含んでいる旨の判定に従って、それぞれ暗号化されたデータを復号化鍵を使用して復号化するステップを更に含む。

40

【0055】

【0057】 一部の実施形態において、鍵情報又はハッシュ・データは、bitstream_encryption_key_idに関連している。

【0056】

【0058】 一部の実施形態において、無許可のアクセス (unauthorized access) / 消費 / 復号化 / 再符号化を防止するために、メディア資産に関連付けられた受信した1つ以上のコーディングされたビットストリームに対して、メディア資産の所有者又は作成者によって暗号化が適用される。

50

【0057】

【0059】 一部の実施形態において、方法は：

第1の電子デバイスに関連付けられた識別情報を、第1の電子デバイスとは異なる第2の電子デバイスへ伝送する送信するステップ；及び

メディア資産に関連付けられた1つ以上のコーディングされたビットストリームへのアクセス、消費、復号化、又は再符号化を可能にする復号鍵を、第2の電子デバイスから受信するステップを更に含む。

【0058】

【0060】 一部の実施形態において、方法は、電子デバイスにおいて：

少なくとも1つのプロセッサを使用して、第1のタイプの第1のコーディングされたビットストリームを取得するステップ；

少なくとも1つのプロセッサを使用して、第1のタイプとは異なる第2のタイプの第2のコーディングされたビットストリームを生成するステップ；及び

少なくとも1つのプロセッサを使用して、第2のコーディングされたビットストリームを1つ以上の第2の電子デバイスへ、1つ以上のネットワーク経路を介して伝送するステップを含む。

【0059】

【0061】 一部の実施形態において、第1のタイプの第1のビットストリームは、第1のタイプの線型コーディング・プロトコル又は標準に従って生成される。

【0060】

【0062】 一部の実施形態において、第1又は第2のビットストリームは、データ・セグメントに関連する系統的パケット (systematic packets) と、データ・セグメントに関連する非系統的パケット (non-systematic packets) とを含む。

【0061】

【0063】 一部の実施形態において、第1又は第2のビットストリームは、データ・セグメントに関連する系統的パケットを含み、データ・セグメントに関連する非系統的パケットを含まない。

【0062】

【0064】 一部の実施形態において、第1又は第2のビットストリームは、データ・セグメントに関連する非系統的パケットを含み、データ・セグメントに関連する系統的パケットを含まない。

【0063】

【0065】 一部の実施形態において、第2のタイプの第2のビットストリームは、第2のタイプの線型コーディング・プロトコル又は標準に従って生成される。

【0064】

【0066】 一部の実施形態において、1つ以上のネットワーク経路は、1つより多いネットワーク、1つより多いネットワーク・タイプ、1つより多いネットワーク伝送媒体、又は、異なるパフォーマンス特性を有する通信チャネルを含む。

【0065】

【0067】 一部の実施形態において、方法は、第1のデバイスにおいて：

第1のタイプのコーディングを使用して符号化されたメディア資産に関連付けられた第1のビットストリームを受信するステップ；

第1のビットストリームが第2のタイプのコーディングに関連付けられたデコーダで復号化されるべきである旨の判定に従って、第1のビットストリームを、第2のタイプの線型コーディングの第2のビットストリームを生成するステップであって：第1のビットストリームの一群のフィールドからデータを取り出し、そのデータを、第2のビットストリームの第2の対応する一群のフィールドにマッピングし；メディア資産に関連するメタデータを、第1のビットストリームから抽出し；第2のビットストリームを、メディア資産のコーディングされていない表現に、第2のタイプのコーディングに関連付けられたデコ

10

20

30

40

50

ーダを用いて復号化することを含むステップ；

第1のビットストリームが第2のタイプのコーディングに関連付けられたデコーダで復号化されるべきではない旨の判定に従って、第1のビットストリームを処理するステップであって：第1のビットストリームを、メディア資産に関連付けられたメタデータ、及び第1のタイプのコーディングに関連付けられたデコーダを使用して、メディア資産のコーディングされていない表現に復号化し；メディア資産のコーディングされていない表現を、メディア資産に関連する取り出されたメタデータとともにアプリケーションに配分することを含むステップを含む。

【0066】

【0068】 一部の実施形態において、非一時的なコンピュータ読み取り可能な記憶媒体は命令を含み、命令は、演算装置により実行されると、前述の何れかの方法を演算装置に実行させる。

【0067】

【0069】 一部の実施形態において、演算装置は、少なくとも1つのプロセッサと命令を記憶するメモリとを含み、命令は、少なくとも1つのプロセッサにより実行されると、前述の何れかの方法を演算装置に実行させる。

【0068】

【0070】 本件で開示される他の実施形態は、システム、装置、及びコンピュータ読み取り可能な媒体を対象としている。開示される実施形態の詳細は、添付の図面及び以下の説明で述べられる。その他の特徴、目的、及び利点は、本説明、図面及びクレームから明らかである。

【0069】

【0071】 本件で開示される特定の実施形態は、以下の1つ以上の利点をもたらす。コーディング・プロセス（例えば、線型コーディング・プロセス）に関連するデータ要素を、送信前に、ビットストリーム識別データ、コンテンツ固有符号化データ、及びメディア・セグメント・データのうちの少なくとも1つを含むデータ要素と結合することは、様々なパフォーマンス特性を有するネットワークを介して、様々な処理能力（例えば、符号化／復号化能力）を有するデバイスへ／からのメディア・データのより効率的な配信を可能にする。結合されたデータは、より効率的な復号化をもたらし、互換性を確保するために複数の冗長タイプのビットストリームを生成及び伝送する必要性を低減し、それによってネットワーク及び処理リソースを節約する。

【0070】

【0072】 開示されるマルチソース方法及びシステム（multisource methods and systems）は、共通の表現及びインターフェースを有する異なるアプリケーションをサポートする、ユニークなビットストリーム・コンテナ・フォーマット（以下「開示されるビットストリーム」とも言及される）を使用するアプリケーション層コーディングをサポートする。開示されるビットストリームは、1つ以上の線型コードの使用をサポートするスケーラブルかつ相互運用可能な方法を可能にし、マルチメディアのクラウド及びネットワーク接続処理、ストレージ、配信、配信及び再生についてのネットワーク側の効率及び信頼性を向上させる。開示されるビットストリームはまた、異なるコーディング・タイプの標準化されたエンコーダ及びデコーダの実装に準拠するために、互換性のあるフォーマットでメタデータ及びその他の情報（例えば、コーディングされたシンボル、ブロック・カウント、ブロック当たりのシンボル数、コーディング行列に関する情報、コーディング行列係数のフィールド・サイズ）を搬送するようにも設計される。

【0071】

【0073】 開示される方法及びシステムはインターチェンジ・ビットストリーム・フォーマットもサポートしており、これは、1つ以上のネットワーク及び/又はアクセス経路を介するシングル・ソース及びマルチソースのメディア配信を含むユース・ケース及び幾つものコード・タイプの新規な利用を可能にする。

【0072】

10

20

30

40

50

【0074】 開示される方法及びシステムはまた、単一の処理エンティティが複数のシグナリングされるコード・タイプをオン・ザ・フライ（on-the-fly）でシームレスに処理及び/又は復号化することが可能な実施形態もサポートしている。

【図面の簡単な説明】

【0073】

【図1A】【0075】 図1A及び図1Bはそれぞれ線型符号化及び復号化の概念を示す。

【図1B】【0075】 図1A及び図1Bはそれぞれ線型符号化及び復号化の概念を示す。

【図2A】【0076】 図2A及び図2Bはそれぞれ一部の実施形態によるビットストリーム符号化及び復号化のブロック図である。

【図2B】【0076】 図2A及び図2Bはそれぞれ一部の実施形態によるビットストリーム符号化及び復号化のブロック図である。

10

【図3A】【0077】 図3Aは、一部の実施形態による開示されるビットストリームのデータ要素の階層的組織構造を示す。

【図3B】【0078】 図3Bは、一部の実施形態による開示されるビットストリームのサブアトムにおけるデータ要素の例を示す。

【図4】【0079】 図4は、一部の実施形態によるメディア処理及び配信システムのブロック図である。

【図5】【0080】 図5は、一部の実施形態による図4に示されるビットストリーム生成器のブロック図である。

【図6】【0081】 図6は、一部の実施形態による図4に示されるビットストリーム・デコーダのブロック図である。

20

【図7】【0082】 図7は、一部の実施形態によるビットストリーム再エンコーダのブロック図である。

【図8A】【0083】 図8Aは、一部の実施形態による統合デコーダのブロック図である。

【図8B】【0084】 図8Bは、一部の実施形態によるデコーダ・スイッチを有する統合デコーダのブロック図である

【図9A】【0085】 図9Aは、一部の実施形態による統合ビットストリーム再エンコーダのブロック図である。

【図9B】【0086】 図9Bは、一部の実施形態による再エンコーダ・スイッチを有する統合ビットストリーム再エンコーダのブロック図である。

30

【図10】【0087】 図10は、一部の実施形態による他のコーディング・ビットストリーム標準に準拠するためのビットストリーム・マッピングを示す。

【図11】【0088】 図11は、一部の実施形態によるマルチソース配信のビットストリーム例である。

【図12】【0089】 図12は、一部の実施形態によるマルチソース、マルチ・パス配信システムを示す。

【図13】【0090】 図13は、一部の実施形態による低レイテンシ・ユニキャスト・ビットストリーム配信システムを示す。

【図14A】【0091】 図14Aは、一部の実施形態による低レイテンシ・ユニキャスト符号化及び復号化のためのシステムのブロック図である。

40

【図14B】【0092】 図14Bは、一部の実施形態による可変データ・サイズのオフバンド処理を示す。

【図15A】【0093】 図15Aは、一部の実施形態による別個の線型エンコーダ・デコーダ対を有する低レイテンシ・ユニキャスト符号化を示す。

【図15B】【0094】 図15Bは、一部の実施形態による、世代間デコーダによる低レイテンシのユニキャスト符号化を示す。

【図16】【0095】 図16は、一部の実施形態によるデジタル著作権管理（DRM）のための開示されるビットストリームの使用例を示す。

【図17】【0096】 図17は、一部の実施形態による開示されるビットストリームによって可能になるデータ認証をとまなう分散型メディア配信システムのブロック図である。

50

【図18】[0097] 図18は、一部の実施形態による非線型又は線型コーディングされたデータからビットストリームを生成するプロセスのフロー図である。

【図19】[0098] 図19は、一部の実施形態による復号化/再符号化のプロセスのフロー・チャートである。

【図20】[0099] 図20は、一部の実施形態によるダウンストリーム互換性のために新しいコード・タイプを変換するプロセスのフロー・チャートである。

【図21】[0100] 図21は、一部の実施形態によるダウンストリーム互換性のために新しいコードタイプを変換する別のプロセスのフロー・チャートである。

【図22】[0101] 図22は、図1ないし図21を参照して説明されるシステム及び方法を実施するのに適した例示的なハードウェア・アーキテクチャのブロック図である。 [0102] 図面において、デバイス、ユニット、命令ブロック、及びデータ要素を表すもののような概略的な要素の具体的な配置又は順序が、説明を容易にするために示されている。しかしながら、図面における概略的な要素の具体的な配置又は順序は、処理の特定の順序若しくはシーケンス、又は複数プロセスの分離が必須であることを意味するようには意図されていない、ということは当業者によって理解されるはずである。更に、図面に概略的な要素が含まれていることは、そのような要素が全ての実施形態で必須とされていること、又は、そのような要素によって表現される特徴が、一部の実装では含まれなかったり又は他の要素と組み合わせられなかったりする可能性があること、を意味するようには意図されていない。

【0074】

[0103] 更に、図面において、実線又は破線又は矢印のような接続する要素は、2つ以上の他の概略的な要素の間又は中での接続、関係又は関連を示すために使用されている場合、そのような何らかの接続する要素が無いことは、接続、関係又は関連が存在し得ないことを意味するようには意図されていない。換言すれば、要素間の幾つかの接続、関係、又は関連は、本開示を不明瞭にしないように図面には示されていない。更に、説明を容易にするために、単一の接続要素を使用して、要素間の複数の接続、関係、又は関連を表している。例えば、接続する要素が信号、データ、又は命令の通信を表している場合、そのような要素は、必要に応じて、通信に影響を及ぼす1つ以上の信号経路を表している、ということは当業者によって理解されるはずである。

【0075】

[0104] 様々な図面で使用される同じ参照記号は、同様な要素を示す。

【発明を実施するための形態】

【0076】

[0105] 以下の詳細な説明では、説明される様々な実施形態の完全な理解をもたらすために、多数の具体的な詳細が述べられる。説明される様々な実施形態がこれらの具体的な詳細なしに実施されてもよい、ということは当業者には明白であろう。他の例において、周知の方法、手順、構成要素、及び回路は、実施形態の態様を不必要に曖昧にしないように詳細には説明されていない。以下で説明されている幾つの特徴は、それぞれ互いに独立して、又は他の特徴との何らかの組み合わせにより使用することが可能である。

【0077】

[0106] 用語

本件で使用される場合、用語「～を含む」及びその変形は、「～を含むがこれに限定されない」を意味するオープン・エンドの用語として読まれるべきである。用語「又は」は、文脈が明確に別意を示していない限り、「及び/又は」として読まれるべきである。用語「～に基づいて」は、「～に少なくとも部分的に基づいて」として読まれるべきである。用語「1つの実装例」及び「実施例」は、「少なくとも1つの実装例」として読まれるべきである。用語「別の実装」は、「少なくとも1つの他の実装」として読まれるべきである。用語「決定された」、「決定する」、又は「判定する」は、取得する、受信する、演算する、計算する、推定する、予測する、又は導出するとして読まれるべきである。更

に、以下の説明及びクレームにおいて、別意に定義されていない限り、本件で使用する全ての技術的及び科学的な用語は、本開示が属する分野における当業者によって一般的に理解されているものと同じ意味を有する。

【0078】

【0107】 上記の定義に加えて、以下の追加の定義が、以下の説明に適用されるものとする。

【0079】

【0108】

「符号化する (encoding)」は、ソース・コーディングされた/非線型コーディングされたメディア・データ入力を符号化すること、又は線型コーディングを使用して線型コーディングされたメディア入力を再コーディング (re-coding) することを指す。

【0080】

【0109】 「有限体 (finite field)」は、有限次数を有する体であり、体のうちの個数は算術的に閉じており完全 (arithmetically closed and complete) である。

【0081】

【0110】 「準同型ハッシュ (homomorphic hash)」は、暗号ハッシュ・アルゴリズムの族 (family) であり、複合データ・ブロック (例えば、個々のデータ・ブロックを線型に結合又は連結することによるもの) のハッシュは、個々のデータ・ブロックのハッシュを用いて導出することが可能である。

【0082】

【0111】 「線型コード (linear codes)」は、冗長コードの族であり、データ・セグメントの線型結合は冗長情報として作成される。線型コードの例は、リード・ソロモン・コード (Reed-Solomon code)、RaptorQコード、ランダム線型 (ネットワーク) コード、ハミング・コード、畳み込みコード、低密度パリティチェック (low-density parity check, LDPC) コード、及び BCH (Bose-Chaudhuri-Hocquenghem) コードを含むが、これらに限定されない。

【0083】

【0112】 「線型符号化」は、データ・セグメントを線型結合することによって線型冗長性を生成する演算である。

【0084】

【0113】 「線型復号化 (linear decoding)」は、元のデータを取り出すために線型方程式を解く演算である。

【0085】

【0114】 「線型再符号化 (linear re-encoding)」は、復号化して符号化するか又は線型結合を更に線型結合することによって、新しい線型冗長性を生成する演算である。

【0086】

【0115】 「メタデータ」は、メディア・タイプ、フレーム・レート、ビット・レート等のような、受信側がメディア・データを処理するのに役立つハイ・レベル記述データである。

【0087】

【0116】 「後処理」は、例えば、アンパッキング (unpackaging) 及び/又は解凍を適用することによって、配信されたデータを消費可能なメディアに変換する処理である。

【0088】

【0117】 「前処理」は、生のメディア・データを適切なフォーマットに処理することである。例えば、ソース・コーディング技術 (例えば、圧縮) を、生のメディア・データに適用すること、及び/又は、HTTPライク・ストリーミング (HLS) 及びHTTPにおけるダイナミック適応ストリーミング (DASH) のような特定の表現を使用してメディア・データをパッケージ化することである。

[0118] アプリケーション層コーディングの概要

開示される実施形態を読者がより良く理解することを支援するために、以下、アプリケーション層コーディング、特に線型コーディングの概要を説明する。

[0089]

[0119] アプリケーション層コードの大部分は線型コードであり、データを符号化することは、典型的には、3つの基本ステップに従う：

1) データを多数のブロックに分ける。ブロックの数は、データのサイズに応じて、小さくてもよいし（例えば、1）大きくてもよい。；

2) 各ブロックを多数のソース・シンボル X_i に分ける。 i はブロック内の元のシンボルのインデックスであり、各ソース・シンボルは指定されたサイズ（例えば、バイト単位）におけるものであり、最終的なソース・シンボルは、必要に応じてゼロ値バイトでパディングされ、ソース・シンボルの数は、典型的には、使用されるアプリケーション及びコーディング・タイプに基づいて選択される。；及び

3) ソース・シンボルを符号化して、（コーディング・タイプに応じて）同等以上の数のコーディングされたシンボル Y_i を作成する。

[0090]

[0120] 線型コード符号化及び線型コード復号化ステップの数学的表現は、それぞれ図1A及び図1Bに示されている。図示されている例では、元のデータ・ファイル又はストリームは複数のブロックに分けられる。この例は、1つのブロックの線型符号化に焦点を当てているが、線型コード符号化/復号化ステップは、データ・ファイル又はストリーム内の各ブロックに対して反復される。

[0091]

[0121] 図1Aを参照すると、一般的な線型符号が与えられている条件下で、データを符号化するステップは、以下のとおりである。

[0092]

ステップ1：データの各ブロックを、多数のソース・シンボルに分ける。この例では、4つのソース・シンボルが入力ブロック・データから作成されている。

[0093]

ステップ2：ソース・シンボルのベクトルを定式化する（ X^A ）。

[0094]

ステップ3：コーディング行列（ G ）を生成する。行列の各要素は、ガロア体 $GF\{q\}$ から選択され、 $q=2^m$ である。この例では、コーディング行列は 6×4 である。

[0095]

ステップ4：コーディング行列（ G ）と入力ソース・シンボル（ X^A ）を乗算することによって、コーディングされたシンボル（ Y ）を計算する。 $GF\{2\}$ のガロア体が使用され、 y_1^A の係数ベクトルが $[a_{1,1}, a_{1,2}, a_{1,3}, a_{1,4}] = [1, 0, 0, 0]$ であるように選択される場合、第1のコーディングされたシンボル y_1 は第1のソース・シンボル x_1 と同等であり、なぜなら係数ベクトルは唯1つの非ゼロ係数しか有していないからである。この場合に、 y_1 は系統的シンボルと言及することが可能である。

[0096]

[0122] 符号化の後に、コーディングされたシンボル及びコーディング行列のコーディング係数は、デコーダへ伝送される。他の実施形態において、符号化されたシンボルは、ネットワーク、CDN等においてキャッシュされ、適切なデコーダを含む1つ以上のクライアント・デバイスからの要求に応じて配信される。コーディング行列のコーディング係数は：

(a) デコーダへ伝送される；

(b) コーディング係数を生成する手段がパラメータ化され、デコーダへ伝送される；又は

(c) 係数は既知であり、エンコーダとデコーダの両方で想定されている、とすることが可能である。

【0097】

【0123】 デコーダにおけるコーディング係数の知識を用いて、例えば、ガウス除去のようなアプローチを使用して、コーディングされたシンボルから元のシンボルを解くことが可能である。このプロセスは、系統的シンボルが存在する場合にはシンプルになり、なぜなら系統的シンボルは線型独立であるように保証されているからである。少なくとも、ソース・シンボルの数以上の数のコーディングされたシンボルが、復号化するためのデコーダによって受信されるべきである。場合によっては、生成されるコーディング係数が互いに線型独立でない場合には、より多くのコーディングされたシンボルが必要とされる。

【0098】

【0124】 図1Bを参照すると、上記の例示的な線型コードを復号可するステップは、以下のとおりである。

【0099】

ステップ1：各々のコーディングされたシンボルに関連する係数ベクトルを決定し、受信コーディング係数行列Aを生成する。

【0100】

ステップ2：Aのインバース、 A^{-1} を算出する。

【0101】

ステップ3：受信コーディング係数行列のインバース (A^{-1}) と受信したコーディングされたシンボルのベクトル ($Y^{\wedge}$) とを乗算することにより、元のシンボルのベクトル ($X^{\wedge}$) を復元する。

【0102】

ステップ4： $X^{\wedge}$ 内の各シンボルを連結して、元のブロック・データを再構成する。

【0125】 コンテナとしての開示されるビットストリーム

図2A及び2Bはそれぞれ一部の実施形態によるビットストリームの符号化及び復号化のブロック図である。開示されるビットストリーム (図3A, 3B参照) は、異なるアプリケーション層のコーディング・タイプをサポートするコンテナと考えることができる。特定のコーディング・タイプが、特定のユース・ケースに対して選択される可能性がある。多くのコーディング・タイプは、異なる種類のビットストリームを生成するそれら独自のフォーマットを有するが、開示されるビットストリームの目標は、共通の表現及びインターフェースを用いて異なるアプリケーションをサポートすることである。

【0103】

【0126】 アプリケーション層コーディングが使用される場合、情報は、元のデータを復元する受信側に伝送される。例えば、コーディングされたシンボルに加えて、ブロック・カウント、ブロック当たりのシンボル数、コーディング行列に関する情報、及び、コーディング行列係数のフィールド・サイズ (field size) が、受信機へ送信される。図3A、図3Bに示される開示されるビットストリームは、メタデータとこの情報を、互換性のあるフォーマットで搬送するように設計され、異なるコーディング・タイプの標準化されたエンコーダ及びデコーダの実装に従う。ビットストリーム・マッパーは、アプリケーション層エンコーダによって生成された情報を、開示されるビットストリームによってサポートされるデータ要素タイプに変換するために使用される。

【0104】

【0127】 図2Aを参照すると、システム200aは、メディア・ファイル201、アプリケーション層エンコーダ202a...202n、ビットストリーム・マッパー203、及びビットストリーム・エンコーダ204を含む。ビットストリーム・マッパー203は、ビットストリーム・エンコーダ202a...202nによって生成された情報を、ビットストリーム・エンコーダ204によって生成される開示されるビットストリームによってサポートされるデータ要素タイプに変換する。ビットストリーム・エンコーダ204は、ビットストリーム・マッパー203によって出力されるサポートされるデータ要素タイプ及びコンテンツ・メタデータ205を含む開示されるビットストリームを生成する。

【0105】

10

20

30

40

50

【0128】 図2Bを参照すると、システム200bは、ビットストリーム・デコーダ/パーサー206、デマッパ207、及びアプリケーション層エンコーダ208a...208nを含む。ビットストリーム・デコーダ/パーサー206は、コンテンツ・メタデータ205を使用して、受信した開示されるビットストリームを復号化し、開示されるビットストリーム内のデータ要素を解析する。ビットストリーム・デマッパ207は、データ要素タイプを、標準化されたアプリケーション層デコーダ208a...208nによって必要とされるタイプに変換する。アプリケーション層デコーダ208...208nの出力は、コンテンツ・メタデータ205とともにメディア・ファイル209に結合される。

【0129】 開示されるビットストリーム構造の例

図3Aは、一部の実施形態による開示されるビットストリーム301の階層的な組織構造300を示す。開示されるビットストリーム301は、既存の方法によって現在提供されているものよりも、マルチメディア・アプリケーションに関して幾つもの新しい、より効率的で信頼性の高い記憶、キャッシング、分配、及び配信パラダイムを可能にする拡張可能なコンテナを提供する。これらの新しいマルチメディア・アプリケーションは、単一の及び/又は複数の通信チャネル/ネットワークを介してメディアを配信するための、今日のソース・コーディング及び非常に重複した記憶方法を利用する手の届く範囲内にはない。より具体的には、開示されるビットストリーム301は、クラウド及びネットワーク接続された、マルチメディアの処理、記憶、分配、配信、及び再生のネットワーク側の効率及び信頼性を高めるために、1つ以上の線型コードの使用をサポートするスケーラブルで相互運用可能な方法を可能にする。

