TLS: Transport Layer Security

(a.k.a.: Secure Sockets Layer (SSL))

Network Security
総合情報学科

証明書の役割

TLSの利用に電子証明書は不可欠

通称	認証局証明書	Server証明書	Client証明書
発行者	認証局	認証局	認証局
発行対象	認証局	Webサーバ	User
役割	Systemが信頼する 認証局を指定	Web サーバの身元 証明・暗号通信	Userの身元証明
Install箇所	認証局, Webサーバ, Client	Webサーバ	Client端末

認証局に関する2つの選択肢

- 商用認証局を利用
  - 電子証明書を「購入」
- 自分で認証局を構築
  - 電子証明書を自組織で発行

価格事例 - https://www.slogical.co.jp/ssl/

https://www.symantec.com/ja/jp/page.jsp?id=compare-ssl-certificates

しかし

- 「費用がかかるのでTLSは利用しない」という選択肢
- “リスク”の増大
  - Webを利用した攻撃行為 (ID窃盗、 Phishing)
  - 盗聴、データ傍受 & Privacy保護
- 「TLS導入= サーバ証明書導入」の障害の1つはコスト

  ⇒ならば、コストなしで導入できる世界を目指そう

  Let’s Encrypt project

無料の証明書発行

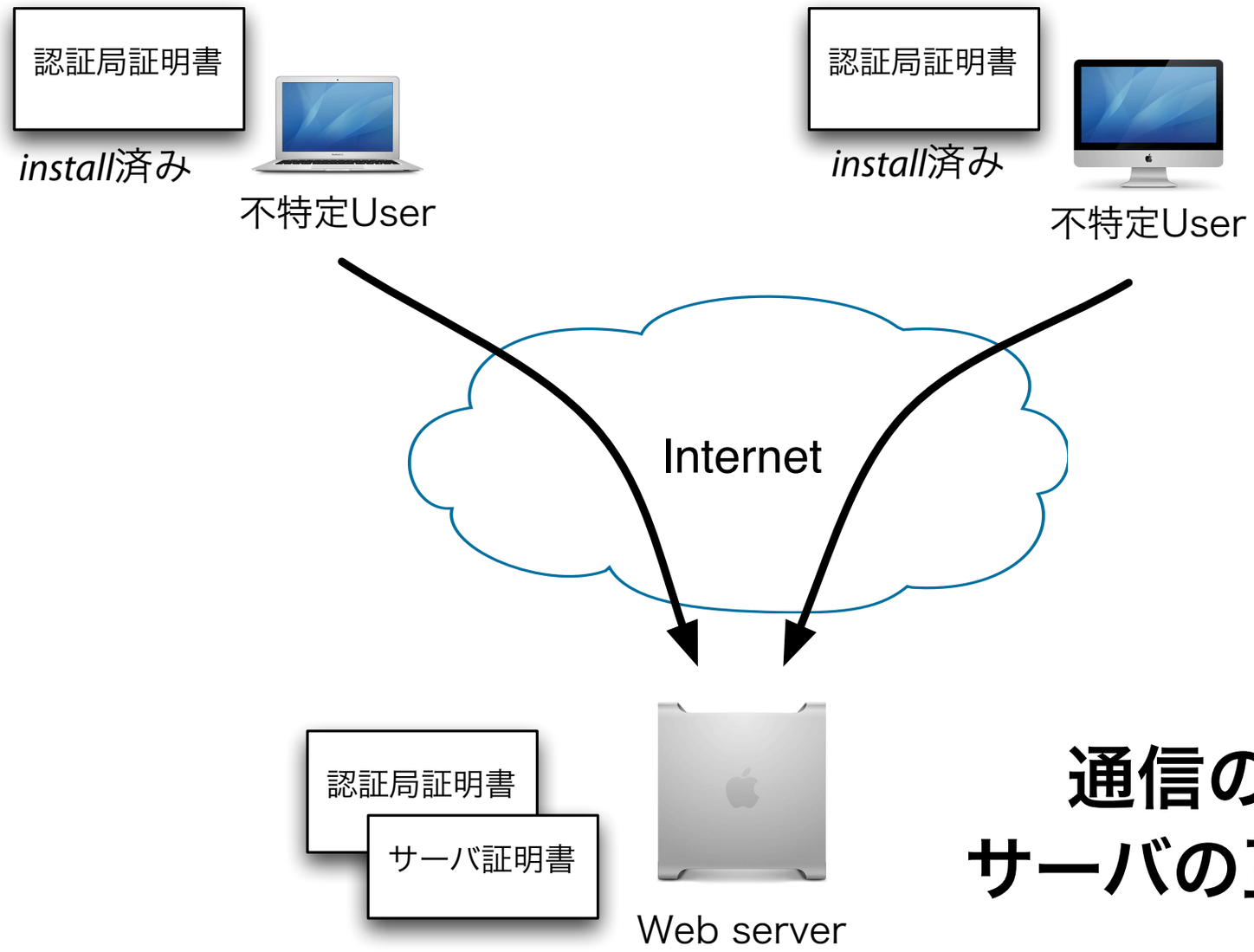
URL: https://letsencrypt.org/

Mozilla, Cisco systems, Akamai, EFF, IdenTrust, ミシガン大学からなる団体
ISRG: Internet Security Research Group