10

20

【0106】

【0130】 例示的な実施形態において、ソース・コーディングされたマルチメディア資産は、1つ以上の線型コーディング・プロセスによって、1つ以上の機能的に等価なバイナリ表現又は「ビットストリーム」にマッピングされ、図3Aに示す開示されるビットストリーム301のフォーマットに従って複数の機能的に等価な (functionally equivalent, FE) ビットストリームを生成する。一部の実施形態において、線型コーディング・プロセスによって生成されるFEビットストリームのそれぞれは、SYNCサブアトム・データ構造302で始まる。SYNCデータ構造302で定義される情報は、開示されるビットストリーム301と同期し、それを識別し、それを解釈する後続のプロセスに使用される。一部の実施形態において、SYNCデータ構造302とSYNCデータ構造302で定義された情報は、再同期を必要とするアプリケーションのために、サブアトム (ビットストリーム・データ構造) 内に含まれ、開示されるビットストリーム301内の任意のポイントに注入されることも可能である。SYNCデータ構造302内に含まれる情報は、開示される実施形態で説明されるアプリケーションのために生成されるビットストリームに、ユニバーサル・エントリ・ポイント (universal entry point) を提供し、開示される実施形態は、将来の線型コード・タイプ及びサブアトムを有する将来のアプリケーションとの前方互換性を有するものを含むが、これらに限定されない。

30

【0107】

【0131】 SYNCデータ構造302に続いて、開示される各ビットストリーム301は、1つ以上のアプリケーションを動作可能にするために必要な情報を含む複数のサブアトム・データ要素303a...303n (以下「サブアトム」と言及される) を含む。各サブアトム303a...303nは、開示されるビットストリーム301が、1つ以上の固有のサブアトムと1つ以上のアプリケーションを動作可能にするために必要な情報とを記述及び搬送することを可能にするsubatom_idフィールド (図示せず) を含む。上述したように、開示されるsubatom_idビットフィールドを使用することは、開示されるビットストリーム301のフォーマットのための幾つかの固有のサブアトム・データ構造の定義及び使用を可能にする。

40

【0108】

【0132】 図3Aに示すように、各サブアトム303a、303b、303c...303nは、異なるタイプのデータを配置することが可能な汎用コンテナである。サブアトムは、唯1つのタ

50

タイプのデータのみを搬送することが可能であるので、サブアトム内で搬送されるデータのタイプは、サブアトム・タイプ及び関連するデータ構造を定める。例えば、BITSTREAM_HEADERサブアトム305は、ビットストリーム・ヘッダー構文構造 (bitstream header syntonical structure) (BITSTREAM_HEADER) を搬送するサブアトム303a構文構造のインスタンスである。

【0109】

【0133】 以下、サブアトム及びそれら各自のビットフィールドは、より一般的に「データ要素」と言及される可能性がある。更に、サブアトム・タイプは、その名称（例えば、ビットストリーム・サブアトム）又はその関連する構文構造 (BITSTREAM_HEADER()) によって言及される場合がある。一部の実施形態において、各サブアトム303a, 303b, 303c...303nは、以下の構文構造：

SYNC 304,
BITSTREAM_HEADER 305,
MEDIA_SEGMENT_INFO 306,
ENCODER_CONTENT_INFO 307,
BLOCK_HEADER 308,
PACKET 309,
BLOCK_DIRECTORY 310,
CHUNKED_SUBATOM 311 及び
PACKET_HEADER_ONLY 312

を含んでいる。有効な開示されるビットストリーム301は、単一のサブアトムを含むことが可能であるが、デコーダは、開示されるビットストリーム301を首尾良くデコードするために、BITSTREAM_HEADER, BLOCK_HEADER(S)及び PACKET(S) を含む複数のサブアトムを受信する。構文は、追加のサブアトム・データ・タイプが、将来、新しいアプリケーションのために定義されることを許容している。

【0110】

【0134】 図3Bは、一部の実施形態による開示されるビットストリーム301のコンテナ・フォーマットの各サブアトムにおけるデータ要素を更に示す。以下で説明されるサブアトムの一意のインスタンスによって搬送されるデータは、独立しているか、又は他のサブアトムに依存しているかの何れかである。サブアトムの正しい解析/解釈及び/又は復号化は、以前のサブアトム・インスタンスからのデータが、正しく解析及び/又は処理されていることを必要とする場合がある。サブアトムのデータ要素について以下で説明される。

【0111】

【0135】 SYNCサブアトム304は、同じソース・メディア資産から生成される2つのビットストリームを識別及び同期するために使用される。SYNCサブアトム304のデータ要素（例えば、ビットフィールド）の意味/使用法の幾つかの例を以下に説明する。

【0112】

●syncword：一意のバイト・シーケンスであって、開示されるビットストリームが、開示されるビットストリーム301のフォーマットに準拠していることを識別するために使用されるものである。

【0113】

●version：後続するビットストリーム・シンタックスの具体的なバージョン。このビットフィールドは、ビットストリームが後方及び前方互換性のある方法で拡張されることを可能にする。

【0114】

●content_encode_uuid：固有の識別子であって、一例においては、マルチソース・メディア・システム（例えば、エンコーダ、デコーダ、再エンコーダ、パーサー、プロセッサなどを含むもの）が、メディア資産の具体的な‘encode’を暗黙的にシグナリングすることを可能にするものである。一例では、この情報は、下流プロセスが、1つ以上

10

20

30

40

50

のエンコーダ又は再エンコーダによって生成される符号化されたビットストリームの系譜を効率的（且つ独立的）に決定するために有用である。このレベルの系譜の暗黙的なシグナリングを用いて符号化されたビットストリームは、全てのコーディングされたメディア・ソース（ネットワーク内のストレージ/キャッシュを含む複数のノードに分散され利用可能であるもの）が、一意で機能的に同等なコーディングされたメディア表現を含むことを保証することによって、マルチソース・メディア・システムが最も効率的に動作可能になる点で有利である。

【0115】

【0136】 BITSTREAM_HEADERサブアトム305は、使用される線型符号化方法のような線型符号化されたデータに関するハイ・レベル汎用ファイル情報、ビットストリーム・プロファイル情報、及び、ビットストリームの一部が暗号化されているかどうかの指示を含む。bitstream_headerビット・フィールドの意味／使用法の幾つかの例は以下の通りである。

10

【0116】

●content_source_size：線型符号化されたメディアのサイズをバイト単位で示す。

【0117】

●content_source_type：符号化されたストリームが元のコンテンツ、符号化されたバリエーション（encoded variant）、又はライブ・ストリームからソース供給されているかどうかのような、後続の線型符号化されたメディア・ソースに関するより多くの情報を示す。

20

【0118】

●linear_code_type：コンテンツを符号化するためにどのタイプの線型コードが使用されているかを示す。

【0119】

●profile_information：アプリケーションで有用な特定のプロファイルに準拠したビットストリームの処理が、制限された又は限定された機能を有する特定のデコーダー/パーサーに特定のビットストリームが適しているかどうかを効率的に判定できるようにする。

【0120】

●encrypted_coefficients：コーディング係数の暗号化がアクティブであることの暗黙的なシグナリングをイネーブルにする。

30

【0121】

【0137】 コーディング係数の暗号化は、デジタル著作権管理の一形態と見なすことが可能であり、従って、線型コーディングされたメディア資産自体を完全に暗号化する必要性を軽減又は排除することが可能である。

【0122】

【0138】 MEDIA_SEGMENT_INFOサブアトム306は、メディア・ストリーミング・アプリケーションに使用され、ここで、ネットワーク及びオーバー・ザ・トップ（over-the-top, OTT）メディア・ストリーミング・アプリケーション/サービスにおいて、コンテンツはセグメント（例えば、オーディオ及びビデオ・セグメント）に分割される。これ以降、MEDIA_SEGMENT_INFOは、より一般的に「メディア・セグメント・データ」とも言及される。MEDIA_SEGMENT_INFOサブアトム306の構造は、オーディオ/ビデオ・プロパティ（例えば、フレーム・レート、ビットレート、コーデック・タイプ等）のような、セグメント内のメディアに関する情報を提供する。この情報は、典型的には、マニフェスト・ファイル（例えば、DASHマニフェスト又はHLSマニフェスト）に含まれる可能性がある。この情報はこの構造内に含まれて、線型符号化されたデータを検査するためにビットストリームを復号化する必要性を回避する。MEDIA_SEGMENT_INFOサブアトム306のフィールドの具体例は以下の通りである：

40

●segment_index：現在のサブアトムで表されるセグメントのメディア・インデ

50

ックスを識別する。

【0123】

●content_type：オーディオ、ビデオ、テキスト、アプリケーション・データのような、このメディア・セグメントに含まれるマルチメディアのタイプを示す。

【0124】

●aspect_ratio：ビデオの幅と高さの間の比率

●frame_rate：ビデオ・フレーム数／秒

●audio_channel_config：オーディオのコンフィギュレーション（例えば、チャンネル、ロケーション、オブジェクト、仮想化されていること等）。

【0125】

【0139】 ENCODER_CONTENT_INFOサブアトム307は、基になるメディア・コンテンツを表す一意の識別子とともに、ビットストリームを作成するために使用される特定の線型エンコーダを識別する。ENCODER_CONTENT_INFOサブアトム307はまた、基になるストリーム/ファイル/ブロック情報の完全性（integrity）（信頼性（trustworthiness））の検証を可能にする情報、及び、復号化される場合の基になるストリーム/ファイル/ブロック情報の持続時間に関する情報を搬送する。ENCODER_CONTENT_INFOサブアトム307の意味／使用法の幾つかの例は以下の通りである。

【0126】

●encoder_uuid：一意の識別子であって、一例では、マルチソース・メディア・システム（例えば、エンコーダ、デコーダ、レコーダ、パーサー、プロセッサなどを含むもの）が、1つ以上のエンコーダ又はレコーダによって生成された符号化されたビットストリームの系譜を本質的に通知すること（intrinsically signal）を可能にするものである。系譜の本質的な通知を伴う符号化されたビットストリームは、開示された実施形態の重要な機能であり、全てのコーディングされたメディア・ソース（ネットワーク内のストレージ/キャッシュを含む複数のノードに分散して利用可能なもの）が、メディアの一意かつ機能的に等価なコーディングされた表現で構成されることを保証することによって、マルチソース・メディア・システムが最も効率的に動作することを可能にする。

【0127】

●content_id_type：EIDR（Entertainment Identification Registry）、Ad-ID（Advertising Digital Identifier）、又はユーザー定義値のような、コンテンツを識別するために使用される識別フォーマットのタイプを示す。

【0128】

●content_id：content_id bitfieldに従ってID値を保持する。

【0129】

【0140】 一部の実施形態において、上述のデータ要素は、分散システム全体（コンテンツ中心のネットワークを含む）の処理エンティティが、メディアをデコードしたり、セントラル化されたサーバーと通信したり、ストリーム/セグメント・マニフェストを探したりする必要性なしに（適用可能な場合）、基礎となるコーディングされたメディアに関する情報を効率的に識別することを可能にする。この機能は、HTTP delivery.file_integrityを利用しないことになる将来のメディア配信パラダイムを実現する幾つかの重要な要素のうちの1つである：一意のデータ構造であって、受信及び復号化されたストリーム/ブロックが元のソースのものと一致することを検証するメカニズムを提供するものである。delivery.file_integrity構造は、fb_integrity()ビットフィールドに対するマッピングであり、これは、例えば、ファイル/ストリーム/ブロックのハッシュに使用されるハッシュ関数、ハッシュ・サイズ、及びハッシュ値などを含む情報を更に指定する。一例では、fb_integrity()ビットフィールドは、ネットワーク内の任意の処理ノードが、コーディングされたマルチソース・ビットストリームの信頼性を独立に検証できるようにする。この機能は、モバイル・デバイス、パーソナル・コンピュータ（PC）、セット・トップ・ボックス、デジタル・メディア・アダプター、ホーム・ゲートウェイのような、従来は信頼されていなかった

10

20

30

40

50

たメディア・ソースを使用して、マルチソース・コーディングされたメディアを格納するために活用できるようにする。

【0130】

【0141】 BLOCK_HEADER 308は、ブロック・サイズ、シンボル・サイズ、シンボル数、有限体サイズのような、データのブロック内のデータに適用される線型符号化パラメータに関する情報を含む。BLOCK_HEADERデータ要素の意味/使用法の例は以下の通りである。

【0131】

●block_index：マルチソース・コーディングされたメディア・セグメントは、ブロックに分けられる可能性があり、各ブロックは個別に符号化される。このフィールドはブロックのインデックスを示す。

10

【0132】

●symbol_size：系統的及び線型コーディングされたシンボルのバイト単位のサイズ。

【0133】

●number_of_symbols：ブロックが線型コーディングのために分けられるコーディングされたシンボルの個数をシグナリングする。

【0134】

●field_size：線型コーディングが作用する基礎となる有限体のサイズ。

【0135】

●encryption

parameters（暗号化パラメータ）：これらのフィールドは、暗号化されたコーディング係数を解読するために使用される。

20

【0136】

●prng_seed：線型コードは擬似乱数生成器を使用して、現在のブロックに対するランダム化されたコーディング判定を生じさせることが可能である。Prng_seedは、デコーダー又はレコーダーで擬似乱数生成器を初期化するために使用されるシードである。

【0137】

●block_integrity：受信及び復号化されたブロックが元のソースのブロックと一致することを確認するメカニズムを提供する構造。構造は、ブロックのハッシュに使用されるハッシュ関数、ハッシュ・サイズ、実際のハッシュのような情報を含むことが可能である。一例において、実装はSHA1又は準同型ハッシュを使用して、現在のブロック内の全ての系統的シンボルをハッシュ演算し、次いで、マークル・ツリー（Merkle tree）のようなハッシュ・アグリゲーターを使用して、シンボル・ハッシュを統合し、ブロック・ハッシュを取得することが可能である。

30

【0138】

【0142】 PACKETサブアトム309は、線型符号化されたデータの基本的な部分、即ち、コーディングされたシンボルを含む。これはまた、系統的シンボルであるかどうか、ウィンドウ・サイズ、及びコーディング係数のような、このコーディングされたシンボルに関する情報を提供するパケット・ヘッダー・データ構造も含む。PACKETサブアトム309のデータ要素の例は次の通りである：

40

●packet_header：一群のビット・フィールドであって、パケットを復号化することを支援するものであり、例えば以下のようなものである：

1) block_index：セグメントはブロックに分けられる可能性があり、各ブロックは個別に符号化される。このフィールドは、このパケットが属するブロックのインデックスを示す。

【0139】

2) b_systematic：このパケットは、系統的シンボル又は線型コーディングされたシンボルのどちらを搬送しているかを示す。

【0140】

50

3) `poly_index`: これは、パケットをインデックス化するために使用される汎用フィールドである。これは、系統的パケットの元のシンボル・インデックスを識別するために使用されることが可能である。これはまた、パケット・ヘッダーのみのサブアトムとパケット・サブアトムを同期させるために使用されることも可能である。

【0141】

4) `coeff_vector`: 系統的パケットを線型結合してこのシンボルを生成するために使用される有限体スカラーのシーケンス (the sequence of finite field scalars)。

【0142】

5) `window_start_index`: パケットは系統的シンボルの連続したサブセット (即ち、ウィンドウ) を使用して生成されることが可能である。このフィールドは、このウィンドウの最初の系統的シンボルのインデックスを示す。

【0143】

6) `window_start_index`: パケットは系統的シンボルの連続したサブセット (即ち、ウィンドウ) を使用して生成されることが可能である。このフィールドは、このウィンドウの最後の系統的シンボルのインデックスを示す。

【0144】

7) `prng_seed`: このシンボルの係数を生成するために使用される擬似乱数生成kのシード。

【0145】

8) `packet_integrity`: このパケットを復号化又は再符号化に使用する前に、受信したパケットが正しく生成されていることを検証するメカニズムを提供する構造。構造は、パケットをハッシュ演算するために使用されるハッシュ関数、ハッシュ・サイズ、及び実際のハッシュのような情報を含むことが可能である。

【0146】

【0143】 一部の実施形態において、最小の `packet_header` が、`PACKET` サブアトム 309 に含まれる可能性はあるが、完全な `packet_header` を個別に/独立して格納し、配信することが可能である (例えば、帯域外 (out of band,)、別のネットワーク・セッションなど)。最小の `packet_header` のみでは、ペイロードはデータを復号化するために使用できない可能性がある。このアプローチは、例えば、完全な `packet_header` へのアクセスを抑制することによって、デジタル著作権管理の追加的且つ簡単な方法を可能にする。

【0147】

【0144】 `BLOCK_DIRECTORY` サブアトム 310 は、開示されるビットストリーム 301 内のブロック・ヘッダー・サブアトム・インスタンスの開始を示すフィールドを含む。一部の実施形態において、ブロック・ディレクトリ・サブアトム 310 は、以下のフィールド: `block_header_subatom_offsets()` の 1 つ以上を含み、これは、同じ開示されるビットストリーム 301 内の `BLOCK_DIRECTORY` サブアトム 310 の末尾から `BLOCK_HEADER` サブアトム 308 の先頭までのオフセットをバイト単位で記述する。

【0148】

【0145】 `CHUNKED_SUBATOM` 311 は、大きなサブアトム・データを、より小さなチャンクに分けるメカニズムを提供する。例えば、`PACKET` サブアトム 309 内のコーディングされたシンボル・サイズが、使用する伝送プロトコルに対して大き過ぎる場合 (例えば、UDP パケットは約 1400 バイトのペイロードを有する)、`PACKET` サブアトム 309 は、伝送要件に合うより小さなチャンクに分割されることが可能である。各チャンクは、次いで、1 つの `CHUNKED_SUBATOM` 311 によって搬送される。幾つかの例示的なビット・フィールドは以下の通りである。

【0149】

● `chunked_segment_id`: 同じサブアトム・データの異なる部分を含むものとして、一連の `chunked_subatom` インスタンスを識別する。サブアトム・データがチャンクに分けられると、作成された全ての `CHUNKED_SUBATOM` 311 インスタンスは同じ

10

20

30

40

50

chunk_segment_idフィールドを有するものとする。

【0150】

●chunked_segment_index: chunked_subatomインスタンス内に含まれるサブアトム・データのチャンク・セグメントのインデックスを識別する。パーサーはこの情報を使用して、チャンクに分けられたサブアトム・データを元の順序に再構成することが可能である。

【0151】

●num_chunked_segments: (所与のchunk_segment_idに関連する) サブアトム・データが分割されているチャンク・セグメントの個数を記述する。

【0152】

●original_subatom_id: このchunked_subatomインスタンス内に含まれるセグメント・データのsubatom_idを指定する。

【0153】

●chunked_data: このchunked_subatomインスタンスによって搬送されるデータのチャンク・セグメント。

【0154】

[0146] PACKET_HEADER_ONLYサブアトム312は、実際のコーディングされたシンボルを含まない点を除いて、PACKETサブアトム309と同一である。

[0147] メディア処理及び配信システムの例

図4は、一部の実施形態によるメディア処理&配信システム400のブロック図である。システム400は、前処理ブロック401a, 401b...401n, ビットストリーム生成器402a, 402b...402n, 通信ネットワーク403, ビットストリーム・デコーダ404, 及び後処理ブロック405を含む。ネットワーク/クラウド403は、再処理406及びビットストリーム再生成器407を含む。一部の実施形態において、システム400は、例えば、コンテンツ配信ネットワーク (content delivery network, CDN) サーバー、ポイント・オブ・プレゼンス (Point-of-presence, PoP) サーバー、アプリケーション・サーバー (例えば、電子ニュース収集アプリケーション、コンテンツ放送アプリケーション、ユニキャスト・アプリケーション等をホストするサーバー)、コンテンツ・キャッシュ、ピア・デバイス、ローカル・キャッシング・デバイス、エッジ・デバイス (例えば、5G/6Gエッジ・アプリケーション・サーバー)、又はクライアント・デバイス (例えば、メディア消費デバイス、コンピュータ、モバイル・デバイス、ゲーミング・システム、ストリーミング・デバイス、オーディオ/ビデオ機器など) のような、1つ以上のサーバーで部分的又は完全に実装することが可能である。

【0155】

[0148] この例示的な実施形態において、メディア・データは、1つ以上のサーバーに供給される。メディア・データは、メディア資産から取得され、画像、オーディオ、ビデオ、ゲーム状態、ゲーム・コマンドなどを含むがこれらに限定されない。一部の実施形態において、メディア・データは、ソース・コード化データ、ストリーミング・プロトコル (例えば、HLS, DASH等) に従ってパッケージ化されたデータである。一部の実施形態において、メディア・データは、メディア資産 (例えば、コーディングされたコンテンツ及び関連するメタデータを含むコーディングされたビットストリーム) に関連するコーディングされたビットストリーム及び線型コーディングされたデータである。一部の実施形態において、メディア資産を表すメディア・データは、複数のネットワーク経路から受信される。

【0156】

[0149] サーバー内の前処理ブロック401a...401nは、生のメディア・データ (raw media data) を前処理する。前処理の例は、ソース圧縮技術を適用すること、及び/又は、例えばHLS及びDASH標準によって定義されるような生のメディア・データをパッケージングすることを含むが、これらに限定されない。サーバーはまた、ビットストリームを生成するために、それぞれのビットストリーム生成器402a...402nも含んでいる

。異なるビットストリーム生成器402a...402nは、異なるビットストリーム、例えば、異なるフォーマットや異なるペイロードを有するものを生成することが可能である。次いで、ビットストリームは、ビットストリームをキャッシュ、ルーティング、及び/又は転送することが可能な通信ネットワーク403を使用して伝送される。通信ネットワーク403はまた、ビットストリームを再処理406及び/又は再生成407することのように、それらを再送信する前にビットストリームを更に修正することが可能なエンティティを含むことも可能である。

【0157】

【0150】 ビットストリーム（例えば、コンシューマー・デバイス）の受信機は、1つ以上のビットストリームをダウンロードし、ビットストリーム・デコーダ404を用いて、処理されたメディアデータを復元する。次いで、処理されたメディア・データは、ポスト・プロセッサ405によって後処理され、使用の準備、例えば、エンド・ユーザーによって消費されるために読み込まれる準備が整う。オプションとして、ビットストリーム・デコーダ404は、フィードバック408をビットストリーム生成器402a...402nへ送信し、システム400内の上流プロセスが通信ネットワーク403内の動的な条件に適合することを支援する。

【0151】 ビットストリーム生成の例

図5は、一部の実施形態によるビットストリーム生成器500（例えば、図4のビットストリーム生成器402a...402nのうちの1つ）のブロック図である。前処理されたメディア・データ、及びオプションとしてシステム情報であって意思決定を支援することが可能であるものが、ビットストリーム生成器500に供給される。ビットストリーム生成器500は、図3A及び図3Bを参照して説明される複数のサブアトムaを含む開示されるビットストリーム301を生成する。

【0158】

この例示的な実施形態において、メタデータ生成器501は、前処理されたメディア・データを分析して、SYNCサブアトム304とMEDIA_SEGMENT_INFOサブアトム306を生成する。

【0159】

線型エンコーダ502は、例えば、図1A及び図1Bを参照しながら説明したように、メディア・データを元の（ソース）シンボルに分け、次いで元のシンボルの線型結合を、線型コーディングされたシンボルとして生成することによって、メディア・データに対して線型符号化を実行する。線型符号化情報は、符号化されたデータとともに、図3A及び図3Bを参照しながら説明されたデータ要素BITSTREAM_HEADER, ENCODER_CONTENT_INFO, BLOCK_HEADER, 及びPACKETサブアトムによって搬送され、ここで、各パケット・サブアトムは、1つの線型コーディングされたシンボルに対応する。

【0160】

【0152】 一部の実施形態において、セキュリティ・シーラー503は、ファイル、データ・ブロック、及びパケットに関するハッシュ化情報を、対応するENCODER_CONTENT_INFO, BLOCK_HEADER, 及びPACKETサブアトムに追加することによって、線型コーディングされたメディア・データに電子的に署名する。このステップは、開示される発明に従って生成されるコーディングされたビットストリーム/ファイルの独立した検証/認証のために、メディア・システム全体の後続の処理に必要な情報の内在的なシグナリング（intrinsic signaling）を可能にする。

【0161】

【0153】 別の例示的な実施形態において、ビットストリーム生成器500はまた、パケット・サブアトムからパケット・ヘッダ情報を分離し、専用のPACKET_HEADER_ONLYサブアトム312を使用してパケット・ヘッダ情報を伝え、対応するPACKETサブアトム309へのアクセスを制御することも可能である。PACKETサブアトム309又はPACKET_HEADER_ONLYサブアトム312の何れかにおけるコーディング係数は、メディア・コンテンツに保護層を加えるために暗号化される。一部の実施形態において、各サブアトム

は、サブアトム・トランケーター504によって更に打ち切られ、複数のCHUNKED_SUBATOMサブアトム311によって運ばれて、サブアトムのサイズを小さくすることができる。この実施形態は、トランスポート・プロトコルがデータ・ユニット・サイズに制限を有する場合に特に有用である。

【0162】

[0154] 次いで、サブアトムは、マルチプレクサ505 (“Muxer”)によって多重化され、1つ以上のビットストリーム301を形成する。異なるビットストリーム301は、異なるサブアトムを運ぶことが可能であり、別様に処理される(例えば、キャッシュされる、配信される)ことが可能である。受信側が各ビットストリーム内のサブアトムをナビゲートすることを支援するために、マルチプレクサ505は、BLOCK_DIRECTORYサブアトム310を生成し、オン・ザ・フライで各ビットストリームに挿入することが可能である。

10

【0155】 開示されるビットストリーム復号化の例

図6は、一部の実施形態によるビットストリーム・デコーダ600(例えば、図4に示されるビットストリーム・デコーダ404)のブロック図である。ビットストリーム・デコーダ600は、開示されるビットストリーム・フォーマットを有する1つ以上のビットストリームからデータを受信し、ここで、各ビットストリームは1つ以上のサブアトムを含む。ビットストリームのダウンロードは、ダウンロードを管理するダウンローダ601によって実行され(例えば、バッファを用いる)、ダウンロードの管理は、メディア・ソースを発見及び選択すること、接続を確立すること、接続をリスニングすること、及び接続を終端すること、並びに、各メディア・ソースからダウンロードされるデータの速度と量を制御することを含むが、これらに限定されない。一部の実施形態において、サブアトム・パーサー602は、ビットストリームを解析して、受信したCHUNKED_SUBATOMサブアトム311を抽出して一時的にキャッシュし、チャンク・サブアトム併合部603は、CHUNKED_SUBATOMサブアトム311をマージする。マージされたCHUNKED_SUBATOMサブアトム311は、次いで、セキュリティ/認証/完全性データがメディア資産に存在する場合に、セキュリティ検査部604によって検証される。

20

【0163】

[0156] 検証されたサブアトムは、デマルチプレクサ605によって分離され、メディア・メタデータが抽出されてポスト・プロセッサ607(例えば、オーディオ/ビデオ・デコーダ)に供給され、線型コーディングされたデータは線型デコーダ606に供給されて元のメディア・データを復元する。復号化されたメディア・データは、抽出されたメディア・メタデータとともに、後処理のためにポスト・プロセッサ607に供給され、その結果、消費可能なメディア・データを、ポスト・プロセッサ607の出力で取得できるようになる。

30

【0157】 開示されるビットストリームの再符号化の例

図7は、一部の実施形態によるビットストリーム再エンコーダ700のブロック図である。再エンコーダ700は、1つ以上の開示されるビットストリーム701からデータをダウンロードし、各ビットストリーム701は、1つ以上のサブアトム304-309, 312を含む。ダウンロードは、ダウンロードを管理するダウンローダ702によって実行され(例えば、バッファを用いる)、ダウンロードの管理は、メディア・コンテンツ・ソースを発見及び選択すること、接続を確立すること、接続をリスニングすること、及び接続を終端すること、並びに、各メディア・ソースからダウンロードされるデータの速度と量を制御することを含むが、これらに限定されない。サブアトム・パーサー703は、ビットストリーム701を解析して、受信したCHUNKED_SUBATOMサブアトム311を抽出して一時的にキャッシュし、チャンク・サブアトム併合部704は、CHUNKED_SUBATOMサブアトム311をマージする。マージされたCHUNKED_SUBATOMサブアトム311は、次いで、セキュリティ/認証/完全性データがメディア資産に存在する場合に、セキュリティ検査部705によって検証される。

40

【0164】

[0158] 検証されたサブアトムは、次いで、デマルチプレクサ706によって分離され

50

、メディア・メタデータは、SYNCサブアトム304とMEDIA_SEGMENT_INFOサブアトム306を再作成するために使用される一方、線型コーディングされたデータは、新しい線型コーディングされたデータを作成するために線型再エンコーダ707に供給される。
【0165】

【0159】 新しいな線型コーディングされたデータの作成は、様々な方法で実施することが可能である。例えば、再エンコーダ700は、先ず、線型復号化を適用して元のシンボルを復号化し、次いで、元のシンボルの新しい線型結合を、新しい線型コーディングされたシンボルとして作成することができる。別の例に関し、再エンコーダ700は、線型コーディングされたシンボルを直接的に線型結合して、新しい線型コーディングされたシンボルを作成することが可能である。

10

【0166】

【0160】 一部の実施形態において、線型符号化情報は、線型符号化データとともに、BITSTREAM_HEADERサブアトム、ENCODER_CONTENT_INFOサブアトム、BLOCK_HEADERサブアトム、PACKETサブアトム、及びPACKET_HEADER_ONLYサブアトムによって伝送され、各パケット・サブアトムは、1つの新しい線型コーディングされたシンボルに対応する。セキュリティ・シーラー708は、ファイル、データ・ブロック、及びパケットに関するハッシュ化情報を、対応するENCODER_CONTENT_INFO、BLOCK_HEADER、及びPACKETサブアトムに追加することによって、メディア・データに電子的に署名する。

【0167】

20

【0161】 一部の実施形態において、各サブアトムは、サブアトム・トランケーター709によって更に打ち切られ、複数のCHUNKED_SUBATOMサブアトムによって運ばれて、サブアトムのサイズを小さくすることができる。これは、トランスポート・プロトコルがデータ・ユニット・サイズに制限を有する場合に特に有用である。次いで、サブアトムは、マルチプレクサ710によって多重化され、1つ以上のビットストリーム711を形成して出力する。異なるビットストリームは、異なるサブアトムを運ぶことが可能であり、別様に処理される（例えば、キャッシュされる、配信される）ことが可能である。受信側が各ビットストリーム内のサブアトムをナビゲートすることを支援するために、マルチプレクサ710は、BLOCK_DIRECTORYサブアトムを生成し、各ビットストリームに挿入する。

30

【0162】 統合デコーダの例

図8Aは、一部の実施形態による、異なる線型コーディング方法を実施した複数のビットストリームからメディア・データを復元することが可能な統合デコーダ800aのブロック図である。異なるビットストリーム801は、異なる線型コードであって、ビットストリーム801のBITSTREAM_HEADERサブアトム305内のcode_typeビットフィールドごとにシグナリングされるものを使用することができる。ダウンロードは、ダウンロードを管理するダウンローダ802によって実行され（例えば、バッファを用いる）、ダウンロードの管理は、メディア・ソースを発見及び選択すること、接続を確立すること、接続をリスニングすること、及び接続を終端すること、並びに、各メディア・ソースからダウンロードされるデータの速度と量を制御することを含むが、これらに限定されない。サブアトム・パーサー803は、ビットストリームを解析して、受信したCHUNKED_SUBATOMサブアトム311を抽出して一時的にキャッシュし、これらはチャンク・サブアトム併合部804によってマージされる。マージされたCHUNKED_SUBATOMサブアトム311は、次いで、セキュリティ・シーリング（securiy sealing）がこのメディア資産に対してイネーブルにされている場合に、セキュリティ検査部805によって検証される。

40

【0168】

【0163】 検証されたサブアトムは、デマルチプレクサ806によって分離され、メディア・メタデータが抽出されてポスト・プロセッサ809（例えば、オーディオ/ビデオ・デコーダ）に供給され、異なる線型コーデックによる線型コーディングされたデータはコーディング統合部807に供給される。コーディング統合部807は、例えば、係数を整合さ

50

せることによって、及び/又はコーディングされたデータを同じ有限体に投影することによって、異なる線型コーデックからの線型コーディングされたデータを統合する。次いで、統合された線型コーディングされたデータは線型デコーダ808に供給されて、元のメディア・データを復元する。次いで、復号化されたメディア・データは、抽出されたメディアメタデータを使用して後処理するためにポスト・プロセッサ809に供給され、その結果、消費可能なメディア・データをポスト・プロセッサ809の出力で取得することが可能になる。

【0169】

【0164】 図8Aは、一部の実施形態によるデコーダ・スイッチ810を用いる統合デコーダ800bのブロック図である。統合デコーダ800bは、複数の異なるデコーダ・タイプをカプセル化する線型デコーダ・モジュール808を含む。モジュール808は、デコーダ・スイッチ810を介して特定タイプの線型コーデックへのアクセスを可能にする。より具体的には、デコーダ800bは、1つ以上のビットストリーム801からデータを受信し、各ビットストリーム801は、1つ以上のサブアトム303（図3A及び3Bを参照されたい）から構成される。異なるビットストリーム801は、異なる線型コーデックを使用する。ダウンロードは、ダウンロードを管理するダウンロード802によって実行され（例えば、バッファを用いる）、ダウンロードの管理は、メディア・ソースを発見及び選択すること、接続を確立すること、接続をリスニングすること、及び接続を終端すること、並びに、各メディア・ソースからダウンロードされるデータの速度と量を制御することを含むが、これらに限定されない。

【0170】

【0165】 一部の実施形態において、サブアトム・パーサー803は、ビットストリーム801を解析して、受信したCHUNKED_SUBATOMサブアトム311を抽出して一時的にキャッシュし、これらはチャンク・サブアトム併合部804によってマージされる。マージされたCHUNKED_SUBATOMサブアトム311は、次いで、セキュリティ・シーリングがこのメディア資産に対してイネーブルにされている場合に、セキュリティ検査部805によって検証される。そして、セキュリティ・シーリングがメディア資産に対してイネーブルにされている場合、完全なサブアトムがセキュリティ検査部805によって検証される。

【0171】

【0166】 検証されたサブアトムは、次いで、デマルチプレクサ806によって分離され、メディア・メタデータが抽出されてポスト・プロセッサ809（例えば、オーディオ/ビデオ・デコーダ）に供給され、固有の線型コーデックを用いて、統合線型デコーダ・モジュール808に供給される。統合線型デコーダ・モジュール808は、使用される線型コーデックのタイプであって開示されるビットストリームの`bitstream_header()`内の`code_type`ビットフィールドごとにシグナリングされるものを識別し、デコーダ・スイッチ801を適切な線型デコーダ・タイプに動的に起動して切り替えて、元のメディア・データを復号化する。次いで、復号化されたメディア・データは、抽出されたメディア・メタデータの助けを借りて後処理のためにポスト・プロセッサ809に供給され、その結果、消費可能なメディア・データをポスト・プロセッサ809の出力から取得することが可能になる。

【0167】 統合ビットストリーム再エンコーダの例

図9Aは、一部の実施形態による統合ビットストリーム再エンコーダ900aのブロック図である。再エンコーダ900aは、異なる線型コーデックを用いる1つ以上のビットストリームから新たな（線型独立なものを含む）ビットストリームを作成することが可能である。より具体的には、再エンコーダ900aは、データを1つ以上のビットストリーム901からダウンロードし、各ビットストリーム901は、1つ以上のサブアトム303（図3A及び図3Bを参照されたい）を含む。ダウンロードは、ダウンロードを管理するダウンロード902によって実行され（例えば、バッファを用いる）、ダウンロードの管理は、ソースを発見及び選択すること、接続を確立すること、接続をリスニングすること、及び接続

を終端すること、並びに、各メディア・ソースからダウンロードされるデータの速度と量を制御することを含むが、これらに限定されない。サブアトム・パーサー903は、ビットストリーム901を解析して、CHUNKED_SUBATOMサブアトム311を抽出して一時的にキャッシュし、これらはチャンク・サブアトム併合部904によりマージされる。マージされるCHUNKED_SUBATOMサブアトム311は、次いで、セキュリティ・シーリングがメディア資産に対してイネーブルにされている場合に、セキュリティ検査部905によって検証される。

【0172】

【0168】 検証されたサブアトムは、次いで、デマルチプレクサ906によって分離され、メディア・メタデータは、SYNCサブアトム304とMEDIA_SEGMENT_INFOサブアトム306を再作成するために使用され、異なる線型コードを用いる線型コーディングされたデータは、コーディング統合部907に供給される。コーディング統合部907は、例えば、係数を整合させることによって、及び/又はコーディングされたデータを同じ有限体に投影することによって、異なる線型コーデックを用いる線型コーディングされたデータを統合する。統合された線型コーディングされたデータは線型再エンコーダ808に供給されて、新しい線型コーディングされたデータを作成する。

10

【0173】

【0169】 新しい線型コーディングされたデータの生成は、様々な方法で実施することが可能である。例えば、再エンコーダ900aは、先ず、線型復号化を適用して元のシンボルを復号化し、次いで、元のシンボルの新しい線型結合を、新しい線型コーディングされたシンボルとして作成することが可能である。別の例として、再エンコーダ900aは、線型コーディングされたシンボルを直接的に線型結合して、新しい線型コーディングされたシンボルを作成することができる。

20

【0174】

【0170】 一部の実施形態において、線型符号化情報は、線型符号化データとともに、BITSTERAM_HEADERサブアトム306、ENCODER_CONTENT_INFOサブアトム307、BLOCK_HEADERサブアトム308、PACKETサブアトム309、及びPACKET_HEADER_ONLYサブアトム310において伝送され、各パケット・サブアトム309は、1つの新しい線型コーディングされたシンボルに対応する。セキュリティ・シーラー909は、ファイル、データ・ブロック、及びパケットに関するハッシュ化情報を、対応するENCODER_CONTENT_INFOサブアトム307、BLOCK_HEADERサブアトム308、及びPACKETサブアトム309に追加することによって、メディア・データのセキュリティを封印する（例えば、電子署名する）。

30

【0175】

【0171】 一部の実施形態において、各サブアトムは、サブアトム・トランケーター910によって更に打ち切れられ、複数のCHUNKED_SUBATOMサブアトム311によって運ばれて、サブアトムのサイズを小さくすることができる。これは、トランスポート・プロトコルがデータ・ユニット・サイズに制限を有する場合に特に有用である。次いで、サブアトムは、マルチプレクサ911によって多重化され、1つ以上のビットストリーム912の出力を形成する。異なるビットストリーム912は、異なるサブアトムを運ぶことが可能であり、別様に処理される（例えば、キャッシュされる、配信される）ことが可能である。受信側が各ビットストリーム901内のサブアトムをナビゲートすることを支援するために、マルチプレクサ911は、BLOCK_DIRECTORYサブアトム310を生成し、各ビットストリーム901に挿入することが可能である。

40

【0176】

【0172】 図9Bは、一部の実施形態による再エンコーダ・スイッチ914を有する統合ビットストリーム再エンコーダ900bのブロック図である。統合ビットストリーム再エンコーダ900bは、図9Aに示される開示される統合ビットストリーム再エンコーダ900aの別の実装例を示しており、これは、複数のタイプの線型コーデックの再エンコーダを実装し、それを線型再エンコーダ・モジュール913内に入れる（encapsulate）ものである

50

。線型再エンコーダ・モジュール913は、スイッチ914を介して特定のタイプの線型コーデックへのアクセスを可能にする。

【0177】

【0173】 より具体的には、再エンコーダ900bは、データを1つ以上のビットストリーム901からダウンロードし、各ビットストリーム901は、1つ以上のサブアトム303（図3A及び図3Bを参照されたい）を含む。ダウンロードは、ダウンロードを管理するダウンロード902によって実行され（例えば、バッファを用いる）、ダウンロードの管理は、ソースを発見及び選択すること、接続を確立すること、接続をリスニングすること、及び接続を終端すること、並びに、各メディア・ソースからダウンロードされるデータの速度と量を制御することを含むが、これらに限定されない。

10

【0178】

【0174】 一部の実施形態において、サブアトム・パーサ903は、ビットストリーム901を解析して、CHUNKED_SUBATOMサブアトム311を抽出して一時的にキャッシュし、これらはチャンク・サブアトム併合部904によりマージされる。マージされるCHUNKED_SUBATOMサブアトム311は、次いで、セキュリティ・シーリングがメディア資産に対してイネーブルにされている場合に、セキュリティ検査部905によって検証される。

【0179】

【0175】 検証されたサブアトムは、次いで、デマルチプレクサ906によって分離され、メディア・メタデータは、SYNCサブアトム304とMEDIA_SEGMENT_INFOサブアトム306を再作成するために使用され、異なる線型コーデックを用いる線型コーディングされたデータは、線型再エンコーダ・モジュール913に供給される。線型再エンコーダ・モジュール913は、例えば、ビットストリーム・メタデータ内のスイッチ信号に基づいて、特定の再エンコーダ・タイプを選択するために、再エンコーダ・スイッチ914を起動する。

20

【0180】

【0176】 新しい線型コーディングされたデータの生成は、様々な方法で実施することが可能である。例えば、再エンコーダ900bは、先ず、線型復号化を適用して元のシンボルを復号化し、次いで、元のシンボルの新しい線型結合を、新しい線型コーディングされたシンボルとして作成することが可能である。別の例として、再エンコーダ900bは、線型コーディングされたシンボルを直接的に線型結合して、新しい線型コーディングされたシンボルを作成することができる。

30

【0181】

【0177】 一部の実施形態において、線型符号化情報は、線型符号化データとともに、BITSTREAM_HEADERサブアトム306、ENCODER_CONTENT_INFOサブアトム307、BLOCK_HEADERサブアトム308、PACKETサブアトム309、及びPACKET_HEADER_ONLYサブアトム312において伝送され、各パケット・サブアトム309は、1つの線型コーディングされたシンボルに対応する。セキュリティ・シーラー909は、次いで、ファイル、データ・ブロック、及びパケットに関するハッシュ化情報を、対応するENCODER_CONTENT_INFOサブアトム307、BLOCK_HEADERサブアトム308、及びPACKETサブアトム309に追加することによって、メディア・データのセキュリティを封印する（例えば、電子署名する）。

40

【0182】

【0178】 一部の実施形態において、各サブアトムは、サブアトム・トランケータ910によって更に打ち切れ、複数のCHUNKED_SUBATOMサブアトムによって運ばれて、サブアトムのサイズを小さくすることができる。これは、トランスポート・プロトコルがデータ・ユニット・サイズに制限を有する場合に特に有用である。次いで、サブアトムは、マルチプレクサ911によって多重化され、1つ以上のビットストリーム912の出力を形成する。異なるビットストリーム912は、異なるサブアトムを運ぶことが可能であり、別様に処理される（例えば、キャッシュされる、配信される）ことが可能である。受信

50

側が各ビットストリーム901内のサブアトムをナビゲートすることを支援するために、マルチプレクサ911は、BLOCK_DIRECTORYサブアトムを生成し、各ビットストリーム901に挿入することが可能である。

【0179】 規格遵守のためのビットストリーム・マッピングの例

図10は、一部の実施形態による他のコーディング・ビットストリーム規格に準拠するためのビットストリーム・マッピング・プロセス1000を示す。開示されるビットストリーム301とその実装はまた、様々な線型コーディング技術を使用して生成された線型符号化データを運ぶためのコンテナフォーマットとして、たとえそれらが以前に規定又は標準化されていたとしても、機能することが可能である。プロセス1000は、ビットストリーム・マッパー1003及びビットストリーム・デマッパー1007を使用することによって、既存の線型コードをサポートする。開示されるマッピング・プロセス1000は、開示されるビットストリームが、既存のメディア製品において標準化及び実装されている既存の線型コーディング方法に準拠して残ることを可能にする。

10

【0183】

【0180】 図10を参照すると、保護されるべきメディア・データは、先ず、考慮される線型コーディング技術の既存の標準化された線型エンコーダ1002を使用して符号化される。出力は、この技術/方法の標準化されたビットストリームである。次いで、ビットストリーム・マッパー1003は、このビットストリームのフィールドを抽出し、それらを、図3A、図3Bを参照して説明したフォーマットを有する開示されるビットストリーム301にマッピングする。ビットストリーム・マッパー1003はまた、アプリケーション層1001からリッチ・メディア・メタデータ (rich media metadata) を抽出して、開示されるビットストリーム301内のメタデータ関連フィールドを充填する。次いで、ビットストリームは、通信ネットワーク1004を介して配信する準備を整える。

20

【0184】

【0181】 受信側では、既存の標準化されたデコーダが使用されるかどうかに応じて、2つのオプションが存在する。第1に、既存の標準化されたデコーダが使用される場合1005、ビットストリーム・デマッパ1007は、コーディング関連データを、標準化線型デコーダ1008に入力される既存の標準ビットストリームにマッピングし、標準化線型デコーダはメディアデータをアプリケーション層1009に出力する。一方、ビットストリーム・デマッパ1007はまた、開示されるビットストリーム301からリッチ・メディア・メタデータを抽出し、それを受信機の実用アプリケーション層1009に渡す。既存の標準デコーダが使用されることを必要としない場合1005、又は既存の標準デコーダが存在しない場合、受信機はビットストリーム・デコーダ1006を使用して、受信機の実用アプリケーション層1009のためにメディア・データ及びメタデータを復号化する。

30

【0185】

【0182】 図11は、一部の実施形態による開示されるビットストリーム及びビット・ジェネレータを使用して生成される、メディア・ファイルの2つのFEコーディングされたコピーの実施例を示す。各々のコーディングされたコピーは、SYNCサブアトム304、BITSTREAM_HEADERサブアトム305、MEDIA_SEGMENT_INFOサブアトム306、ENCODER_CONTENT_INFOサブアトム307、BLOCK_HEADERサブアトム308、PACKETサブアトム309、BLOCK_DIRECTORYサブアトム310及びPACKET_HEADER_ONLY 312サブアトムのうちの全部又は一部を含む、前述の開示されたサブアトムのインスタンスを含む。

40

【0186】

【0183】 特に、異なるコピーの中のBLOCK_HEADERサブアトム310とPACKETサブアトム309は、使用される全てのメディア・ソースにわたってPACKETサブアトム309間で最大の線型独立性 (maximum linear independency) を提供する方法で生成される。これは、開示されるビットストリーム生成器を、異なるメディア・ソースに対して一意に構成することによって達成される。線型独立性をもたらすことが可能な構成の例は、異なるビットストリーム生成器において、異なるランダム・シードを適用すること、

50

又は異なるコーディング係数ベクトルを使用することを含む。このように、生成された任意のコピーからの任意のコーディングされたメディアPACKETサブアトム309は、高い確率で復元するために有用なものになり、従って、FEコピーとなる（以下、「FEコピー（単数又は複数）」とも言及される。）。FEコピーが、メディア・ストリーム/ファイル復元を可能にする十分に線型独立なPACKETサブアトム309を運んでいる場合、このコピーは、完全FEコード化コピー（full FE coded copy）と呼ばれる。そうでなければ、部分的FEコード化コピー（partial FE coded copy）と呼ばれる。

【0187】

【0184】 図12は、一部の実施形態による開示されるビットストリーム・フォーマットによって可能になる分散型マルチソース、マルチ・パス配信システム1200を示す。システム1200は：