TLS:３つの導入事例

- 不特定多数のユーザ(Client)からのアクセスを暗号化
  - Server認証、通信路暗号化
- 特定のユーザであることを認証し、暗号化サービスを提供
  - Server, Client認証、通信路暗号化
- 企業システムでのアクセス制御
  - Client認証、通信路暗号化

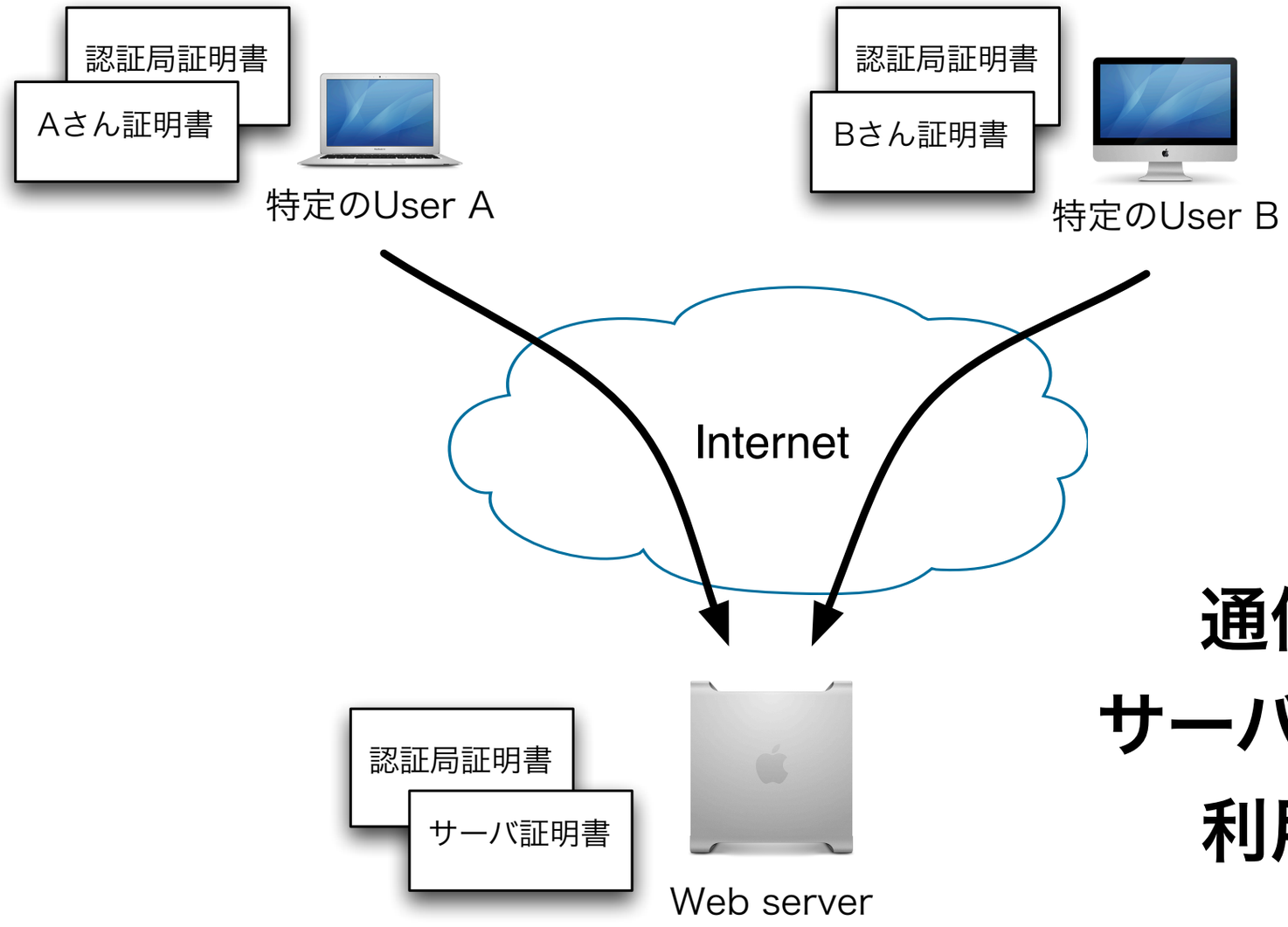
不特定多数へのアクセスを暗号化



通信の暗号化
サーバの正当性証明

著名な商用認証局なら「認証局証明書」はすでにWebブラウザにinstall済み
⇒ サーバ証明書のinstallのみで対応可能

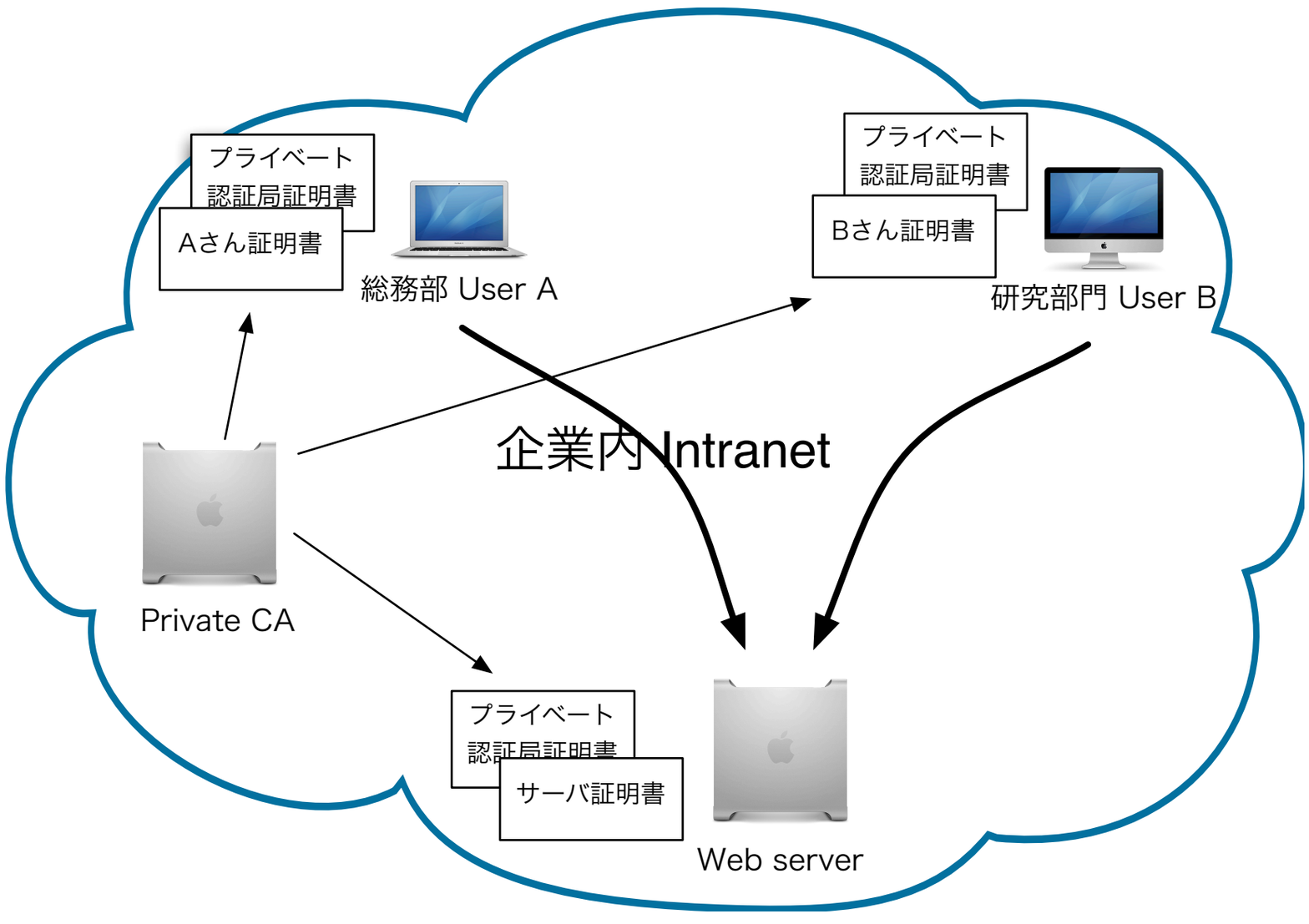
特定のユーザを認証、アクセスの暗号化を実現



通信の暗号化
サーバの正当性証明
利用者の認証

サーバ証明書の他、各ユーザにClient証明書を発行し
installする必要がある

企業システムでのアクセス制御



- アクセス制御に主眼
- 2つの情報源
  - 証明書の有無
  - 証明書内の属性情報
- Private CAの場合はserverやclientに認証局証明書のinstallが必要

電子証明書の問題

- 「鍵マークがあれば、正規のWebサイト」という誤解
  ⇒ 不正行為者もTLSを使用している...
- サーバ証明書の乱発行
  - 申請者/組織(会社)を十分審査せず/無審査で証明書発行
  - 「安易な審査」と「厳密な審査」の区別が証明書では不能

「信頼できる第三者(CA)による証明」という
PKIの信頼が揺らぐ事態に

SSLにかかわる不正行為の事例

- (2009/01/06), SSL証明書の偽造に成功、ハッシュ関数の脆弱性を応用, ITmedia, http://www.itmedia.co.jp/enterprise/articles/0901/06/news030.html
- (2009/09/03) SSLの「鍵マーク」が出ても注意を！正規のSSL証明書を悪用したフィッシングに注意, Securityblog.jp, http://securityblog.jp/news/775.html
- (2010/03/24) なりすましで認証局にログイン、電子証明書を不正発行, @IT, http://www.atmarkit.co.jp/news/201103/24/comodo.html