- 1) ソースコーディングされた元のメディア・ファイル；
- 2) キャッシュの幾つかのクラスタ1202, 1203, 1204であって、各クラスタは、データ・ファイルを格納、処理（例えば、再符号化及び復号化のようなもの）、及び提供することが可能な接続されたキャッシュを含む（クラスタの幾つかの具体例は：CDN, 一組のエッジ演算ノード, キャッシング機能を有するISPノード, 一組のピア・ユーザー, 異なるタイプのキャッシュのハイブリッドを含むが、これらに限定されない）；及び
- 3) エンド・ユーザー1205であって、元のメディア・ファイルを所望して消費する者；を含む。

【0188】

【0185】 システム1200の既存の構成では、元のサーバー1201は、複製のために元のメディア・ファイルを、接続されたキャッシュ1202, 1203, 1204全てに分配する。対照的に、開示されるビットストリーム301を使用する開示されたシステムでは、元のサーバー1201は、開示されるビットストリーム生成器（図4参照）を使用して、元のメディア・ファイルの異なるFEコーディングされたコピーを作成し、次いで、それぞれを異なる接続されたキャッシュ1202, 1203, 1204に分配する。

【0189】

【0186】 システム1200の既存の構成では、元のサーバー1201に接続されていないキャッシュは、隣接するものに接続して、元のメディア・ファイルの複製を作成する。対照的に、開示されるビットストリーム301を使用する開示されたシステムでは、そのようなキャッシュは、複数の隣接するもの（neighbors）に接続し、隣接するもののFEコピーを効率的に利用して、それ自信のFEコーディングされたコピーを、図9を参照しながら説明した開示されるビットストリーム再エンコーダを使用して作成する。また、新たに作成されるFEコピーは、完全な又は部分的なコピーである可能性がある。キャッシュ1202, 1203, 1204は、ピアとして貢献することを希望するエンド・ユーザー1205であるということさえ可能である。

【0190】

【0187】 システム1200は、分散されたメディアFE配信システムであり、この場合において、全てのキャッシュは、相違してはいるがそれでも同じオリジナル・メディア・ファイルのFEコーディングされたコピーを格納している。これは、効率的なマルチ・ソース・マルチ・パス・ダウンロードを可能にする：エンド・ユーザー1205は、異なる通信リンクを介して複数のキャッシュ1202, 1203, 1204から並列にダウンロードすることが可能であり、ダウンロードされたデータの総量が、（ヘッダー・オーバーヘッドなどに起因して）元のメディア・ファイルより僅かに多い限り、エンド・ユーザー1205は、全てのダウンロードを終了し、図4に示される開示されるビットストリーム・デコーダ404を使用して元のメディア・ファイルを復元することが可能である。このようにして、エンド・ユーザー1205は、ダウンロード・コストを増加させることなく、全ての通信リンクの統合された帯域幅及び安定性を享受することができる。

【0191】

【0188】 上述のダウンロード・プロセスは、既存の単一ソース、単一経路のダウンロ

ード・プロセスよりも優れていおり、この場合において、エンド・ユーザー1205は元のメディア・ファイルをダウンロードするためにキャッシュ1202, 1203, 1204のうちの何れかに接続するだけである。既存のダウンロード・プロセスのパフォーマンスは、キャッシュとエンド・ユーザー1205との間の単一リンクによって制限されており、キャッシュ又はリンク障害に対して脆弱である。開示されたダウンロード・プロセスは、既存の複製ベースのマルチ・ソース・マルチ・パス・ダウンロード・プロセスよりも優れており、なぜなら少なくとも既存の複製ベースのマルチ・ソース・マルチ・パス・ダウンロード・プロセスは、メディア・ソースからの複製データのダウンロードを回避するために、ダウンロード間で時間がかかるエラーが生じやすいスケジューリングを必要としているからである、ということに留意されたい。

10

【0192】

【0189】 図13は、一部の実施形態による低レイテンシ・ユニキャスト・ビットストリーム配信システム1300を示す。システム1300は、低レイテンシ・ユニキャスト用アプリケーションにおける開示されるビットストリーム301の使用例を示し、ここで、受信機は、損失のある通信チャネルを介するデータ・ストリームの低レイテンシで信頼性の高い配信を要求している。このユース・ケースでは、メディア・ストリームは、プリ・プロセッサ・ユニット1301によって、ビデオ・フレームのような短時間で受信機に送信されることが可能な小さなチャンクに前処理される。ビットストリーム生成器1302は、前処理されたビットストリームに適用され、開示されるビットストリーム301を生成する。次いで、SYNCサブアトム304, BITSTREAM_HEADERサブアトム305, MEDIA_SEGMENT_INFOサブアトム306のような、メタデータを搬送し且つ頻繁には変化しないサブアトム1303が、他のチャネルより高いレイテンシ及びより低い帯域幅を有する可能性がある信頼性の高い制御チャネルを介して受信機に配信される。

20

【0193】

【0190】 一方、BLOCK_HEADERサブアトム308及びCHUNKED_PACKET_SUBATOMサブアトム311のような線型コーディングされた関連サブアトム1304は、後処理のためのデータ・チャンクの低レイテンシ復号化を可能にするために、低レイテンシ損失データ・チャネルを介して受信機に送信される。オプションとして、ビットストリーム・デコーダ1305は、チャネル状態を観察し、フィードバック・チャネルを介してビットストリーム生成器1302にフィードバックを送信して、ビットストリーム生成器1302が十分な情報に基づく符号化判断を行うことを支援する。

30

【0194】

【0191】 図14Aは、一部の実施形態によるPACKETサブアトム309に焦点を当てた低遅延ユニキャスト符号化及び復号化のためのシステム1400のブロック図である。前処理されたメディア・ストリーム1401は、時間ティック (time tick) 単位でビットストリーム生成器1402に供給され、各ティックは、線型符号化されるべき1つ以上の等しい長さのデータ・シンボルを含む。図示の例では、ティック1, 2及び4はそれぞれ1つのデータ・シンボルを含んでいるが、ティック3は2つのデータシンボル (3.2, 3.1) を含む。ビットストリーム生成器1402は、各データ・シンボルに対して1つのPACKETサブアトム309を系統的PACKETサブアトム309として作成し、また、これらのデータ・シンボルに線型コーディングを適用することによって、線型コーディングされたパケットを作成する。受信機側のビットストリーム・デコーダ1403は、受信した系統的PACKETサブアトム309とコーディングされたパケットとを、他のサポートしているサブアトムとともに使用して、ティック・データを復号化し、そのデータを受信機のアプリケーション層に渡す。

40

【0195】

【0192】 図14Bは、一部の実施形態によるサイズ・マーカー1404を介した可変データ・サイズのオフバンド処理を示す。プリ・プロセッサ1401はティック単位でデータを生成し、各ティックのデータは異なるサイズを有する可能性がある。一部の実施形態において、サイズ・マーカー1404は、各ティック・データのサイズを測定し、サイズ情報を

50

ティック・データに付加することを、ビットストリーム生成器1405へ送る前に行う。このサイズ情報は、ビットストリーム生成器1405からのデータに含まれ、線型符号化される。図示の例では、P1, P2, 及びP3の3つのデータのティックが存在する。それらのサイズは、データの一部となるサイズ・マーカー1404によって付加される。

【0196】

【0193】 ビットストリーム生成器1405の内部では、ゼロ・パディング部1406は、線型符号化のためにそれらの長さを等しくするために、データをゼロで埋める。一部の実施形態において、その長さは、一緒に符号化されるティックの中で最大のティック・データである。一部の実施形態において、グローバル最大長 (global maximum length) が適用される。図示される例では、3つのティックが一緒に線型符号化されており、P1が最大であるので、シンボル長は、サイズが前に付加されたP1の長さに設定される。従って、P2とP3は、線型符号化の前にゼロでパディングされる。

10

【0197】

【0194】 ティック・データを系統的PACKETサブアトム309として伝送する場合に、パディングされたゼロが伝送される必要はない。これは、ビットストリーム・デコーダ1408が系統的PACKETサブアトム309を受信する場合に、PACKETサブアトム309内のシンボルを、以下で説明するパディング・ピーラー (padding peeler) 1409に直接的に渡すことができるからである。パディング・ピーラー1409は、サイズを解析し、剥離するゼロが無いことを実現する。パディング・ピーラー1409は、サイズ・プレフィックスを取り除き、ティック・データをアプリケーションに渡す。

20

【0198】

【0195】 図示される例では、系統的PACKETサブアトム309としてP3の伝送が成功していると仮定し、ビットストリーム・デコーダ1408は、ゼロ・パディング無しでサイズ情報が付加されたP3を受信し、それをパディング・ピーラー1407に渡す。PACKETサブアトム309でb_systematicフラグをTrueに設定することは、判定を行うために使用される。ビットストリーム・デコーダ1408は、符号化されたPACKETサブアトム309を受信した場合に、符号化されたシンボルのサイズを通じて、復号化を可能にするために受信した系統的シンボルに幾つのゼロがパディングされるべきかを知る。

【0199】

【0196】 系統的シンボルが復号化によって取得される場合、ビットストリーム・デコーダ1408は、幾つのゼロがパディングされていたか分からない。この場合、ビットストリーム・デコーダ1408は、ゼロ・パディングされたシンボルをパディング・ピーラー1409に渡し、パディング・ピーラー1409は、サイズ情報を解析し、それに応じて剥離処理を行う。図示されている例では、P2の系統的PACKETサブアトム309の伝送は失敗していると仮定されており、そして、復号化によってP2が取り出される。復号化されたP2はパディングされたゼロを有し、これは、サイズ情報を解析した後に、パディング・ピーラー1409によって除去される。

30

【0200】

【0197】 図15Aは、一部の実施形態による低レイテンシのユニキャスト個別線型エンコーダ・デコーダ対を示す。特に、図15Aは、線型符号化及び復号化に焦点を当てて、開示されるビットストリーム実装を使用して、複数のティックのデータ1501を保護する方法を示す。この方法では、各ティックの間に、1つの線型エンコーダ1502が生成されている。この線型エンコーダ1502は、このティック及び幾つかの古いティックからのデータを、ブロックとして取り扱い、このブロックに線型コーディングを適用する。受信側では、各線型エンコーダ1502は、対応するブロックのデータを復号化する線型デコーダ1503とペアにされ、復号化されたティック・データを、アプリケーションに直接的に渡すか、又は、ティック・データを並べ替えて重複を除去するバッファ1504に渡す。

40

【0201】

【0198】 上述の方法は、適応的な方法で実施することが可能である。例えば、一緒に符号化されるティック数は、ネットワーク品質、システム・レイテンシ・バジェット、及

50

びその他の適切な要因のような要因に基づいて、動的にすることが可能である。別の例として、エンコーダが生成されるべき時点のティックは、ティックごとに1つのエンコーダの代わりに、ネットワーク品質、及びシステム・レイテンシ・バジェットのような要因に基づいて動的に決定することも可能である。

【0202】

【0199】 図15Bは、一部の実施形態によるクロス生成デコーダを用いる低レイテンシ・ユニキャストを示す。特に、図15Bは、線型符号化及び復号化に焦点を当てて、開示されるビットストリーム301を使用して、複数のティックのデータ1501を保護する方法を示す。この方法では、各ティックの間に、1つの線型エンコーダ1502が生成されている。この線型エンコーダは、このティック及び幾つかの古いティックからのデータを、ブロックとして取り扱い、このブロックに線型コーディングを適用する。受信側では、クロス・ブロック線型デコーダ1505が実装されており、これは、1つ以上のブロックからPACKETサブアトム309を収集し、例えば、拡張された一群の線型方程式を解くことによって、それらを一緒に復号化する。クロス・ブロック・デコーダ1505はまた、ティック・データを並べ替え、重複を除去し、次いで、データを受信側のアプリケーション層に渡す。

10

【0203】

【0200】 一部の実施形態において、上述の方法は、適応的な方法で実施することが可能である。例えば、一緒に符号化されるティック数は、ネットワーク品質、システム・レイテンシ・バジェットのような要因に基づいて、動的にすることが可能である。別の例として、クロス・ブロック・デコーダ1505と一緒に復号化するブロックの数は、ネットワーク品質、及びシステム・レイテンシ・バジェットのような要因に基づいて動的に決定されることも可能である。

20

【0204】

【0201】 図16は、一部の実施形態によるデジタル著作権管理（DRM）のために、開示されるビットストリーム301を使用するシステム1600を示す。特に、図16は、提案されるビットストリームがどのようにしてシンプルなデジタル著作権保護を可能にするかを示す。線型符号化メディア・データを復号化又は再符号化するために、受信機1602は、コーディング係数、生成サイズ、パケット・サイズ、ランダム・シード等のような線型コーディング情報を取得する。そうでない場合、受信機1602はメディア・データを復元することはできない。従って、提案されたビットストリームの対応するサブアトム内のそのような情報を暗号化することによって、コンテンツ所有者1601は、メディアデータへの不正アクセスを防止することができる。これは、提案されたビットストリームを用いてシンプルな簡単なDRMシステムを可能にする。

30

【0205】

【0202】 一部の実施形態において、コンテンツを準備する場合に、コンテンツ所有者1601は、特定の暗号的暗号化方式（例えば、AES）を使用して、線型コーディング情報を暗号化する。次いで、暗号化されたビットストリームは、通信ネットワークを介して受信機1602へ配信される。次いで、コンテンツ所有者1601は、復号鍵を認証サーバー1603に格納する。認証サーバー1603は、そのIDに基づいて受信機1602を認証し、認証をパスした場合に、復号鍵を受信機1602へ送信する。次いで、受信機1602は、復号鍵を使用して線型コーディング情報を復号化し、その情報を使用してメディア・データを復号化又は再符号化する。

40

【0206】

【0203】 図17は、一部の実施形態によるデータ認証を行う分散型メディア配信システム1700のブロック図である。特に、図17は、ピアのような信頼できないキャッシング・ノードが存在する可能性がある分散型メディア配信システムにおいて、開示されるビットストリームがどのようにしてデータ認証を可能にするかを示す。

【0207】

【0204】 図示されている例示的な実施形態では、コンテンツ制作者（又はコンテンツ

50

・クリエータ) 1701は、セキュリティ・シーリング機能を有効にした状態で、開示されるビットストリームを使用して、メディア・データを複数のビットストリームに符号化する。例えば、ファイル完全性は、ENCODER_CONTENT_INFOサブアトム307に含まれ、ブロック完全性は、BLOCK_HEADERサブアトム308に含まれ、パケット完全性は、PACKETサブアトム309に含まれる。次いで、コンテンツ制作者1701は、ビットストリームを、キャッシング・ノード1704-1706のネットワーク1703に配信する。この例では3つのキャッシング・ノードが示されているが、多数のキャッシング・ノードがネットワーク1703に含められることが可能である。

【0208】

[0205] キャッシング・ノード1704-1706は、ビットストリームを変更せずに格納するか、又は、開示されるビットストリーム・フォーマットによって可能になる再符号化機能を介して新しいビットストリームを作成することが可能である。その中で、信頼できないキャッシング・ノード1705、1706は敵対的である可能性があり、破損したビットストリーム、特に正しく線型コーディングされていないPACKETサブアトム309を生成することが可能である。

【0209】

[0206] 受信機ノード1707（エンド・ユーザー又は新しいキャッシング・ノードであることが可能であるもの）は、先ず、正しいENCODER_CONTENT_INFOサブアトム307とBLOCK_HEADERサブアトム308をダウンロードしてファイル完全性情報とブロック完全性情報を取得することによって、信頼できないキャッシング・ノード1705、1706からダウンロードするPACKETサブアトム309を認証することができる。このような情報は、適切に生成された場合（例えば、準同型ハッシングを使用して）、信頼できないノードによって生成されたPACKETサブアトム309の完全性を検証するために使用することが可能である。

【0210】

[0207] 正しいENCODER_CONTENT_INFOサブアトム307とBLOCK_HEADERサブアトム308をダウンロードするために、受信機ノード1707は、認可された者によって処理されるもののような、信頼できるキャッシング・ノード1704を使用することが可能である。代替的に、ENCODER_CONTENT_INFOサブアトム307とBLOCK_HEADERサブアトム308は各メディア資産に対して静的であるので、コンテンツ・クリエータ1701は、これらの2つのサブアトムに暗号的に署名することが可能であり、その結果、受信機1707は、これら2つのサブアトムをどこからでもダウンロードすることができ、コンテンツ・クリエータの暗号署名によってそれらを検証することができる。

【0211】

[0208] 更に、受信機ノード1707は、ノードから受信したPACKETサブアトム309が破損していることを発見した場合に、例えば、フォレンジック詐欺証明 (forensic fraud proof) を提出することによって、この違反をシステム1700の規制マネージャ1702に報告することができる。次いで、規制マネージャ1702は、罰金、加入取り消し、及び、ネットワーク1703からのこのノードをブラックリスト化すること等のような手段によって、不正行為ノードにペナルティを課すことができる。

【0209】 プロセス例

図18は、一部の実施形態による非線型又は線型コーディングされたデータからビットストリームを生成するプロセス1800の流れ図である。プロセス1800は、例えば、図22に関連して開示される電子デバイス・アーキテクチャを使用して実施されることが可能である。

【0212】

[0210] プロセス1800は、第1のデバイスにおいて：メディア資産を表現するメディア・データを受信し (1801)；ビットストリーム識別データ、コンテンツ固有符号化データ、及びメディア・セグメント・データのうちの少なくとも1つを含む第1の複数のデータ要素を取得し (1802)；メディア・データの少なくとも一部分を、第1の線型コ

ーディング・プロセスに従って、メディア資産に対応するコーディングされたデータに符号化し(1803)；第1の線型コーディング・プロセスに関連する情報に基づいて、第1の複数のデータ要素とは異なる第2の複数のデータ要素を生成し(1804)；第1の複数のデータ要素と第2の複数のデータ要素とを、メディア資産を表現する1つ以上のコーディングされたビットストリームに結合し(1805)；1つ以上のコーディングされたビットストリームを1つ以上の第2のデバイスへ、1つ以上のネットワーク経路を用いて伝送するステップ(1806)を含む。

【0213】

【0211】 一部の実施形態において、プロセス1800は、1つ以上のデバイスにより実施され、1つ以上のデバイスは、1つ以上のサーバー：例えば、コンテンツ配信ネットワーク(CDN)サーバー、ポイント・オブ・プレゼンス(PoP)サーバー、1つ以上のアプリケーション・サーバー(例えば、電子ニュース収集アプリケーション、コンテンツ放送アプリケーション、ユニキャスト・アプリケーション等をホストするサーバー)、コンテンツ・キャッシュ、ピア・デバイス、ローカル・キャッシング・デバイス、エッジ・デバイス(例えば、5G/6Gエッジ・アプリケーション・サーバー)又はクライアント・デバイス(例えば、メディア消費デバイス、コンピュータ、モバイル・デバイス、ゲーミング・システム、ストリーミング・デバイス、オーディオ/ビデオ機器など)を含むがこれらに限定されない。

10

【0214】

【0212】 一部の実施形態において、メディア資産は、画像、ビデオ、ゲーム状態、ゲーム・コマンド・データなどを含むが、これらに限定されない。

20

【0215】

【0213】 一部の実施形態において、メディア・データは、ソース・コーディングされたデータ、ストリーミング・プロトコルに従ってパッケージングされたデータ(例えば、HLS(HTTP-like streaming)、DASH(dynamic adaptive streaming over HTTP)等)を含むが、これらに限定されない。一部の実施形態において、メディア・データは、メディア資産に関連する線型コーディングされたデータ又はコーディングされたビットストリームである(例えば、コーディングされたコンテンツ及び関連するメタデータを含むコーディングされたビットストリーム)。一部の実施形態において、メディア資産を表すメディア・データは、複数のネットワーク経路から受信される。

30

【0216】

【0214】 一部の実施形態において、メディア・セグメント・データは、それぞれのセグメント又はブロックにおけるメディア資産の特性を識別する(例えば、MEDIA_SEGMENT_INFO)。一部の実施形態において、第1の複数のデータ要素は、メディア資産に関連するメタデータ(例えば、メディア資産から導出される、対応する、一緒に受信される)を含む(SYNC及びMEDIA_SEGMENT_INFOサブアトム)。一部の実施形態において、第1の複数のデータ要素は、線型コーディングされたペイロード・データを含まない。

【0217】

【0215】 一部の実施形態において、第1の線型コーディング・プロセス(例えば、サポートされる線型コード・タイプに対応する線型コーディング・プロセス)による符号化は、メディア・データの少なくとも第1の部分を含むパケットに線型符号化技術を適用することを含む(例えば、メディア・データの少なくとも一部分に対応する複数のパケットの組み合わせであるコーディングされたパケットを生成することを含む)。

40

【0218】

【0216】 一部の実施形態において、第1の線型符号化プロセスによる符号化は、メディア・データの少なくとも一部分に線型コーディング技術を適用することなく実行される(例えば、プロセスは、メディア・データの少なくとも一部分に対応する複数のパケットを組み合わせることによって符号化パケットを作成することを含まない)。例えば、符号

50

化プロセスは、第1の線型コーディング・プロセスの適用を必要とすることなく、メディア・データの少なくとも一部分を、コーディングされたデータに変換するために、第1の線型コーディング・プロセスに関連するマッピング関数（例えば、1つ以上のマッピング・ルール又はテーブル）を適用することを含むことが可能である。

【0219】

【0217】 一部の実施形態において、第2の複数のデータ要素は、コーディングされたペイロード・データ及びメディア資産に関連するメタデータ（例えば、BITSTREAM_HEADERサブアトム305、ENCODER_CONTENT_INFOサブアトム307、BLOCK_HEADERサブアトム308、PACKETサブアトム309、又はPACKET_HEADER_ONLYサブアトム312及びそれらに関連するビットフィールドのうちの1つ以上）を含む。

10

【0220】

【0218】 一部の実施形態において、結合は、インターリーブ、多重化、又はその他の方法で、送信又は記憶のためのシーケンシャル信号又は表現に変換することを含む。

【0221】

【0219】 一部の実施形態において、1つ以上のネットワーク経路は、1つより多いネットワーク、1つより多いネットワーク・タイプ（例えば、LAN、VLAN、WAN、SAN、WLAN、VPN等）、1つより多いネットワーク標準（例えば、1G/2G/3G/4G/5G/6Gモバイル、Bluetooth、802.11ベースのWIFIなど）、1つより多いネットワーク伝送媒体（例えば、無線、有線、ファイバなど。）、異なるパフォーマンス特性を有する通信チャンネル（例えば、非ロスレス・プロトコル、信頼できるプロトコル、低遅延を利用するチャンネル）を含む。

20

【0222】

【0220】 図19は、一部の実施形態による復号化/再符号化のプロセス1900の流れ図である。プロセス1900は、例えば、図22に関連して開示される電子デバイス・アーキテクチャを使用して実装することが可能である。

【0223】

【0221】 プロセス1900は、第1のデバイスにおいて：メディア資産に関連する1つ以上のコーディングされたビットストリームを受信するステップ（1901）；1つ以上のコーディングされたビットストリームをバッファに格納するステップ（1902）；バッファに格納された1つ以上のコーディングされたビットストリームを処理するステップであって：ビットストリーム識別データ、コンテンツ固有符号化データ、及びメディア・セグメント・データのうちの少なくとも1つを含む第1の複数のデータ要素を、1つ以上のコーディングされたビットストリームから抽出し（1903）；1つ以上のコーディングされたビットストリームを生成するために使用されるそれぞれのコーディング・プロセスに関連する、1つ以上のコーディングされたビットストリームからの情報を含む第2の複数のデータ要素を取り出すこと（1904）を含むステップ；及び第2の複数のデータ要素に基づいてメディア資産を表現する復号化されたデータを生成すること、及び、第2の複数のデータ要素に基づいてメディア資産を表現する新たにコーディングされたデータを生成すること、のうちの一方を実行するステップ（1905）を含む。