乱発行(?)の事例

無いもので、まさにSSL普及の為のサービスと言えるでしょう。法的な書類確認が不可欠なサービスをあえて扱わず、オンライン本人確認システムを採用、全ての手続きのオンライン化・徹底したコスト削減により**国内最低級価格・即発行（最短数分）**[*1]を実現しました。

・RSA2048bitルート証明書対応

・99% 以上のブラウザー対応

・完全自動発行，1日24時間いつでも即発行[*1]。書類送付など不要

・法人だけでなく個人でも取得可能

・30日間返金保証

・有効期限内無制限再発行オプション付き[*2]

・追加ライセンス不要（1つの証明書を複数のサーバで使用可能）

コスト重視 or 信用重視

例: http://www.rapid-ssl.jp/

EV SSL証明書

- “Extended Verification SSL Certificate”
  ⇒ 厳密で統一された審査基準による審査を経た証明書
- CA/Browser Forumという業界団体が審査基準を策定
  - 組織/団体の物理的かつ法的な存在を証明
  - 当該URLへの排他的な制御が確立している
  - 作業者と責任ある役員により署名された法的義務を伴う文書の確認
- TLSのサーバ証明書として使用された場合、EV SSL証明書と呼ばれる
  - Web browserでは、アドレスバーが緑色に変化/組織名表示で明示化

Extended Validation (EV SSL): 日本ベリサイン,
http://jp.globalsign.com/knowledge/ssl/cert/type.html#04

2種類のSSL証明書、さぁどっちを選ぶ?
http://www.atmarkit.co.jp/ad/geotrust/sslcert0905/sslcert.html

User Interface 比較

EV SSL

SSL

No SSL

Twitterは？ Googleは？

Firefox ver.26 on Mac OS X Lionでの表示

あらためて注意

- TLS/SSL通信が可能なWebサーバ ≠ ”Secure Web server”
  - TLS/SSLは「通信の機密性・完全性」と「通信相手の認証」だけ保証
  - Webサーバ自身の安全性を保証するものではない
- 通信相手の認証は「電子証明書」による
  ⇒ つまるところ「所有物認証」
  - 事例) サーバ認証: 今閲覧中のweb page URLが、証明書内のURLと同一であることが確認 ”できる” だけ
  - “www.mybank.com”と”www.mybanks.com”の違いは、人間が見分ける必要がある