30

【0224】

【0222】 一部の実施形態において、第1のデバイスは、サーバー（例えば、コンテンツ配信ネットワーク（CDN）サーバー、ポイント・オブ・プレゼンス（PoP）サーバー、アプリケーション・サーバー（例えば、電子ニュース収集アプリケーション、コンテンツ放送アプリケーション、ユニキャスト・アプリケーション等をホストするサーバー）、コンテンツ・キャッシュ、ピア・デバイス、ローカル・キャッシング・デバイス、エッジ・デバイス（例えば、5G/6Gエッジ・アプリケーション・サーバー）又はクライアント・デバイス（例えば、メディア消費デバイス、コンピュータ、モバイル・デバイス、ゲーミング・システム、ストリーミング・デバイス、オーディオ/ビデオ機器など）であってもよい。

40

【0225】

50

【0223】 一部の実施形態において、1つ以上のコーディングされたビットストリームは、少なくとも2つの異なる線型コーディング・プロトコル/標準のうちの1つに従って符号化されたビットストリームを含む。

【0226】

【0224】 一部の実施形態において、処理は、逆多重化及び/又は解析を含む

【0225】 一部の実施形態において、メディア・セグメント・データは、それぞれのセグメント又はブロックにおけるメディア資産の特性を識別するために使用される（例えば、MEDIA_SEGMENT_INFOサブアトム306）。

【0227】

【0226】 一部の実施形態において、第1の複数のデータ要素は、メディア資産に関連する（例えば、メディア資産から導出される、対応する、一緒に受信される）メタデータを含む（例えば、SYNCサブアトム304及びMEDIA_SEGMENT_INFOサブアトム306）

【0227】 一部の実施形態において、第1の複数のデータ要素は、線型コーディングされたペイロード・データを含まない。

【0228】

【0228】 一部の実施形態において、第2の複数のデータ要素は、コーディングされたペイロード・データ及びメディア資産に関連するメタデータ（例えば、BITSTREAM_HEADERサブアトム305、ENCODER_CONTENT_INFOサブアトム307、BLOCK_HEADERサブアトム308、PACKETサブアトム309、又はPACKET_HEADER_ONLYサブアトム312のうちの1つ以上）を含む。

【0229】

【0229】 一部の実施形態において、コーディングされたデータは、少なくとも2つの異なる線型コーディング・プロトコル/標準のうちの異なるものに対応する。

【0230】

【0230】 図20は、一部の実施形態によるダウンストリーム互換性に関する新たなコード・タイプを変換するプロセス2000の流れ図である。プロセス2000は、例えば、図22に関連して開示される電子デバイス・アーキテクチャを使用して実装することが可能である。

【0231】

【0231】 プロセス2000は、第1の電子デバイスにおいて：第1のタイプの第1のコーディングされたビットストリームを取得するステップ（2001）；第1のタイプとは異なる第2のタイプの第2のコーディングされたビットストリームを生成するステップ（2002）；及び第2のコーディングされたビットストリームを1つ以上の第2の電子デバイスへ、1つ以上のネットワーク経路を介して伝送するステップ（2002）を含む。

【0232】

【0232】 一部の実施形態において、第1のタイプの第1のビットストリームは、第1のタイプの線型コーディング・プロトコル又は標準（例えば、RaptorQTM）に従って生成される。

【0233】

【0233】 一部の実施形態において、第1又は第2のビットストリームは、データ・セグメントに関連する系統的パケットと、データ・セグメントに関連する非系統的パケットとを含む。

【0234】

【0234】 一部の実施形態において、第1又は第2のビットストリームは、データ・セグメントに関連する系統的パケットを含み、データ・セグメントに関連する非系統的パケットを含まない。

【0235】

【0235】 一部の実施形態において、第1又は第2のビットストリームは、データ・セグメントに関連する非系統的パケットを含み、データ・セグメントに関連する系統的パケットを含まない。

10

20

30

40

50

【0236】

【0236】 一部の実施形態において、第2のタイプの第2のビットストリームは、第2のタイプの線型コーディング・プロトコル又は標準（例えば、リード・ソロモン・コード）に従って生成される。

【0237】

【0237】 一部の実施形態において、1つ以上のネットワーク経路は、1つより多いネットワーク、1つより多いネットワーク・タイプ（例えば、LAN，VLAN，WAN，SAN，WLAN，VPNなど）、1つより多いネットワーク標準（例えば、1G/2G/3G/4G/5G/6Gモバイル、Bluetooth、802.11ベースのWIFIなど）、1つより多いネットワーク伝送媒体（例えば、無線、有線、ファイバなど）、異なるパフォーマンス特性を有する通信チャンネル（例えば、非ロスレス・プロトコル、信頼できるプロトコル、低遅延プロトコル）などを含む。

10

【0238】

【0238】 図21は、一部の実施形態によるダウンストリーム互換性に関する新たなコード・タイプを変換する別のプロセス2100の流れ図である。プロセス2100は、例えば、図22に関連して開示される電子デバイス・アーキテクチャを使用して実装することが可能である。

【0239】

【0239】 プロセス2100は、第1の電子デバイスにおいて：第1のタイプの線型コーディングを使用して符号化されたメディア資産に関連付けられた第1のビットストリームを受信するステップ（2101）；第1のビットストリームが第2のタイプの線型コーディングに関連付けられたデコーダで復号化されるべきである旨の判定に従って、第1のビットストリームを、第2のタイプの線型コーディングの第2のビットストリームに処理するステップであって：第1のビットストリームの一群のフィールドからデータを取り出し、そのデータを、第2のビットストリームの第2の対応する一群のフィールドにマッピングし（2102）；メディア資産に関連するメタデータを、第1のビットストリームから抽出し（2103）；第2のビットストリームを、メディア資産のコーディングされていない表現に、第2のタイプのコーディングに関連付けられたデコーダを用いて復号化することを含むステップ（2104）；第1のビットストリームが第2のタイプのコーディングに関連付けられたデコーダで復号化されるべきではない旨の判定に従って、第1のビットストリームを処理するステップであって：メディア資産に関連付けられたメタデータ、及び第1のタイプのコーディングに関連付けられたデコーダを使用して、メディア資産のコーディングされていない表現に第1のビットストリームを復号化し（2105）；メディア資産のコーディングされていない表現を、メディア資産に関連する取り出されたメタデータとともにアプリケーションに配分すること（2106）を含むステップを含む。

20

30

【0240】

【0240】

上記で開示されたプロセス1800、1900、2000及び2100はそれぞれ、線型コーディング・プロセスに関連するデータ要素を、伝送前に、ビットストリーム識別データ、コンテンツ固有符号化データ、及びメディア・セグメント・データのうちの少なくとも1つを含むデータ要素と組み合わせることを可能にし、異なるパフォーマンス特性を有するネットワークを介して、様々な処理能力（例えば、符号化/復号化の能力）を有するデバイスへ/からのメディア・データのより効率的な配信を可能にする。組み合わせられたデータは、より効率的な復号化をもたらし、互換性を保証するために複数の冗長タイプのビットストリームを生成して伝送する必要性を低減し、それによってネットワーク及び処理リソースを節約する。

40

【0241】 システム・アーキテクチャの例

図22は、本開示の例示的な実施形態を実施するのに適した例示的な電子デバイス・アーキテクチャ2200のブロック図を示す。アーキテクチャ2200は、図1-21を参照して前述したように、サーバー及びクライアント・デバイスを含むが、これらに限定されない

50

。

【0241】

【0242】 図示されるように、アーキテクチャ2200は、例えばリード・オンリー・メモリ（ROM）2202に格納されたプログラム、又は、例えばストレージ・ユニット2208からランダム・アクセス・メモリ（RAM）2203にロードされたプログラムに従って、様々な処理を実行することが可能な中央処理ユニット（CPU）2201を含む。RAM 2203には、CPU 2201が様々な処理を実行する際に必要とされるデータも、必要に応じて格納されている。CPU 2201、ROM 2202、及びRAM 2203は、バス2204を介して互いに接続される。また、入力／出力（I/O）インターフェース2205もバス2204に接続されている。

10

【0242】

【0243】 以下の構成要素が入力／出力インターフェース2205に接続される：入力ユニット2206であって、キーボード、マウス等を含む可能性のあるもの；出力ユニット2207であって、液晶ディスプレイ（LCD）のようなディスプレイ及び1つ以上のスピーカを含むもの；ストレージ・ユニット2208であって、ハードディスク又は他の適切なストレージ・デバイスを含むもの；及び通信ユニット2209であって、ネットワーク・カード（例えば、有線又は無線）のようなネットワーク・インターフェース・カードを含むもの。

【0243】

【0244】 一部の実施形態において、入力ユニット2206は、様々なフォーマット（例えば、モノラル、ステレオ、空間、イマーシブ、及びその他の適切なフォーマット）でのオーディオ信号のキャプチャを可能にする、（ホスト・デバイスに依存して）異なる位置にある1つ以上のマイクロフォンを含む。

20

【0244】

【0245】 一部の実施形態において、出力ユニット2207は、様々な数のスピーカを有するシステムを含む。出力ユニット2207は、（ホスト・デバイスの能力に依存して）様々なフォーマット（例えば、モノラル、ステレオ、イマーシブ、バイノーラル、及びその他の適切なフォーマット）でオーディオ信号をレンダリングすることができる。

【0245】

【0246】 一部の実施形態において、通信ユニット2209は、他のデバイスと（例えば、ネットワークを介して）通信するように構成される。ドライブ2210は、必要に応じてI/Oインターフェース2205にも接続される。磁気ディスク、光ディスク、光磁気ディスク、フラッシュ・ドライブ、又はその他の適切な取り外し可能な媒体のような、取り外し可能な媒体2211がドライブ2210に取り付けられており、その結果、そこから読み込まれるコンピュータ・プログラムが必要に応じてストレージ・ユニット2208にインストールされる。システム2200が上述の構成要素を含むものとして説明されているが、実際のアプリケーションでは、これらの構成要素の幾つかを追加、除去、及び/又は置換することが可能であり、これらの変形又は変更は全て本開示の範囲内にある、ということを当業者は理解するであろう。

30

【0246】

【0247】 本開示の例示的な実施形態によれば、上述のプロセスは、コンピュータ・ソフトウェア・プログラムとして、又は、コンピュータ読み取り可能な記憶媒体において実施されてもよい。例えば、本開示の実施形態は、機械読み取り可能な媒体において有形的に具現化されたコンピュータ・プログラムを含むコンピュータ・プログラム製品を含み、コンピュータ・プログラムは、方法を実行するためのプログラム・コードを含む。このような実施形態において、コンピュータ・プログラムは、図22に示されるように、通信ユニット2209を介してネットワークからダウンロードされ搭載され、及び/又は、取り外し可能な媒体2211からインストールされることが可能である。

40

【0247】

【0248】 一般に、本開示の様々な例示的な実施形態は、ハードウェア又は特殊目的回

50

路（例えば、制御回路）、ソフトウェア、ロジック、又はそれらの任意の組み合わせで実施されることが可能である。例えば、上述のユニットは、制御回路（例えば、図22の他の構成要素との組み合わせにおけるCPU 2201）によって実行されることが可能であり、従って制御回路は本開示で説明される動作を実行することが可能である。一部の態様はハードウェアで実装される可能性があり、他の態様は、コントローラ、マイクロプロセッサ、又はその他の演算デバイス（例えば、制御回路）によって実行される可能性があるファームウェア又はソフトウェアで実装される可能性がある。本開示の例示的な実施形態の様々な態様は、ブロック図、フローチャート、又は他の何らかの図式的表現を使用して図示及び説明されているが、本件で説明されるブロック、装置、システム、技術又は方法は、非限定的な例として、ハードウェア、ソフトウェア、ファームウェア、特殊目的回路又はロジック、汎用ハードウェア又はコントローラ又はその他の演算デバイス、又は、それらの何らかの組み合わせで実施されてもよい、ということが理解されるであろう。

10

【0248】

【0249】 更に、フローチャートに示される様々なブロックは、方法ステップとして、及び/又はコンピュータ・プログラム・コードの動作の結果から生じる動作として、及び/又は関連する機能を実行するように構成された複数の結合された論理回路要素として見ることが可能である。例えば、本開示の実施形態は、機械読み取り可能な媒体において有形的に具現化されたコンピュータ・プログラムを含むコンピュータ・プログラム製品を含み、コンピュータ・プログラムは、上述の方法を実行するように構成されたプログラム・コードを含む。

20

【0249】

【0250】 本開示の文脈において、機械読み取り可能な媒体は、命令実行システム、装置、又はデバイスによって、又はそれらに関連して使用されるプログラムを含むか、又は格納することが可能な何らかの有形媒体である可能性がある。機械読み取り可能な媒体は、機械読み取り可能な信号媒体又は機械読み取り可能な記憶媒体であってもよい。機械読み取り可能な媒体は、非一時的であることが可能であり、電子的、磁氣的、光学的、電磁的、赤外線的、又は半導体的なシステム、装置、又はデバイス、又は前述の任意の適切な組み合わせを含むことが可能であるが、これらに限定されない。機械読み取り可能な記憶媒体のより具体的な例は、1つ以上のワイヤを有する電気接続、ポータブル・コンピュータ・ディスク、ハード・ディスク、ランダム・アクセス・メモリ（RAM）、リード・オンリー・メモリ（ROM）、消去可能なプログラマブル・リード・オンリー・メモリ（EPROM又はフラッシュ・メモリ）、光ファイバー、ポータブル・コンパクト・ディスク・リード・オンリー・メモリ（CD-ROM）、光ストレージ・デバイス、磁気ストレージ・デバイス、又は、これらの任意の適切な組み合わせを含む。

30

【0250】

【0251】 本開示の方法を実行するためのコンピュータ・プログラム・コードは、1つ以上のプログラミング言語の任意の組み合わせで書かれている可能性がある。これらのコンピュータ・プログラム・コードは、制御回路を有する汎用コンピュータ、特殊目的コンピュータ、又はその他のプログラマブル・データ処理装置のプロセッサに提供される可能性があり、その結果、プログラム・コードは、コンピュータ又はその他のプログラマブル・データ処理装置のプロセッサにより実行されると、フローチャート及び/又はブロック図で規定されている機能/動作を実行させる。プログラム・コードは、コンピュータ上で完全に、コンピュータ上で部分的に、スタンド・アロンのソフトウェア・パッケージとして動作することが可能であり、部分的にコンピュータ上で且つ部分的にリモート・コンピュータ上で、又は、完全にリモート・コンピュータ又はサーバー上で、又は、1つ以上のリモート・コンピュータ及び/又はサーバー上に分散されて動作することが可能である。

40

【0251】

【0252】 その他の実施形態

一部の実施形態において、第1の線型コーディング・プロセス（例えば、サポートされる線型コード・タイプに対応する線型コーディング・プロセス）に従う符号化は、メディ

50

ア・データの少なくとも一部分を含むパケットに、線型符号化技術を適用することを含む（例えば、メディア・データの少なくとも一部分に対応する複数のパケットの組み合わせであるコーディングされたパケットを生成することを含む）。

【0252】

【0253】 一部の実施形態において、第1の線型コーディング・プロセスによる符号化は、メディア・データの少なくとも一部分に線型コーディング技術を適用することなく実行される（例えば、プロセスは、メディア・データの少なくとも一部分に対応する複数のパケットを組み合わせることによって符号化パケットを作成することを含まない）。例えば、符号化プロセスは、第1の線型コーディング・プロセスの適用を必要とすることなく、メディア・データの少なくとも一部分を、コーディングされたデータに変換するために、第1の線型コーディング・プロセスに関連するマッピング関数（例えば、1つ以上のマッピング・ルール又はテーブル）を適用することを含むことが可能である。

10

【0253】

【0254】 開示される実施形態の利点

開示されるビットストリームは、以下の利点をもたらす：

1. 開示される実施形態は、ブロック・コードを含むがこれに限定されない任意の線型コードを含む、同じビットストリーム内の複数の線型コーディング技術の使用をサポートする。

【0254】

2. 開示される実施形態は、マルチ・ソース・マルチ・パス・メディア・データ配信システムを介してメディア資産を符号化、配信、再符号化、キャッシュし、配信ポイントで復号化されることを可能にするメタデータを含む。

20

【0255】

3. 開示されるビットストリーム301（サブアトム及びメタデータ）は、セントラリ化された制御を必要とせず、ソース間の調整を必要とせずに、マルチ・ソース・マルチ・パス配信を可能にし、配信ポイントでの最小限の冗長性を可能にする。

【0256】

4. 開示される実施形態は、サブアトム・システムが異なる方法を使用して異なるタイプのデータを送信することを可能にし、これは、異なるサブアトムの柔軟性のあるヘテロジニアス配信を可能にする。例えば、メタデータのような一部のサブアトムは、信頼できるチャネルを経由し、メディア・エッセンスのような他のサブアトムは、信頼できないチャネルを経由することができる。一部のサブアトムは暗号化することが可能であり、一部のサブアトムは暗号化しないことが可能である。

30

【0257】

5. 開示される実施形態は、CHUNKED_SUBATOMサブアトムの使用を通じて、通信プロトコル・データ・ユニット・サイズに適合する柔軟なビットストリーム・サブアトム・サイズを可能にする。

【0258】

6. 開示される実施形態は、メディア資産をより小さなブロックにチャンキングし（チャンクに分け）、次いで、BLOCK_HEADERサブアトムのblock_indexフィールドを使用して異なるブロックを示すことによって、異なるブロックに線型コードを別々に適用することを可能にする。

40

【0259】

7. 開示される実施形態は、サブアトム構造を変更することなく、特定のサブアトム内で、符号化されたデータの線型再符号化を可能にする（例えば、出力は同じサブアトム構造であり、変更なしにデコーダによって消費されることが可能である）。

【0260】

8. 開示される実施形態は、サブアトム構造を変更することなく、異なるデバイスのストレージ制約を満たすために、様々なサイズの機能的に等価なコーディングされたコピーを作成することを可能にする。

50

【0261】

9. 開示される実施形態は、複数の異なる線型コードを使用してメディア資産を符号化し、複数のソース（例えば、CDN）を介して配信する能力をサポートし、この複数のビットストリームが復号化の場所で利用可能である場合、それらは同じビットストリーム・デコーダを使用して復号化されることが可能である。

【0262】

10. 開示される実施形態は、複数の異なる線型コードを使用して符号化され且つ複数のソース（例えば、CDN）を介して配信されるメディア資産を、線型再符号化する能力をサポートし、この複数のビットストリームが再符号化の場所で利用可能である場合、それらは同じビットストリーム・コーデックを使用して再符号化される。

10

【0263】

11. 開示される実施形態は、既存の標準化された線型コード・フォーマット（例えば、RaptorQ, R-S）を使用してビットストリームにエンコードされたメディア資産の変換を可能にし、次いで、この標準化されたビットストリームは、マッピング機能により、提案されるビットストリームにマッピングされることが可能である。デマッピング機能は、提案されるビットストリームのデコーダにおいて適用され、元の標準化されたビットストリームを、標準化されたフォーマットに復元する。

【0264】

12. 開示される実施形態は、スライディング・ウィンドウ又はブロック・コードである線型コードの選択及び使用をサポートする。このフォーマットは、エンコーダでスライディング・ウィンドウ・コードが使用される場合に、デコーダがブロック又はスライディング・ウィンドウの何れかをを用いて復号化することを可能にする。

20

【0265】

13. 開示される実施形態は、メディア資産が信頼できないチャネルを介して送信される一方で、信頼できるチャネルを介して送信されることが可能な適切な線型コード（例えば、スライディング・ウィンドウ・コードのサポート）及びサブアトムを選択を介して、低遅延の非ロスレス・メディア・データ・ユニキャスト・システムの配信をサポートする。

【0266】

14. 開示される実施形態は、ビットストリーム内の選択されたサブアトム及びフィールドの暗号化を可能にし、認証されていないデコーダがメディア資産（例えば、DRM）を復元することを防止する。

30

【0267】

15. 開示される実施形態は、認証情報を搬送するフィールドを用いて、異なるレベルのデータ完全性チェック（例えば、ファイル・レベル、ブロック・レベル、及びパケット・レベルのチェック）をサポートする。特に、それは、準同型ハッシュ演算のような特別なハッシュ演算アルゴリズムの使用をサポートし、復号化を行うことなく線型コーディングされたパケットの完全性チェックを可能にする。

【0268】

【0255】 更に、開示されるビットストリーム生成器は、以下の利点をもたらす。

40

【0269】

1. 開示されるビットストリーム生成器は、通信プロトコル・データ・ユニット・サイズに適合するように、サブアトムをより小さなピース（CHUNKED_SUBATOMサブアトム311）にチャンク化することができる。

【0270】

2. 開示されるビットストリーム生成器は、メディア資産をより小さなブロックにチャンク化し、線型コードを様々なブロックに別々に適用し、BLOCK_HEADERサブアトムのblock_indexフィールドを使用して様々なブロックを指定することができる。

【0271】

3. 開示されるビットストリーム生成器は、サブアトム・マルチプレクサを使用して

50

複数の異なるビットストリームを生成することが可能であり、異なるビットストリームを異なるネットワークエンティティに配信することができる。

【0272】

4. 開示されるビットストリーム生成器は、チャネル情報を受信し、それを使用して符号化方式を適応させることができる。

【0273】

5. 開示されるビットストリーム生成器は、メディアをビットストリームに符号化するための複数の線型符号化技術の中から或る線型符号化技術を選択することを、フィールド `linear_code_type` を使用してサブアトム `BITSTREAM_HEADER` でこれをシグナリングすることによって行う。

【0274】

6. 開示されるビットストリーム生成器は、異なるバリエーションにおいて、同じメディア情報（例えば、同じ `SYNC`、`BITSTREAM_HEADER`、`ENCODER_CONTENT_INFO`、`MEDIA_SEG_INFO` サブアトム）を配置してこのメディア資産を一意的に識別する一方、異なる線型コード情報（例えば、`BLOCK_HEADER` サブアトムにおける異なる `prng_seed`、`PACKET` サブアトムにおける異なる `poly_index` 及び異なる係数）を配置してそのバリエーションを区別することによって、同じメディア資産の複数の線型コーディングされたバリエーション（`linear coded variants`）を生成することができる。このようにして、バリエーションを、マルチ・ソースデータ配信システム上で配信することができ、全てがメディア資産の復号化に貢献することができる。

【0275】

7. 開示されるビットストリーム生成器は、`BITSTREAM_HEADER` サブアトムの `linear_code_type` フィールドを様々な値に設定したり、`BITSTREAM_HEADER` サブアトムの `content_source_size` フィールドをスライディング・ウィンドウ・コード用に0に設定したり、及び/又は、スライディング・ウィンドウ・コード用に `PACKET` サブアトムの動的な `window_start_index` 及び `window_end_index` を使用したりすること等のように様々なフィールドを介して、ブロック・コード又はスライディング・ウィンドウ・コードが使用されるかどうかをシグナリングすることができる。

【0276】

8. 開示されるビットストリーム生成器は、受信機フィードバックに基づいて線型コーディング手法及びパラメータを動的に変更することができる。

【0277】

9. 開示されるビットストリーム生成器は、（`PACKET` サブアトムの `coefficient_vector` フィールドのような）コーディング情報を暗号化し、（例えば、`PACKET_HEADER_ONLY` サブアトムを使用して）コーディング情報を個別に伝送して、メディア資産の不正な復号化を防止することができる。

【0278】

10. 開示されるビットストリーム生成器は、データ完全性チェックを可能にするために、様々なハッシュ手法（`SHA2`、`SHA256`、準同型ハッシュ演算など）を使用して、様々な方法でメディア資産をハッシュ演算する（ファイルをハッシュ演算する、ブロックをハッシュ演算する、元のシンボルをハッシュ演算する、等々を行う）ことができる。特に、準同型ハッシュ演算のような特殊なハッシュ演算アルゴリズムを使用して、復号化することなく、線型コーディングされたパケットの完全性チェックを可能にすることができる。