SSL protocol 基礎

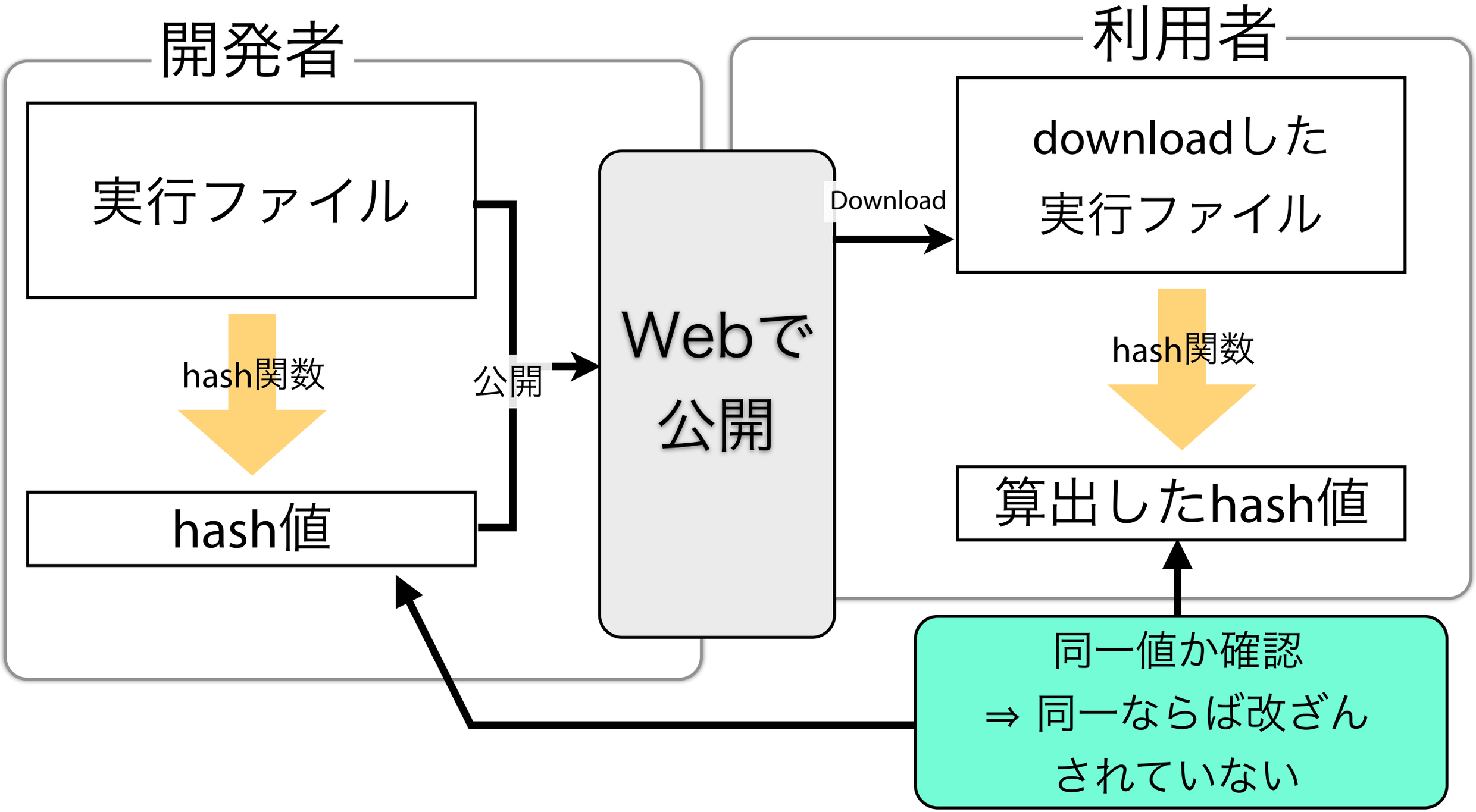
SSL 基礎

- SSL protocol はさまざまな暗号技術を組みあわせた (Integrated) ネットワーク・プロトコル
  - 特定の暗号技術には依存して「いない」
  - 利用している技術
    - 説明済み
      ⇒共通鍵暗号と公開鍵暗号, 電子署名、PKI
    - 未説明
      ⇒ハッシュ関数、どうやって機密性、完全性を担保

ハッシュ関数

- あるデータを入力とし、それを代表する”数値”を得る関数
  - 厳密には「任意長のデータブロックを入力とし、固定長ビット列であるハッシュ値を返す決定的手順」と定義。
- 理想的には以下の4つの特性を有する
  - 計算容易性
  - 不可逆性 (hash値⇒元のデータの変換が”事実上”不可能)
  - hash値を変更せずにメッセージを変更することが事実上不可能
  - 同一hash値を持つ2種類のmessageを得ることが事実上不可能

よくある使い方


開発者
実行ファイル
hash関数
hash値
公開
Webで
公開
Download
利用者
downloadした
実行ファイル
hash関数
算出したhash値
同一値か確認
⇒ 同一ならば改ざん
されていない

Hash値提示の例

Download Kali Linux Images

We generate fresh Kali Linux image files every few months, which we make available for download. This page provides the links to **download Kali Linux** in it's latest release. For a release history, check our Kali Linux Releases page. Please note: remaining torrent files for the 2016.2 release will be posted in the next few hours.

Image Name	Direct	Torrent	Size	Version	SHA1Sum
Kali Linux 64 bit	ISO	Torrent	2.9G	2016.2	25cc6d53a8bd8886fcb468eb4fbb4cdfac895c65
Kali Linux 32 bit	ISO	Torrent	2.9G	2016.2	9b4e167b0677bb0ca14099c379e0413262eefc8c
Kali Linux 64 bit Light	ISO	Torrent	1.1G	2016.2	f7bdc3a50f177226b3badc3d3eafcf1d59b9a5e6

Hash値が提示されている

ハッシュ関数

- 著名なハッシュ関数
  - MD5
  - SHA-1, SHA-2(SHA-256, SHA-512)
    - GNU/Linuxではmd5sum, sha1sumというコマンドが存在する
  - 生成値は“Message Digest”とも呼ばれる

```
SHA (pureGumi.jpg) = 2aff12539296a87ca225d52815da022c1550fcb9
```

- 主な用途
  - Message Authentication Code(MAC)と電子署名

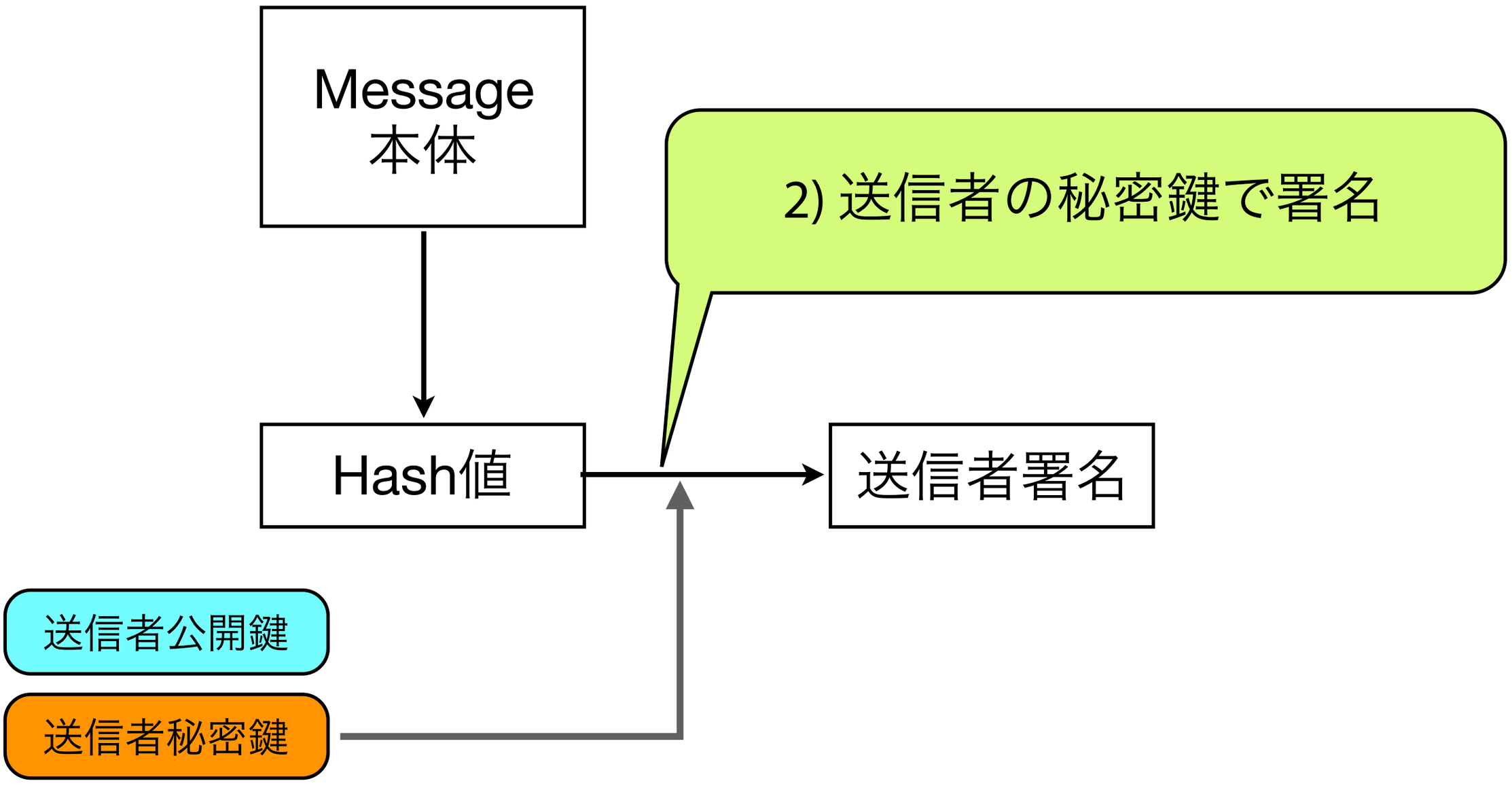
暗号の処理速度について

- 処理速度
  - 遅い：公開鍵暗号方式
  - 早い：共通鍵暗号方式
  - 公開鍵暗号方式 (低速) << 共通鍵暗号方式 (高速)
- セオリー（常套手段）
  ⇒ 公開鍵暗号の処理量を少なくする
  - 「共通鍵暗号の鍵」を公開鍵暗号で暗号化
  - 電子署名も 「任意長のメッセージのHash」を対象
    ⇒ データ長が一定化（少量）