【0279】

【0256】 更に、開示されるビットストリーム再エンコーダの実施形態は、以下の利点をもたらす。

【0280】

1. 開示されるビットストリーム再エンコーダは、ビットストリーム再エンコーダを実施し、互いにコーディネートすることなく、機能的に等価な完全コード化コピー及び部

10

20

30

40

50

分的コード化コピーを生成するために、異なるネットワーク・エンティティで実施されることが可能である。

【0281】

2. 開示されるビットストリーム再エンコーダは、異なるバリエーションからの、同じメディア情報（例えば、同じSYNC、BITSTREAM_HEADER、ENCODER_CONTENT_INFO、及びMEDIA_SEG_INFOサブアトム）及び異なる線型コード情報（例えば、BLOCK_HEADERサブアトムにおける異なるprng_seed、PACKETサブアトムにおける異なるpoly_index及び異なる係数）を通じて、同じメディア資産の異なる線型コーディングされたバリエーションを用いて、識別及び再エンコードを行うことができる。

【0282】

3. 開示されるビットストリーム再エンコーダは、異なる線型コードを使用して生成された同じメディア資産の異なるビットストリームを統合及び再エンコードすることができる。

【0283】

4. 開示されるビットストリーム・エンコーダは、コーディング情報（例えば、PACKETサブアトムのcoefficient_vectorフィールド）を（再符号化のために）復号化及び再暗号化し、コーディング情報を別々に伝送して（例えば、PACKET_HEADER_ONLYサブアトムを使用する）、メディア資産の不正な復号化を防止することができる。

【0284】

5. 開示されるビットストリーム再エンコーダは、ハッシュ技術（SHA2、SHA256、準同型ハッシュ演算など）及びサブアトムで指定されたデータ完全性フィールドに従って、様々な方法（ファイル検証、ブロック検証、パケット検証など）でメディア資産の完全性を検証することができる。特に、準同型ハッシュ演算のような特殊なハッシュ技術が使用される場合、それは、復号化することなく、線型コーディングされたパケットを検証することができる。

【0285】

【0257】 更に、開示されるビットストリーム・デコーダの実施形態は、以下の特徴/利点をもたらす。

【0286】

1. 開示されるビットストリーム・デコーダは、同じメディア資産の複数の異なるビットストリームを識別してダウンロードし、それらを使用してメディア資産を復号化することが可能である。

【0287】

2. 開示されるビットストリーム・デコーダは、BLOCK_HEADERサブアトムのblock_indexフィールドをインジケータとして使用することにより、同じメディア資産の異なるブロックを別々に復号化し、それらを正しい順序で連結して元のメディア資産を復元することができる。

【0288】

3. 開示されるビットストリーム・デコーダは、CHUNKED_SUBATOMサブアトム内の関連するフィールド（例えば、original_subatom_id、chunk_size、chunked_segment_index、num_chunked_segments）をコネクタとして使用することにより、まだ復元することができないCHUNKED_SUBATOMサブアトムを一時的に格納するためにバッファを使用することにより、及び、復元を試みるために併合器を使用することにより、複数のCHUNKED_SUBATOMサブアトムを使用して完全なサブアトムを回復することができる。

【0289】

4. 開示されるビットストリーム・デコーダは、異なる線型コードを使用して生成されたビットストリームを使用して、メディア資産を統合及び復号化することができる。

【0290】

5. 開示されるビットストリーム・デコーダは、符号化方式及びレイテンシ要件に基

10

20

30

40

50

づいて、ブロック復号化及びスライディング・ウィンドウ復号化のような様々なタイプの復号化をサポートする。

【0291】

6. 開示されるビットストリーム・デコーダは、BITSTREAM_HEADERサブアトムのlinear_code_typeフィールド、BITSTREAM_HEADERサブアトムの内容sizeフィールド、及び/又は、PACKETサブアトムのwindow_start_index及びwindow_end_indexのような様々なフィールドを介して、ブロック・コード又はスライディング・ウィンドウ・コードが使用されるかどうかを識別することができる。

【0292】

7. 開示されるビットストリーム・デコーダは、BITSTREAM_HEADERサブアトム及びフィールドlinear_code_typeを読み取ることによって、複数の線型コーディング技術の中から、ビットストリームを適切に復号化するために線型コード・タイプを識別することができる。

【0293】

8. 開示されるビットストリーム・デコーダは、異なるバリエーション同士の間で、同じメディア情報（例えば、同じSYNC、BITSTREAM_HEADER、ENCODER_CONTENT_INFO、及びMEDIA_SEG_INFOサブアトム）及び異なる線型コード情報（例えば、BLOCK_HEADERサブアトムにおける異なるprng_seed、PACKETサブアトムにおける異なるpoly_index及び異なる係数）を通じて、同じメディア資産の異なる線型コーディングされたバリエーションを用いて、識別及び復号化を行うことができる。

【0294】

9. 開示されるビットストリーム・デコーダは、ウィンドウ・サイズと生成サイズの組み合わせを使用して、線型コードがスライディング・ウィンドウであるか又はブロック・コードであるかを識別する。

【0295】

10. 開示されるビットストリーム・デコーダは、受信したサブアトムに基づいてチャンネル情報を分析し、その情報を生成器に報告して十分な情報に基づいた符号化判定に備えることができる。

【0296】

11. 開示されるビットストリーム・デコーダは、符号化情報（例えば、PACKETサブアトムのcoefficient_vectorフィールド）を復号化して、メディア資産の復号化を可能にすることができる。

【0297】

12. 開示されるビットストリーム・デコーダは、サブアトムにおいて指定されたデータ完全性フィールド及びハッシュ技術（SHA2、SHA256、準同型ハッシュ演算など）に従って、様々な方法（ファイル検証、ブロック検証、パケット検証など）でメディア資産の完全性を検証することができる。特に、準同型ハッシュ演算のような特殊なハッシュ技術が使用される場合、復号化することなく、線型コーディングされたパケットを検証することができる。

【0298】

【0258】 更に、開示される実施形態は以下をもたらす：

1. 線型コード化の前にデータの予備的処理を使用して、可変サイズのデータ・パケットを処理するメカニズム。従って、ビットストリームはメディアが可変サイズのパケットにおけるものであることを知らない。

【0299】

2. 信頼できるが遅延に敏感ではない接続（TCP）を介して、時間に敏感ではないが復号化に重要な特定のサブアトムを送信し、信頼できない（且つ低遅延の）接続を介して、符号化されたメディア・ストリームを送信することによって、メディア・ストリームの低遅延配信を可能にすることが可能なシステム。

【0300】

10

20

30

40

50

3. マルチ・パス・マルチ・ソース・メディア配信システム。この場合において、オリジナル・サーバーは、異なるキャッシュに対して、同じメディア資産の異なる機能的に等価な線型コード化されたコピーを生成し、新たなキャッシュは、既存のコピーを再符号化して新しい機能的に等価な線型コード化されたコピーを取得し、受信者は複数のソース/キャッシュからデータをダウンロードして元のメディア資産を復元する。

【0301】

4. コンテンツ制作者が線型コーディング情報を暗号化し、解読鍵を認定者に与えるだけであるDRMシステム。これにより、認定者のみがコンテンツを再符号化及び復号化することができる。

【0302】

5. データ完全性認証を可能にする分散型メディア配信システム。この場合において、ネットワーク・エンティティは、開示されるビットストリームのデータ完全性フィールドを使用して、受信したデータの完全性を検証し、敵対的なノードに対する不正証明 (fraud proofs) を、ペナルティのために規制マネージャに提出することができる。

【0303】

6. 無線アップリンク及びダウンリンク・チャネルの双方において、メディア及び/又はデータのコーディングされたマルチソース配信を可能にすること。アップリンク・チャネルの例示的な実施形態は、電子ニュース収集アプリケーションを含む。

【0304】

7. 5G/6Gエッジ・アプリケーション・サーバーにおける効率的/スケーラブルな処理のために、開示される発明に従って生成及び定式化されたコーディングされたマルチソース・メディアをサポートするエンド・ツー・エンドの解決策を可能にするとともに、同じ5G/6Gエッジ・アプリケーション・サーバーが、遅延に敏感なアプリケーションのために、コーディングされたユニキャスト又はブロードキャスト・トラフィックへのリアルタイム変換をサポートすることを可能にする。

【0305】

[0259] 本件明細書は、多くの具体的な実装の詳細を含んでいるが、これらは、クレームされる可能性のある範囲に対する限定として解釈されるべきではなく、むしろ、特定の実施形態に特有である可能性のある特徴の説明として解釈されるべきである。個別の実施形態の文脈で本件明細書に記載されている特定の特徴は、単一の実施形態の中で組み合わせて実施されることも可能である。逆に、単一の実施形態の文脈で記載されている様々な特徴は、複数の実施形態において別々に、又は何らかの適切なサブ・コンビネーションで実施されることも可能である。更に、特徴は、特定の組み合わせで動作するものとして上述され、そのように当初はクレームされいたとしても、クレームされる組み合わせ中の1つ以上の特徴は、場合によっては、組み合わせから除去することが可能であり、クレームされた組み合わせは、サブ・コンビネーション又はサブ・コンビネーションの変形に関連している可能性がある。図面に示される論理フローは、所望の結果を達成するために、図示された特定の順序又は連続した順序を必須としていない。更に、他のステップが提供されてもよいし、又は、説明されたフローからステップが除去されてもよく、他の構成要素が、説明されたシステムに追加されてもよいし、又は、説明されたシステムから除去されてもよい。従って、他の実装は、以下のクレームの範囲内にある。

10

20

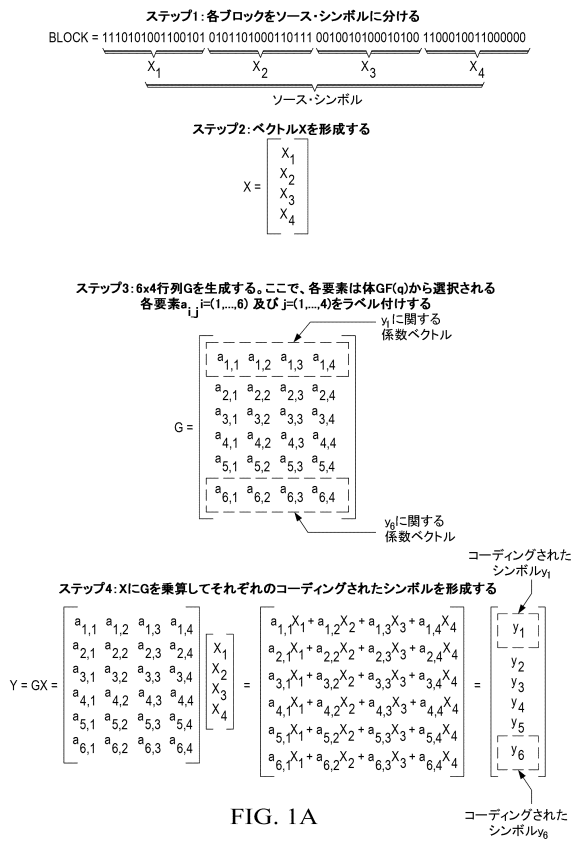
30

40

50

【図面】

【図 1 A】



【図 1 B】

ステップ1: それぞれ受信したコーディングされたシンボル y_i に関連する係数ベクトルを決定する
 このベクトルは行列Aの行になる

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix}$$

y_1 に関する係数ベクトル

y_6 に関する係数ベクトル

ステップ2: Aのインバースをとり、 A^{-1} を形成する
 $A^{-1} = \text{INVERSE}(A)$

ステップ3: コーディングされたシンボル y_i のベクトルに A^{-1} を乗算して、 x_i のベクトルを復元する

$$X = A^{-1} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

y_1 に関するコーディングされたシンボル

y_6 に関するコーディングされたシンボル

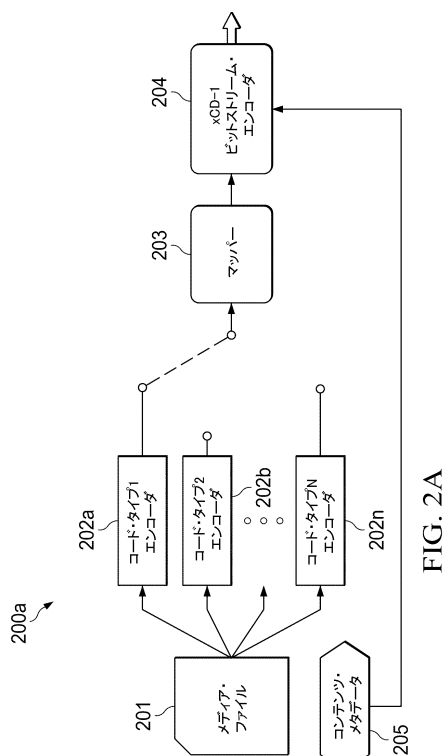
ステップ4: 各々のシンボル x_i を一緒に連結してブロックを再形成する

BITSTREAM FILE = 1110101001100101 0101101000110111 0010010100010100 1100010011000000

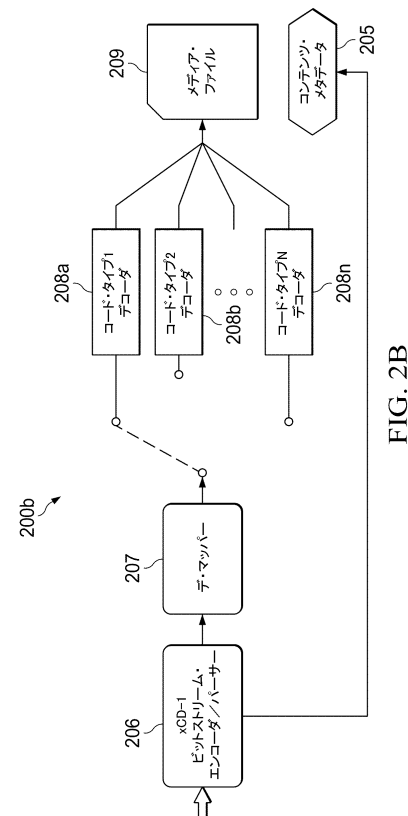
ソース・シンボル

FIG. 1B

【図 2 A】



【図 2 B】



10

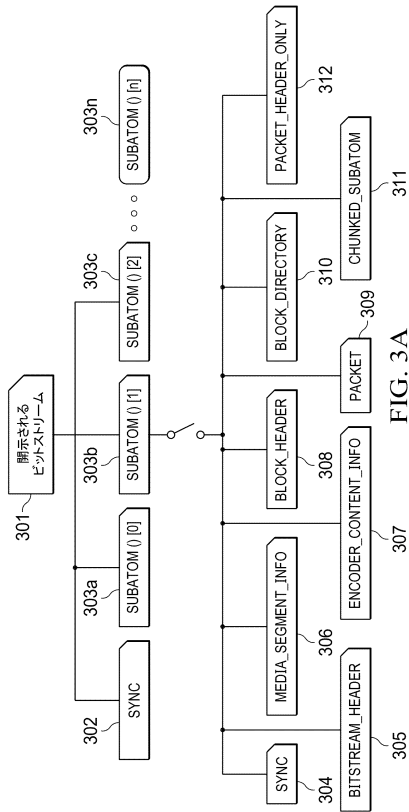
20

30

40

50

【図 3 A】



【図 3 B】

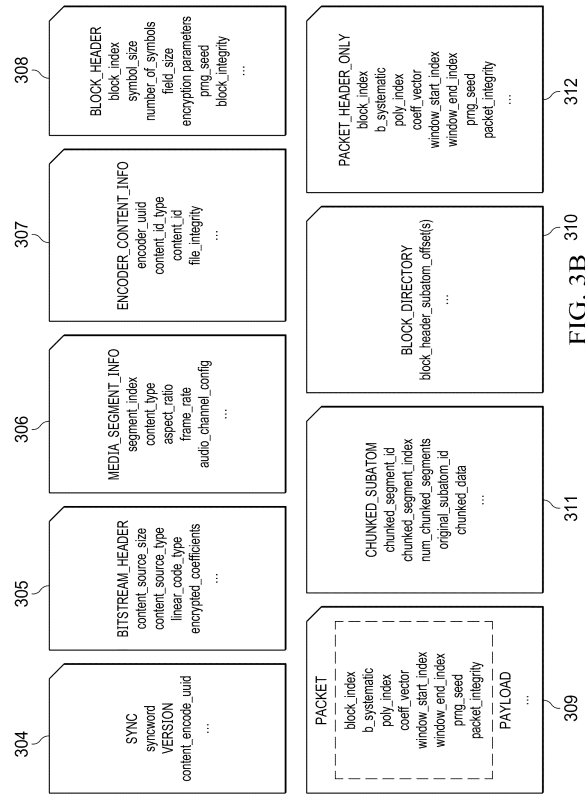


FIG. 3

【圖 4】

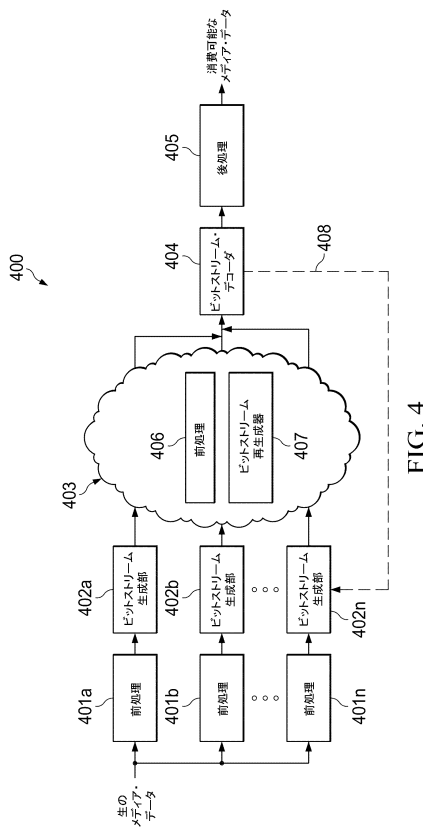


FIG. 4

【図 5】

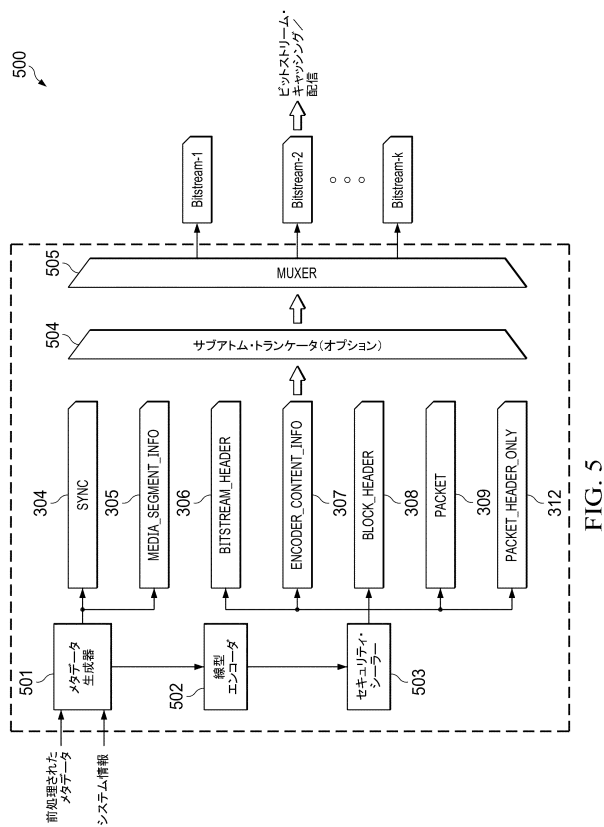


FIG. 5

【図 6】

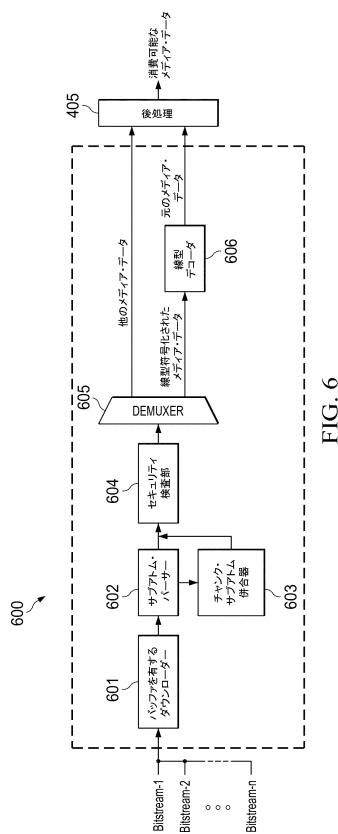


FIG. 6

【図 7】

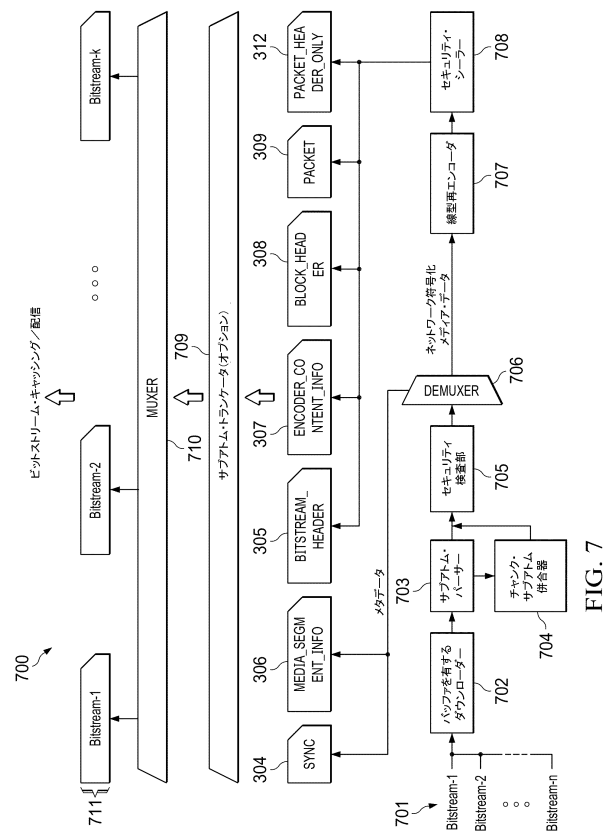


FIG. 7

【図 8 A】

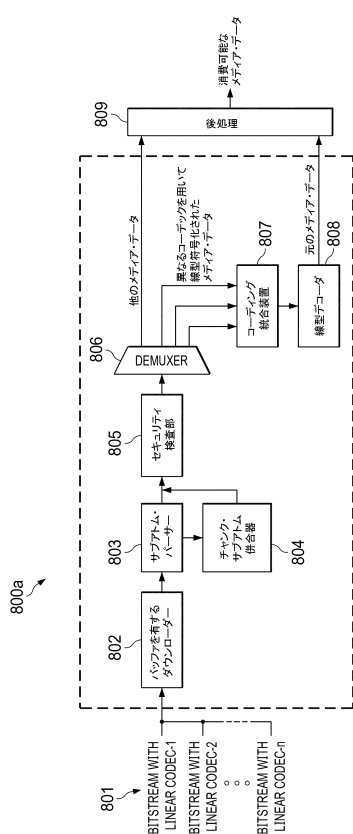


FIG. 8A

【図 8 B】

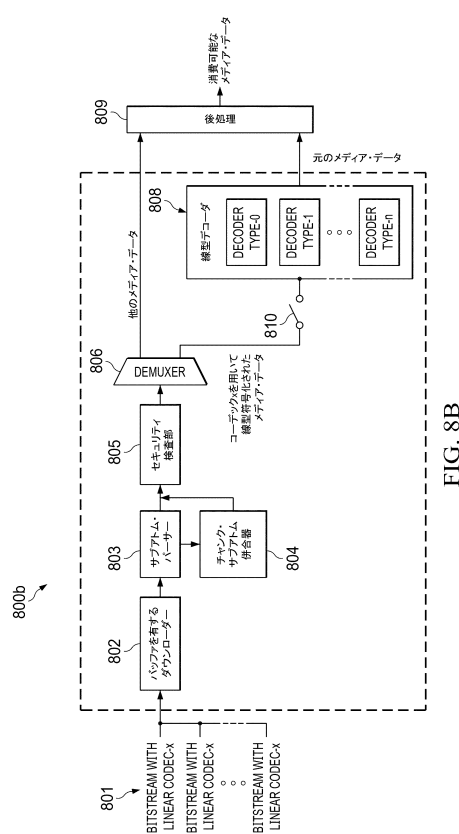
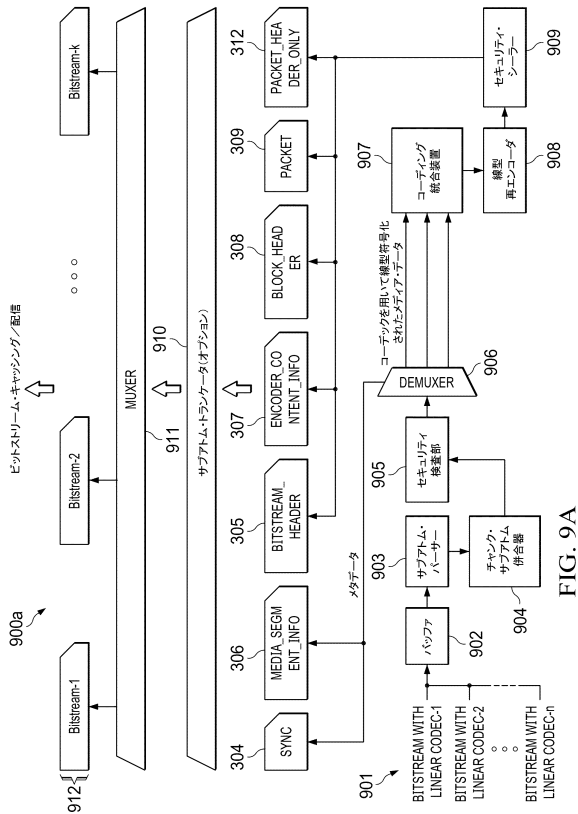
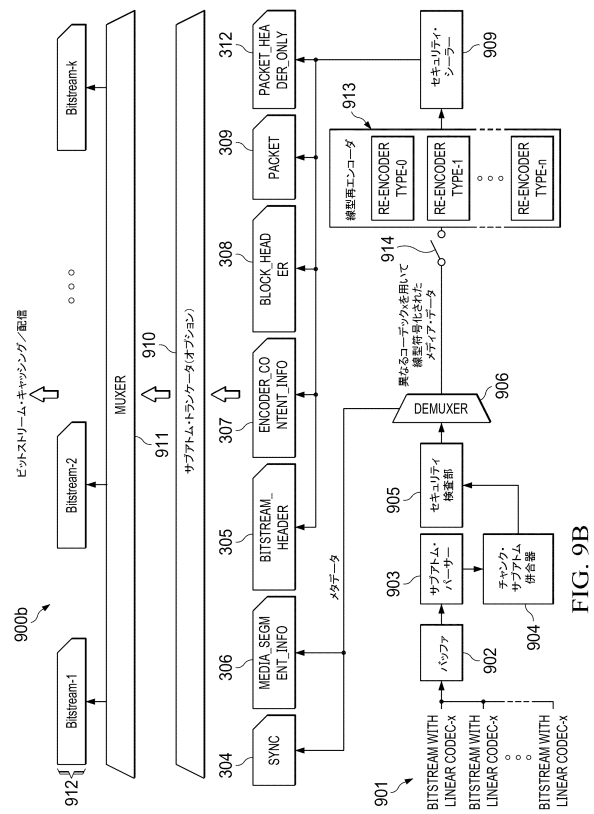