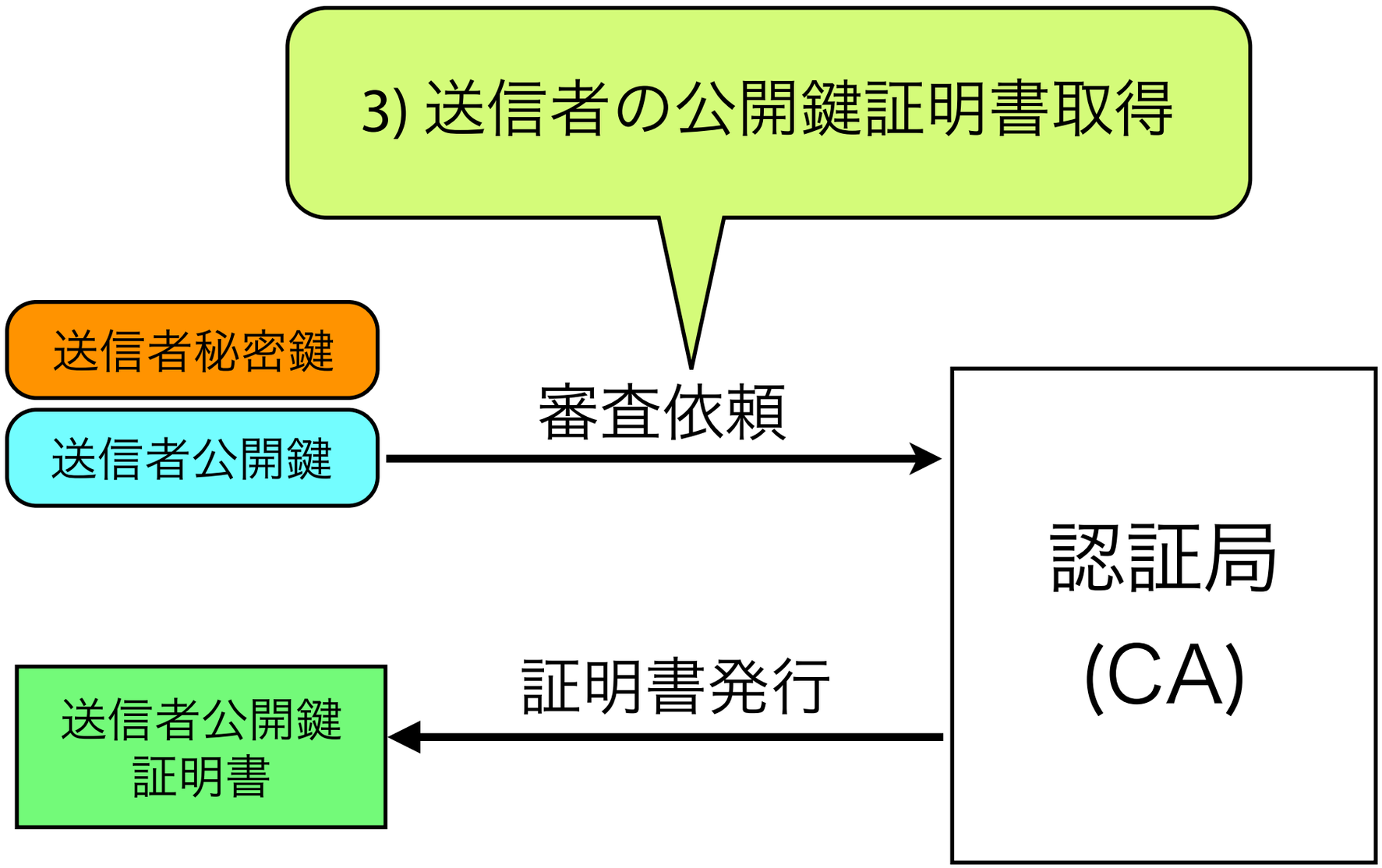
安全なMessage交換 (送信側)

あるMessageを通信相手に安全に送付する基本モデル

安全なMessage交換 (送信側)