FIG. 8B

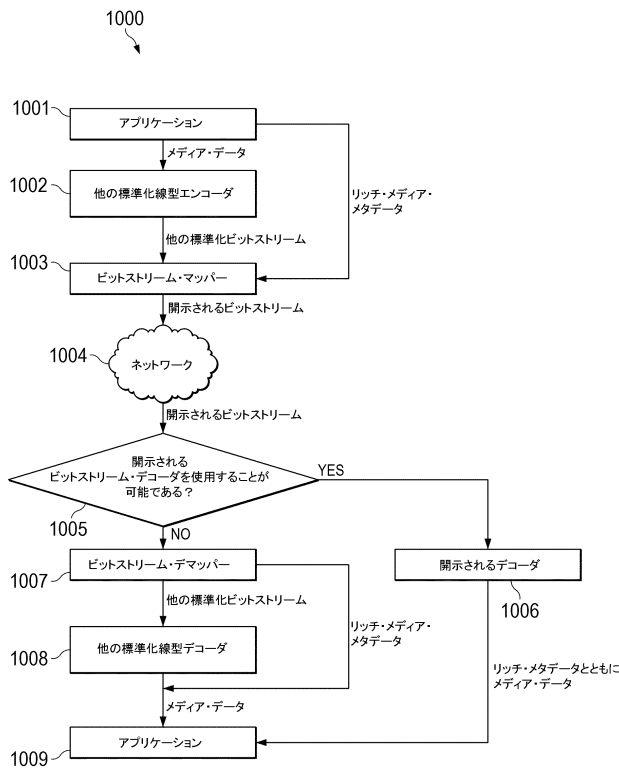
【図 9 A】



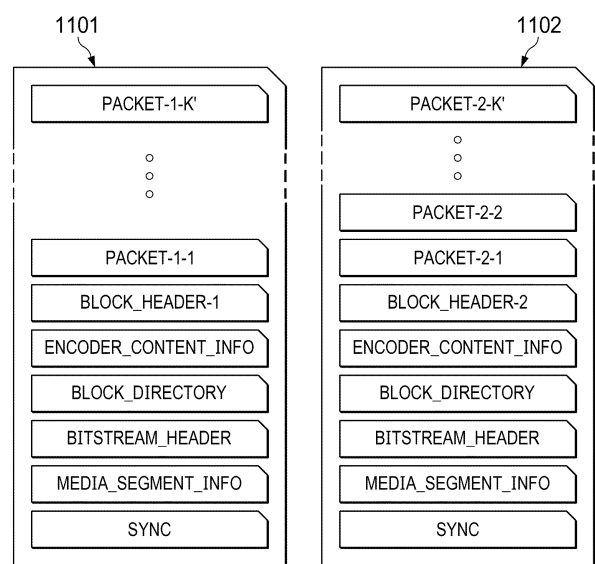
【図 9 B】



【図 10】



【図 11】



10

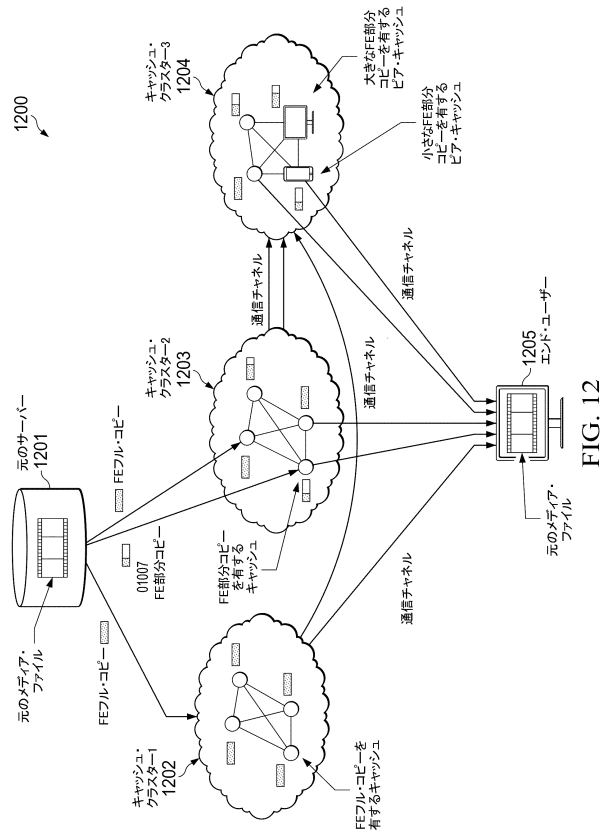
20

30

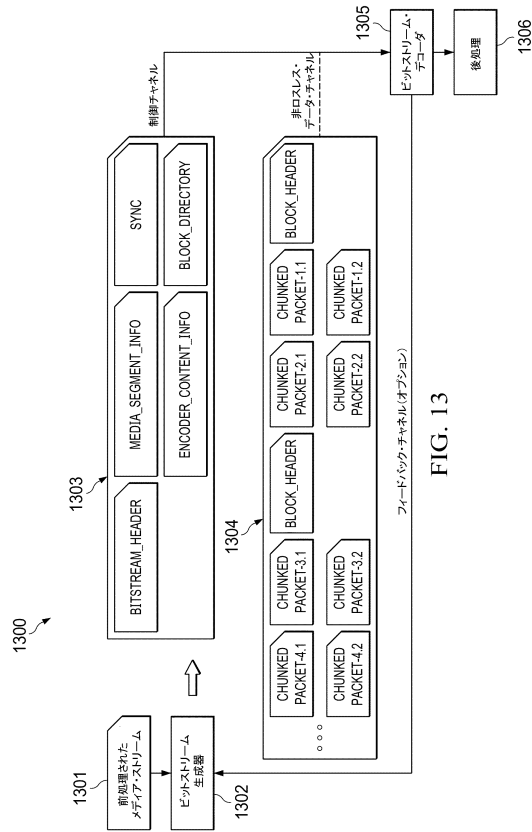
40

50

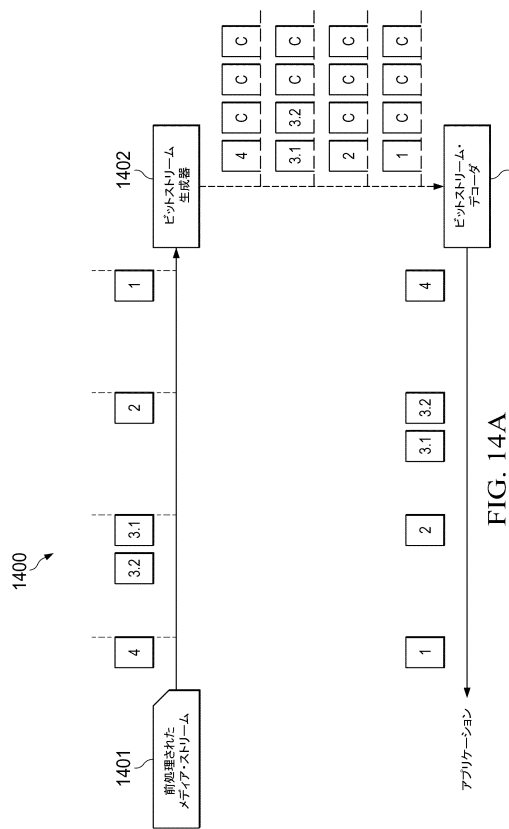
【図 1 2】



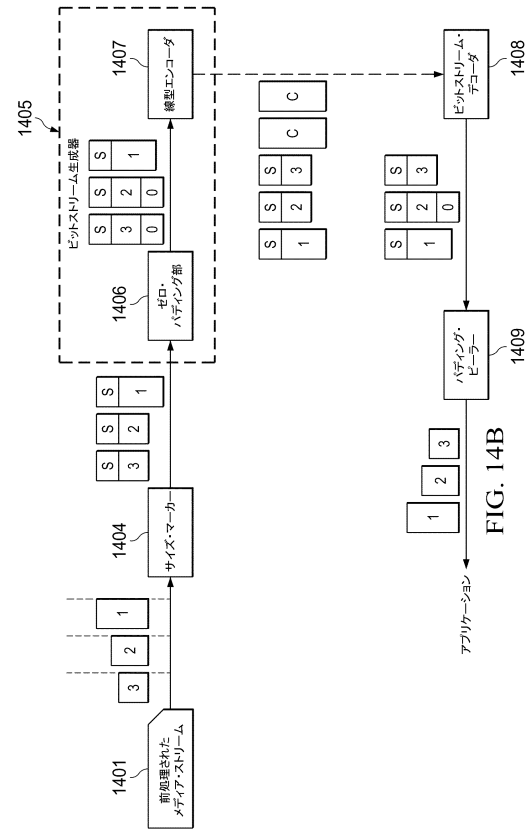
【図 1 3】



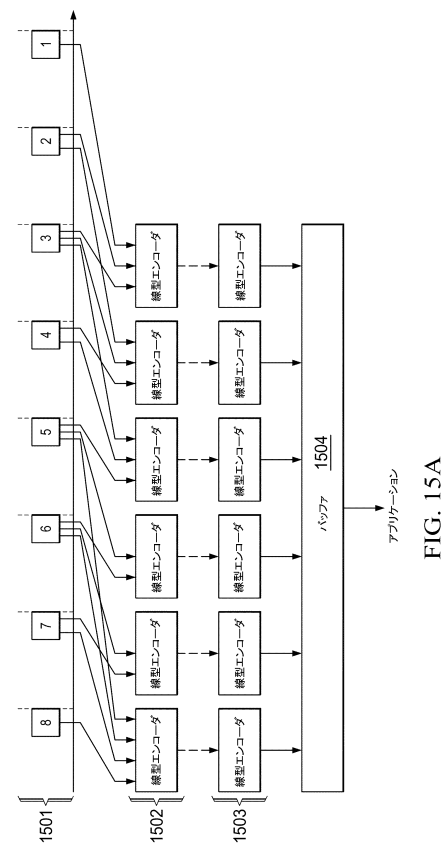
【図 1 4 A】



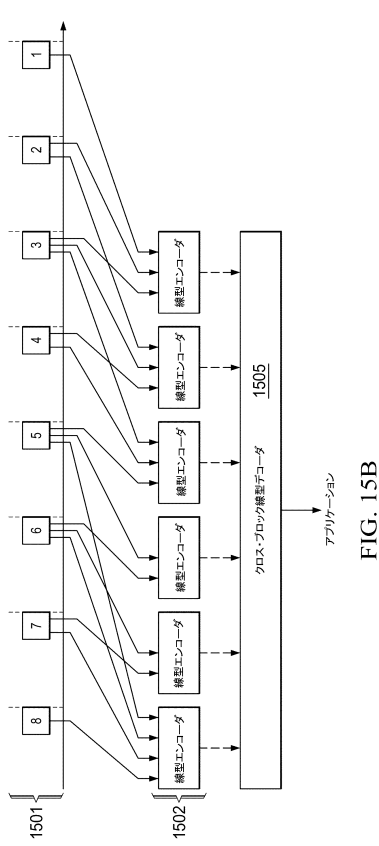
【図 1 4 B】



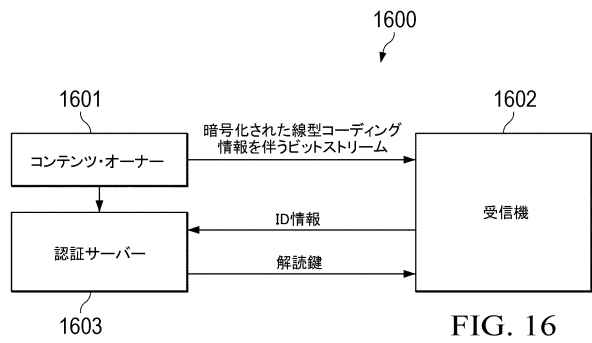
【図 15 A】



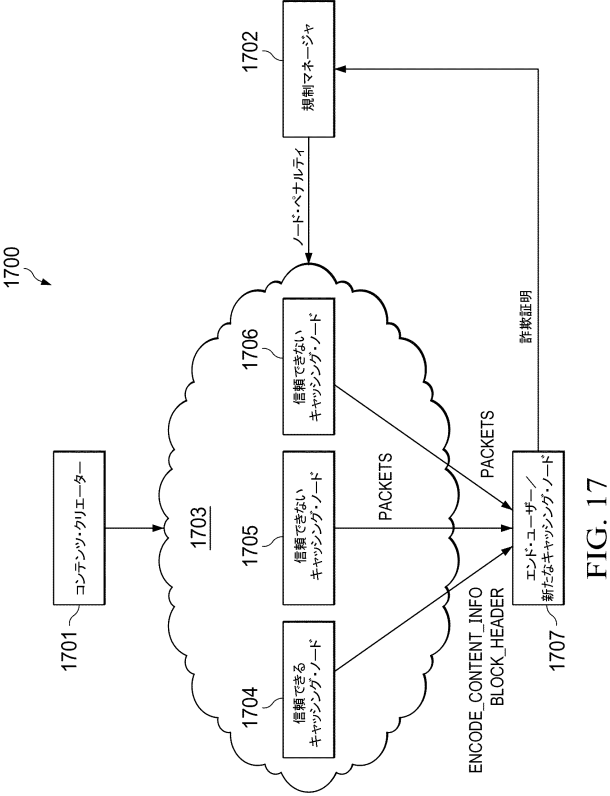
【図 15 B】



【図 16】



【図 17】



10

20

30

40

50

【図 18】

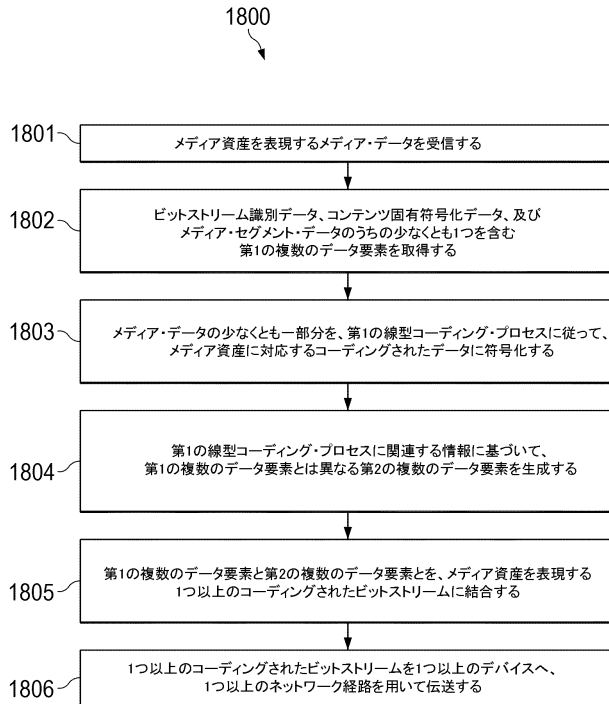


FIG. 18

【図 19】

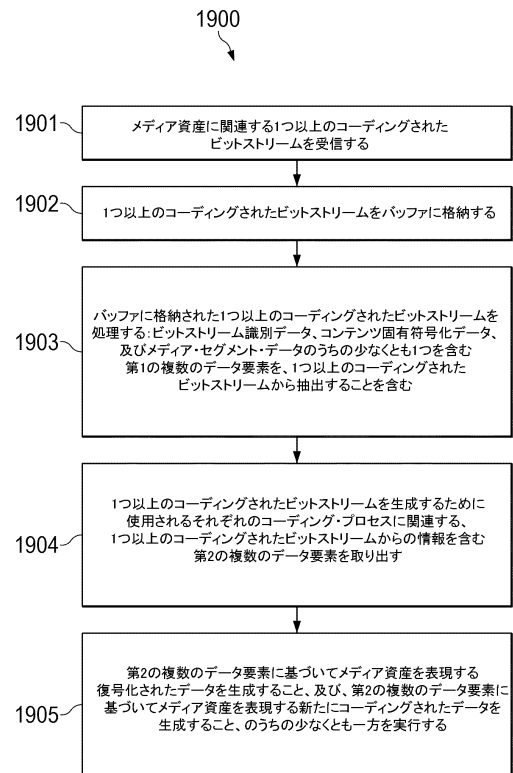


FIG. 19

【図 20】

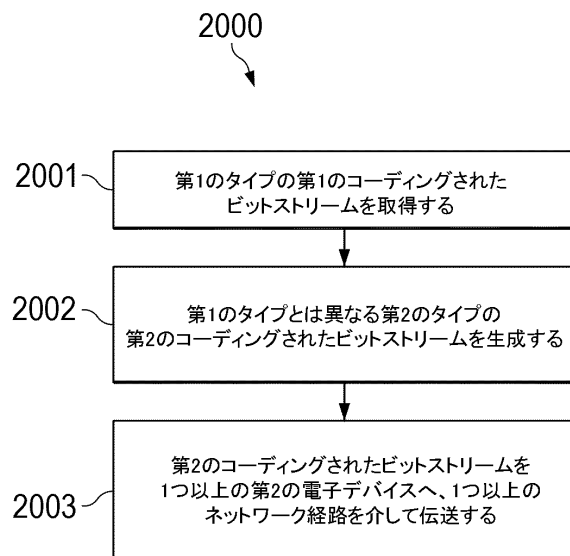


FIG. 20

【図 21】

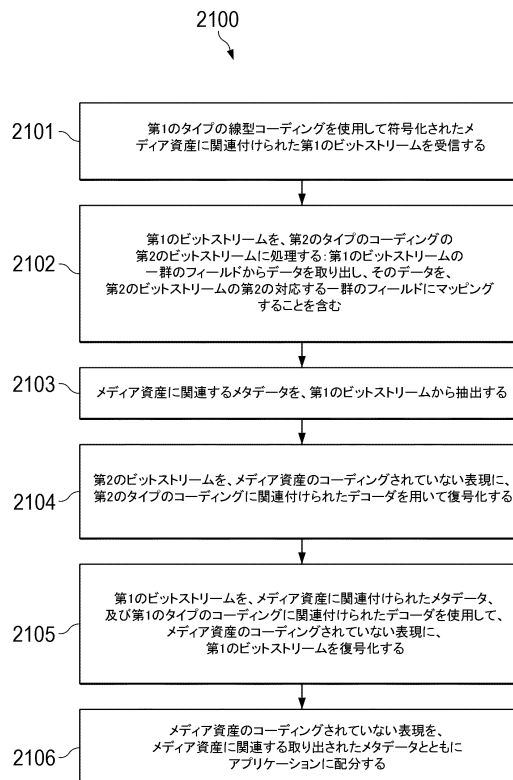


FIG. 21

【図 22】

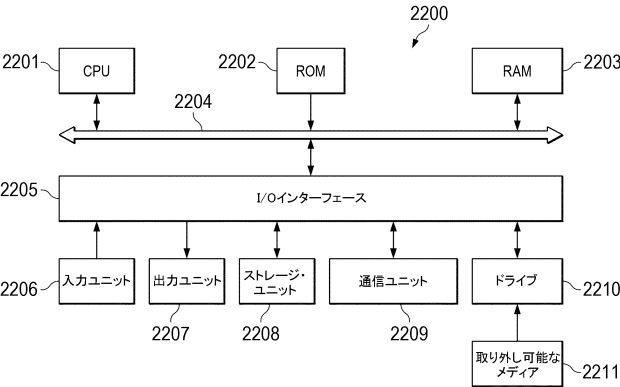


FIG. 22

10

20

30

40

50

【手続補正書】

【提出日】令和6年10月22日(2024.10.22)

【手続補正1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項1】

第1のデバイスにおける方法であって：

10

少なくとも1つのプロセッサを使用して、メディア資産を表現するメディア・データを受信するステップ；

前記少なくとも1つのプロセッサを使用して、前記メディア資産に関連する第1の複数のデータ要素であって、ビットストリーム識別データ、コンテンツ固有エンコード・データ、及びメディア・セグメント・データのうちの少なくとも1つを含む第1の複数のデータ要素を取得するステップ；

前記少なくとも1つのプロセッサを使用して、前記メディア・データの少なくとも一部分を、第1のコーディング・プロセスに従って符号化して、前記メディア資産に対応するコーディングされたデータを取得するステップ；

前記少なくとも1つのプロセッサを使用して、前記第1の複数のデータ要素とは異なる第2の複数のデータ要素を生成するステップであって、前記第2の複数のデータ要素は、前記第1のコーディング・プロセスに関連する符号化情報と前記コーディングされたデータを搬送している、ステップ；

20

前記少なくとも1つのプロセッサを使用して、前記第1の複数のデータ要素と前記第2の複数のデータ要素とを、前記メディア資産を表現する1つ以上のコーディングされたビットストリームに結合するステップ；及び

前記少なくとも1つのプロセッサを使用して、前記1つ以上のコーディングされたビットストリームを1つ以上の第2のデバイスへ、1つ以上のネットワーク経路を用いて伝送するステップ；

を含む方法。

30

【請求項2】

請求項1に記載の方法において、前記第1のコーディング・プロセスは、前記メディア資産に対応するコーディングされたデータを取得するための線型コーディング・プロセスである、方法。

【請求項3】

請求項1に記載の方法において、前記第1のコーディング・プロセスは、非線型コーディング・プロセスであって、前記メディア・データを、別の線型コーディング・プロセスに合致するコーディングされたデータに変換するものである、方法。

【請求項4】

請求項1に記載の方法において、前記第1の複数のデータ要素は、コンテンツ固有エンコード・データ又はメディア・セグメント・データのうちの少なくとも1つとビットストリーム識別データとを含む、方法。

40

【請求項5】

請求項1に記載の方法において、前記データ要素はデータ構造である、方法。

【請求項6】

請求項1に記載の方法において、前記第1の複数のデータ要素は同期データ要素を含み、前記1つ以上のコーディングされたビットストリームの各々は同期データ要素で始まる、方法。

【請求項7】

請求項1に記載の方法において、同期データ要素は：

50

前記1つ以上のコーディングされたビットストリームのそれぞれのコーディングされたビットストリームが、共通のソース・メディア資産から生成されたことを識別する識別データ；

前記1つ以上のコーディングされたビットストリームのそれぞれのコーディングされたビットストリームを、前記共通のソース・メディア資産から生成された他のコーディングされたビットストリームと同期するための同期データ；及び

前記1つ以上のコーディングされたビットストリームのそれぞれのコーディングされたビットストリームの系譜データ；

のうちの少なくとも1つを含む、方法。

【請求項8】

請求項7に記載の方法において、系譜データは：

前記それぞれのコーディングされたビットストリームに含まれるコーディングされたシンボルを生成するために使用された元の又はコーディングされたシンボルを識別するデータ；

前記それぞれのコーディングされたビットストリームに含まれるコーディングされたシンボルを生成するために、元の又はコーディングされたシンボルがどのように結合されたかを識別するデータ；及び

前記それぞれのコーディングされたビットストリームに含まれるコーディングされたシンボルを生成するために使用された符号化係数を識別するデータ；

を含む、方法。

【請求項9】

請求項1に記載の方法において、前記1つ以上のコーディングされたビットストリーム全体にわたって同期データ要素をインターリーブするステップを更に含み、前記同期データ要素は、前記1つ以上のコーディングされたビットストリームのうちの1つを受信する第2の電子デバイスが、前記1つ以上のコーディングされたビットストリームのうちの1つの中で、複数のポイントから復号化を開始できるようにする、方法。

【請求項10】

請求項1に記載の方法において、前記第1の複数のデータ要素又は前記第2の複数のデータ要素は、受信したメディア・データに関連するシステム・データに少なくとも部分的に基づいて生成される、方法。

【請求項11】

請求項1に記載の方法において、前記第1の複数のデータ要素を取得するステップは、受信したメディア・データ又は受信したメディア・データに関連するシステム・データから、前記第1の複数のデータ要素を生成するステップを含む、方法。

【請求項12】

請求項1に記載の方法において、前記メディア・データを第1のコーディング・プロセスに従って符号化することは、前記メディア・データを複数の元のシンボルに分け、前記複数の元のシンボルに基づいて複数のコーディングされたシンボルを生成することを含む、方法。

【請求項13】

請求項12に記載の方法において、ネットワーク・コーディング技術に従って前記複数の元のシンボルのサブセットを線型結合することによって、コーディングされたシンボルが生成される、方法。

【請求項14】

請求項1に記載の方法において、前記メディア・データは線型コーディングされたビットストリームを含む、方法。

【請求項15】

請求項1に記載の方法において、第1の複数のデータ要素を取得するステップは：

前記メディア・データからメタデータを取り出すステップ；及び

取り出されたメタデータから前記第1の複数のデータ要素を演算するステップ；

10

20

30

40

50

を含む、方法。

【請求項 1 6】

請求項 1 に記載の方法において、線型復号化を実行することなく、前記メディア・データからメタデータが取り出される、方法。

【請求項 1 7】

請求項 1 に記載の方法において、前記メディア・データを第 1 のコーディング・プロセスに従って符号化することは、前記コーディングされたビットストリーム内の線型コーディングされたシンボルを、前記メディア資産を表現する新たな線型コーディングされたシンボルに再符号化することを含む、方法。

【請求項 1 8】

請求項 1 に記載の方法において、結合するステップの前に、第 1 のサイズを有する 1 つ以上のデータ要素を、前記第 1 のサイズより小さな第 2 のサイズを有するデータ要素に分けるステップを更に含む、方法。

【請求項 1 9】

請求項 1 に記載の方法において、ハッシュ・データを、前記第 2 の複数のデータ要素のうちの 1 つ以上のデータ要素に追加するステップを更に含み、これにより、前記 1 つ以上のビットストリームを受信するデバイスによる検証、認証、又は完全性のうちの少なくとも 1 つを可能にする、方法。

【請求項 2 0】

請求項 1 に記載の方法において、前記第 2 の複数のデータ要素の 1 つ以上のデータ要素であって、前記メディア資産に対応するファイル、前記メディア資産のブロック、前記メディア資産のセグメント、及び前記メディア資産のパケットのうちの 1 つ以上に関連付けられるものに、ハッシュ・データを追加するステップを更に含む方法。

【請求項 2 1】

請求項 1 に記載の方法において、結合するステップは、ディレクトリ・データ要素を、前記 1 つ以上のコーディングされたビットストリームに挿入するステップを含む方法。

【請求項 2 2】

請求項 2 1 に記載の方法において、前記ディレクトリ・データ要素は、各々のビットストリーム内のサブアトムを識別する際に受信デバイスを支援するために使用される、方法。

【請求項 2 3】

請求項 1 に記載の方法において、前記 1 つ以上のコーディングされたビットストリームは、第 1 のタイプのデータ要素を含む第 1 のコーディングされたビットストリームと、前記第 1 のタイプのデータ要素を含まない第 2 のコーディングされたビットストリームとを含む、方法。

【請求項 2 4】

請求項 1 に記載の方法において、クライアントにおけるデコーダを初期化するために使用される前記データ要素又はサブアトムは、遅延に敏感なデータではない、方法。

【請求項 2 5】

請求項 1 に記載の方法において、ビットストリームは、`a bitstream_header()`、`block_header()`、`encoder_content_info()` 又は `media_segment_info()` `subatoms` のうちの少なくとも 1 つと、`sync()` `subatom` とを含む、方法。

【請求項 2 6】

請求項 1 に記載の方法において、前記データ要素又はサブアトムは、符号化されたデータ又は遅延に敏感なデータを搬送している、方法。

【請求項 2 7】

請求項 1 に記載の方法において、ビットストリームは、`packet()` `subatoms` と `sync()` `subatoms` を含むが、`bitstream_header()`、`block_header()`、`encoder_content_info()` 又は `media_segment_info()` `subatoms` のうちの少なくとも 1 つを含んでいない、方法。

10

20

30

40

50

【請求項 28】

請求項1に記載の方法において、伝送するステップは：

第1のタイプの通信チャネルを用いて前記第1のコーディングされたビットストリームを伝送するステップ；及び

前記第1のタイプとは異なる第2のタイプの通信チャネルを用いて第2のコーディングされたビットストリームを伝送するステップ；

を含む、方法。

【請求項 29】

請求項28に記載の方法において、前記第2のタイプの通信チャネルは、高遅延チャネル、低帯域幅チャネル、信頼性のあるチャネル、又は、第1のタイプの伝送プロトコルとは異なる第2のタイプの伝送プロトコルを使用するチャネルである、方法。

10

【請求項 30】

第1のデバイスにおける方法であって：

少なくとも1つのプロセッサを使用して、メディア資産に関連する1つ以上のコーディングされたビットストリームを受信するステップ；

前記少なくとも1つのプロセッサを使用して、前記1つ以上のコーディングされたビットストリームをバッファに格納するステップ；

前記バッファに格納された前記1つ以上のコーディングされたビットストリームを処理するステップであって：

前記少なくとも1つのプロセッサを使用して、前記メディア資産に関連する第1の複数のデータ要素であって、ビットストリーム識別データ、コンテンツ固有エンコード・データ、及びメディア・セグメント・データのうちの少なくとも1つを含む第1の複数のデータ要素を、前記1つ以上のコーディングされたビットストリームから抽出し；

20

前記少なくとも1つのプロセッサを使用して、前記1つ以上のコーディングされたビットストリームを生成するために使用されるそれぞれのコーディング・プロセスに関連する、前記1つ以上のコーディングされたビットストリームからの情報と前記メディア資産に対応するコーディングされたデータとを含む第2の複数のデータ要素を取り出すことを含むステップ；及び

前記少なくとも1つのプロセッサを使用して、前記第2の複数のデータ要素に基づいて前記メディア資産を表現する復号化されたデータを生成すること、及び、前記少なくとも1つのプロセッサを使用して、前記第2の複数のデータ要素に基づいて前記メディア資産を表現する新たにコーディングされたデータを生成すること、のうちの一方を実行するステップ；

30

を含む方法。

【請求項 31】

請求項30に記載の方法において、復号化されたデータを生成することは、コーディングされたデータを生成するために使用された符号化タイプの決定に従って、複数のデコーダのうちのそれぞれのデコーダを使用して、コーディングされたデータを復号化することを含む、方法。

【請求項 32】

請求項31に記載の方法において、前記符号化タイプは、取り出された第1の複数のデータ要素から決定される、方法。

40

【請求項 33】

請求項31に記載の方法において、前記符号化タイプは、取り出された第2の複数のデータ要素から決定される、方法。

【請求項 34】

請求項31に記載の方法において、前記符号化タイプは、取り出された第2の複数のデータ要素を使用することなく、前記第1の複数のデータ要素から決定される、方法。

【請求項 35】

請求項31に記載の方法において、前記符号化タイプは、取り出された第1の複数のデ

50

ータ要素を使用することなく、前記第2の複数のデータ要素から決定される、方法。

【請求項36】

請求項31に記載の方法において、復号化されたデータ及び前記第1の複数のデータ要素から、前記メディア資産に対応する消費可能データを生成するステップを更に含む、方法。

【請求項37】

請求項33に記載の方法において、消費可能データは、第1の電子デバイスで実行するアプリケーション、又はアプリケーションによってサポートされるメディア・ファイル・タイプと互換性がある、方法。

【請求項38】

請求項30に記載の方法において、復号化されたデータを生成することは：

線型コーディングを実行することなく、第1のタイプのコーディングされたデータを第2のタイプのコーディングされたデータに処理し；且つ

前記第2のタイプのコーディングされたデータを復号化することが可能なデコーダを使用して、前記第2のタイプのコーディングされたデータを復号化すること；

を含む、方法。

【請求項39】

請求項38に記載の方法において、前記コーディングされたデータを処理することは、係数を整合させるか、又は前記コーディングされたデータを共通の有限体に射影するコーディング統合装置によって実行される、方法。

【請求項40】

請求項30に記載の方法において、新たにコーディングされたデータを生成することは、前記第2の複数のデータ要素に含まれるコーディングされたデータを、コーディングされたデータを生成ために使用された符号化タイプの決定に従って、複数の再エンコーダのうちのそれぞれの再エンコーダを使用して処理することを含む、方法。

【請求項41】

請求項40に記載の方法において、前記符号化タイプは、取り出された第1の複数のデータ要素から決定される、方法。

【請求項42】

請求項40に記載の方法において、前記符号化タイプは、取り出された第2の複数のデータ要素から決定される、方法。

【請求項43】

請求項40に記載の方法において、前記符号化タイプは、取り出された第2の複数のデータ要素を使用することなく、前記第1の複数のデータ要素から決定される、方法。

【請求項44】

請求項40に記載の方法において、前記符号化タイプは、取り出された第1の複数のデータ要素を使用することなく、前記第2の複数のデータ要素から決定される、方法。

【請求項45】

請求項40に記載の方法において、新たにコーディングされたデータを生成することは：

線型復号化を実行することなく、第1のタイプのコーディングされたデータを、第2タイプのコーディングされたデータに処理すること；及び

第2タイプのコーディングされたデータを符号化することが可能な再エンコーダを用いて、第2タイプのコーディングされたデータを再符号化すること；

を含む、方法。

【請求項46】

請求項30に記載の方法において、コーディングされたデータを処理することは、係数を整合させるか、又は前記コーディングされたデータを共通の有限体に射影するコーディング統合装置によって実行される、方法。

【請求項47】

請求項30に記載の方法において、新たにコーディングされたデータを格納するか、又

10

20

30

40

50

は前記新たにコーディングされたデータから1つ以上のビットストリームを生成するステップ；

を更に含む方法。

【請求項 48】

請求項30に記載の方法において、前記バッファに格納された前記1つ以上の符号化ビットストリームを処理するステップの前に、第1のサイズを各々が有する前記第1の複数のデータ要素又は前記第2の複数のデータ要素のうちの第1の数量のデータ要素を、前記第1のサイズより大きな第2サイズを有する、より少ない数量のデータ要素に結合するステップを更に含む、方法。

【請求項 49】

請求項48に記載の方法において、結合するステップは、CHUNKED_SUBATOMサブアトム内のデータに基づいている、方法。

【請求項 50】

請求項30に記載の方法において：

復号化又は再－符号化の前に、第2の複数のデータ要素から完全性データを取り出すステップ；

それぞれのデータ要素がそれぞれの完全性データに基づいて検証の成功を判定したことに応じて、復号化又は再－符号化のうちの少なくとも一方を実行するステップ；及び

それぞれのデータ要素がそれぞれの完全性データに基づいて検証の不成功を判定したことに応じて、復号化又は再－符号化のうちの少なくとも一方の実行を控えるステップ；

を更に含む方法。

【請求項 51】

請求項50に記載の方法において、前記完全性データはハッシュ・データを含み、前記ハッシュ・データの検証は鍵情報に基づいている、方法。

【請求項 52】

請求項51に記載の方法において、前記鍵情報は、公開鍵又は秘密鍵のうちの少なくとも1つを含む、方法。

【請求項 53】

請求項51に記載の方法において、前記鍵情報は、認証サーバーから受信される、方法。

【請求項 54】

請求項51に記載の方法において：

復号化又は再－符号化の前に、前記第2の複数のデータ要素がそれぞれ暗号化されたデータを含んでいる旨の判定に従って、それぞれ暗号化されたデータを復号化鍵を使用して復号化するステップ；

を更に含む方法。

【請求項 55】

請求項51に記載の方法において、前記鍵情報又はハッシュ・データは、bitstream_encryption_key_idに関連している、方法。

【請求項 56】

請求項30に記載の方法において、無許可のアクセス、消費、復号化、又は再－符号化を防止するために、メディア資産に関連付けられた受信した1つ以上のコーディングされたビットストリームに対して、前記メディア資産の所有者又は作成者によって暗号化が適用される、方法。

【請求項 57】

請求項30に記載の方法において、

第1の電子デバイスに関連付けられた識別情報を、前記第1の電子デバイスとは異なる第2の電子デバイスへ伝送する送信するステップ；及び

メディア資産に関連付けられた1つ以上のコーディングされたビットストリームへのアクセス、消費、復号化、又は再－符号化を可能にする復号鍵を、前記第2の電子デバイスから受信するステップ；

を更に含む方法。

【請求項 58】

第1のデバイスにおける方法であって：

少なくとも1つのプロセッサを使用して、第1のタイプのコーディングを使用して符号化されたメディア資産に関連付けられた第1のビットストリームを受信するステップ；

前記第1のビットストリームが第2のタイプのコーディングに関連付けられたデコーダで復号化されるべきである旨の判定に従って、前記第1のビットストリームを、第2のタイプの線型コーディングの第2のビットストリームに処理するステップであって：

前記少なくとも1つのプロセッサを使用して、前記第1のビットストリームの一群のフィールドからデータを取り出し、そのデータを、前記第2のビットストリームの第2の対応する一群のフィールドにマッピングし；

前記少なくとも1つのプロセッサを使用して、前記メディア資産に関連するメタデータを、前記第1のビットストリームから取り出し；

前記少なくとも1つのプロセッサを使用して、前記第2のビットストリームを、前記メディア資産のコーディングされていない表現に、前記第2のタイプのコーディングに関連付けられたデコーダを用いて復号化することを含むステップ；

前記第1のビットストリームが第2のタイプのコーディングに関連付けられたデコーダで復号化されるべきではない旨の判定に従って、前記第1のビットストリームを処理するステップであって：

前記少なくとも1つのプロセッサを使用して、前記メディア資産に関連付けられたメタデータ、及び第1のタイプのコーディングに関連付けられたデコーダを使用して、前記メディア資産のコーディングされていない表現に、前記第1のビットストリームを復号化し；

前記少なくとも1つのプロセッサを使用して、前記メディア資産のコーディングされていない表現を、前記メディア資産に関連する取り出されたメタデータとともにアプリケーションに配分することを含むステップ；

を含む、方法。

【請求項 59】

命令を記憶する非一時的なコンピュータ読み取り可能な記憶媒体であって、前記命令は、演算装置により実行されると、請求項1ないし58のうちの何れか一項に記載の方法を前記演算装置に実行させる、記憶媒体。

【請求項 60】

少なくとも1つのプロセッサ；及び

命令を記憶するメモリ；

を含む演算装置であって、前記命令は、前記少なくとも1つのプロセッサにより実行されると、請求項1ないし58のうちの何れか一項に記載の方法を前記演算装置に実行させる、通信装置。

10

20

30

40

50

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/018469

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/40
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2018/012610 A1 (RIEDMILLER JEFFREY [US] ET AL) 11 January 2018 (2018-01-11) paragraphs [0060] - [0073], [0138], [0139], [0190]; figures 2,8 -----	1-64, 66, 67
X	US 2015/271541 A1 (GONDER TOM [US] ET AL) 24 September 2015 (2015-09-24) paragraphs [0092], [0107] ----- -/-	65

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

10 November 2023

Date of mailing of the international search report

20/11/2023

Name and mailing address of the ISA/
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Bardella, Xavier

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2023/018469

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Shilarnav Shashi: "Transcoding Transport Stream MPEG2", / 1 May 2007 (2007-05-01), pages 1-93, XP093097910, Retrieved from the Internet: URL:https://mospace.umsystem.edu/xmlui/bitstream/handle/10355/5068/research.pdf?sequence=3&isAllowed=y [retrieved on 2023-11-03] the whole document -----	1-67

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2023/018469

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2018012610 A1	11-01-2018	AU 2014281794 A1	23-07-2015
		BR 112015019435 A2	18-07-2017
		BR 122016001090 A2	27-08-2019
		BR 122017011368 A2	03-09-2019
		BR 122017012321 A2	03-09-2019
		BR 122020017896 B1	24-05-2022
		BR 122020017897 B1	24-05-2022
		CA 2898891 A1	24-12-2014
		CL 2015002234 A1	29-07-2016
		CN 104240709 A	24-12-2014
		CN 104995677 A	21-10-2015
		CN 106297810 A	04-01-2017
		CN 106297811 A	04-01-2017
		CN 110459228 A	15-11-2019
		CN 110473559 A	19-11-2019
		CN 110491395 A	22-11-2019
		CN 110491396 A	22-11-2019
		CN 110600043 A	20-12-2019
		CN 203415228 U	29-01-2014
		DE 202013006242 U1	01-08-2013
		EP 2954515 A1	16-12-2015
		EP 3373295 A1	12-09-2018
		EP 3680900 A1	15-07-2020
		ES 2674924 T3	05-07-2018
		ES 2777474 T3	05-08-2020
		FR 3007564 A3	26-12-2014
		HK 1204135 A1	06-11-2015
		HK 1214883 A1	05-08-2016
		HK 1217377 A1	06-01-2017
		IL 239687 A	29-02-2016
		JP 3186472 U	10-10-2013
		JP 6046275 B2	14-12-2016
		JP 6561031 B2	14-08-2019
		JP 6571062 B2	04-09-2019
		JP 6866427 B2	28-04-2021
		JP 7090196 B2	23-06-2022
		JP 2016507088 A	07-03-2016
		JP 2017004022 A	05-01-2017
		JP 2017040943 A	23-02-2017
		JP 2019174852 A	10-10-2019
		JP 2021101259 A	08-07-2021
		JP 2022116360 A	09-08-2022
		KR 20140006469 U	30-12-2014
		KR 20150099615 A	31-08-2015
		KR 20160088449 A	25-07-2016
		KR 20190125536 A	06-11-2019
		KR 20210111332 A	10-09-2021
		KR 20220021001 A	21-02-2022
		MX 342981 B	20-10-2016
		MX 367355 B	16-08-2019
		MY 171737 A	25-10-2019
		MY 192322 A	17-08-2022
		PL 2954515 T3	28-09-2018
		RU 2589370 C1	10-07-2016
		RU 2619536 C1	16-05-2017
		RU 2624099 C1	30-06-2017
		RU 2017122050 A	24-12-2018
		RU 2019120840 A	11-01-2021

Form PCT/ISA/210 (patent family annex) (April 2005)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2023/018469

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		SG 10201604617V A	28-07-2016
		SG 10201604619R A	28-07-2016
		SG 11201505426X A	28-08-2015
		TR 201808580 T4	23-07-2018
		TW M487509 U	01-10-2014
		TW 201506911 A	16-02-2015
		TW 201635276 A	01-10-2016
		TW 201635277 A	01-10-2016
		TW 201735012 A	01-10-2017
		TW 201804461 A	01-02-2018
		TW 201921340 A	01-06-2019
		TW 202042216 A	16-11-2020
		TW 202143217 A	16-11-2021
		TW 202244900 A	16-11-2022
		UA 111927 C2	24-06-2016
		US 2016196830 A1	07-07-2016
		US 2016307580 A1	20-10-2016
		US 2016322060 A1	03-11-2016
		US 2018012610 A1	11-01-2018
		US 2020219523 A1	09-07-2020
		US 2023023024 A1	26-01-2023
		WO 2014204783 A1	24-12-2014

US 2015271541 A1	24-09-2015	CA 2941117 A1	24-09-2015
		CA 3189286 A1	24-09-2015
		EP 3120565 A2	25-01-2017
		EP 3687175 A1	29-07-2020
		JP 6612768 B2	27-11-2019
		JP 7121693 B2	18-08-2022
		JP 2017519382 A	13-07-2017
		JP 2019208210 A	05-12-2019
		US 2015271541 A1	24-09-2015
		US 2019158906 A1	23-05-2019
		WO 2015142741 A2	24-09-2015

フロントページの続き

(32)優先日 令和5年2月7日(2023.2.7)

(33)優先権主張国・地域又は機関
米国(US)

(81)指定国・地域 AP(BW,CV,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,ST,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,ME,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CV,CZ,DE,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IQ,IR,IS,IT,JM,JO,JP,KE,KG,KH,KN,KP,KR,KW,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,MG,MK,MN,MU,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,WS,ZA,ZM,ZW

(特許庁注：以下のものは登録商標)

1. B L U E T O O T H

ニア州 サンフランシスコ マーケット ストリート 1 2 7 5 ドルビー ラボラトリーズ インコーポレイテッド 内

(72)発明者 ユィ, ミンチャオ

オーストラリア連邦 2 0 6 0 ニューサウスウェールズ州 シドニー ドリス ストリート 4 0 1 / 2 2

(72)発明者 クラウド, ジェイソン マイケル

アメリカ合衆国 9 4 1 0 3 カリフォルニア州 サンフランシスコ マーケット ストリート 1 2 7 5 ドルビー ラボラトリーズ インコーポレイテッド 内

(72)発明者 オズボーン, エリオット

アメリカ合衆国 9 4 1 0 3 カリフォルニア州 サンフランシスコ マーケット ストリート 1 2 7 5 ドルビー ラボラトリーズ インコーポレイテッド 内

(72)発明者 アンティオキア, トーマス フランクリン

アメリカ合衆国 9 4 1 0 3 カリフォルニア州 サンフランシスコ マーケット ストリート 1 2 7 5 ドルビー ラボラトリーズ インコーポレイテッド 内

F ターム (参考) 5C164 MB13S SA32S SB11P SB21S UA24S UB11P UB21S