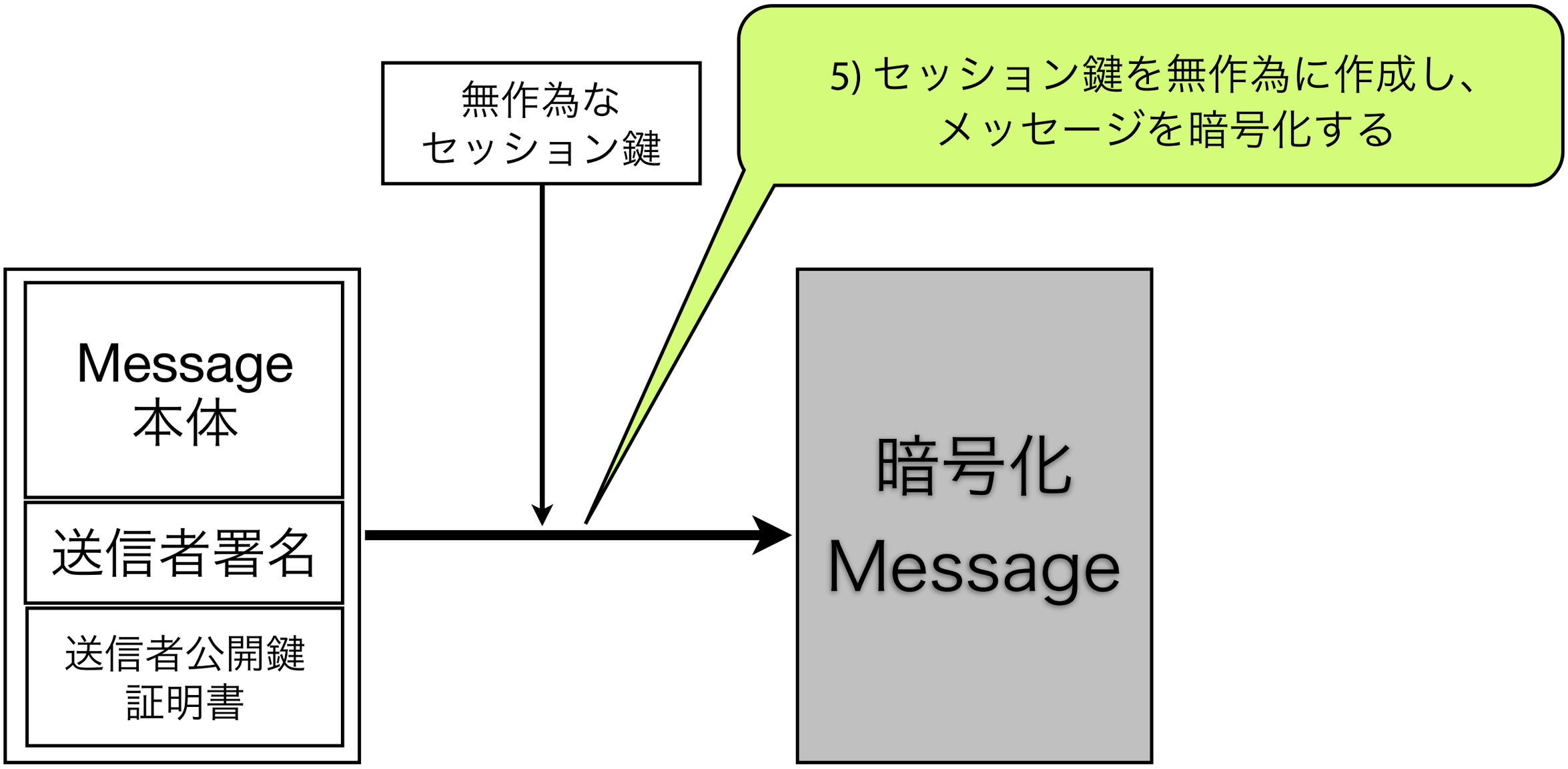


安全なMessage交換 (送信側)

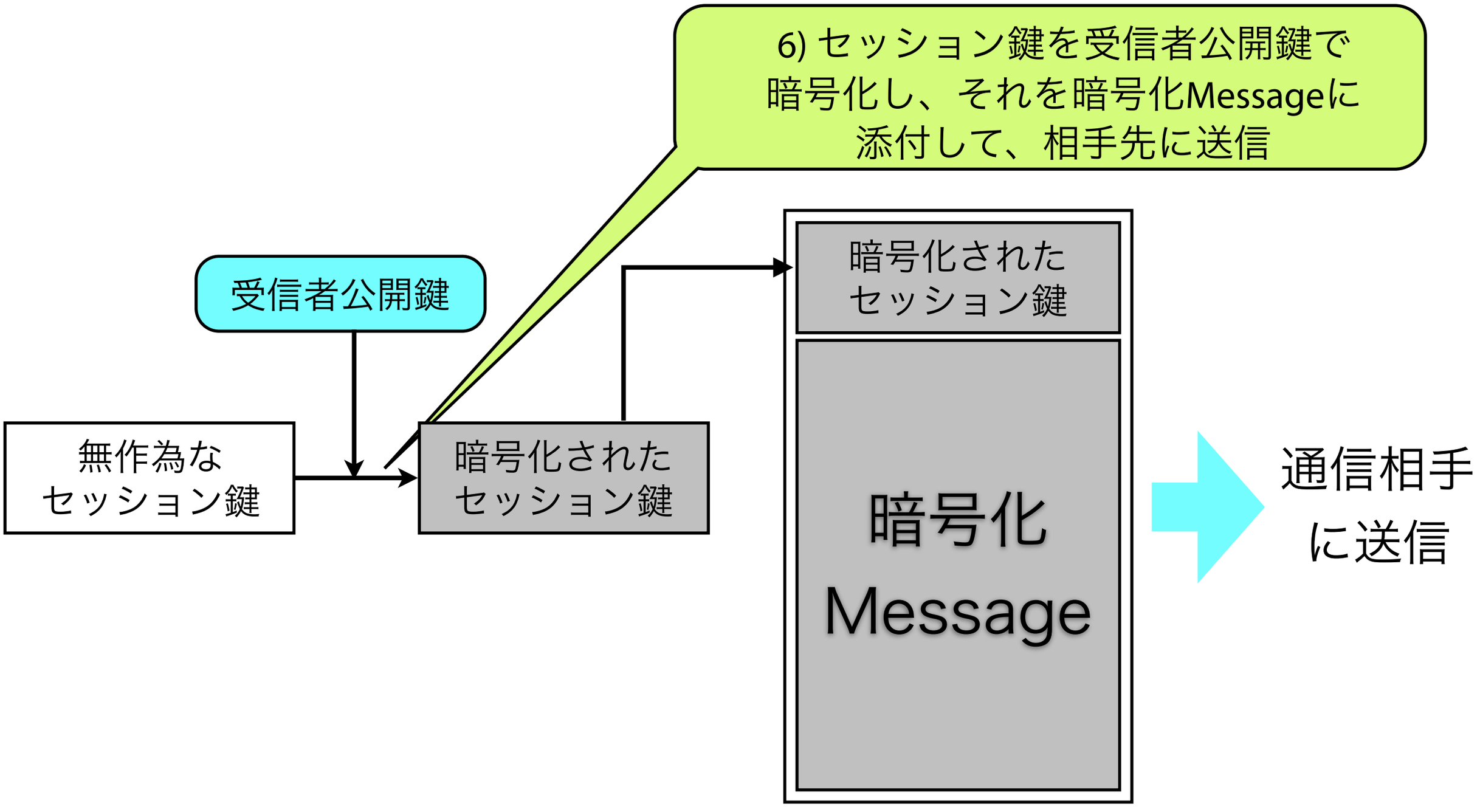


安全なMessage交換 (送信側)

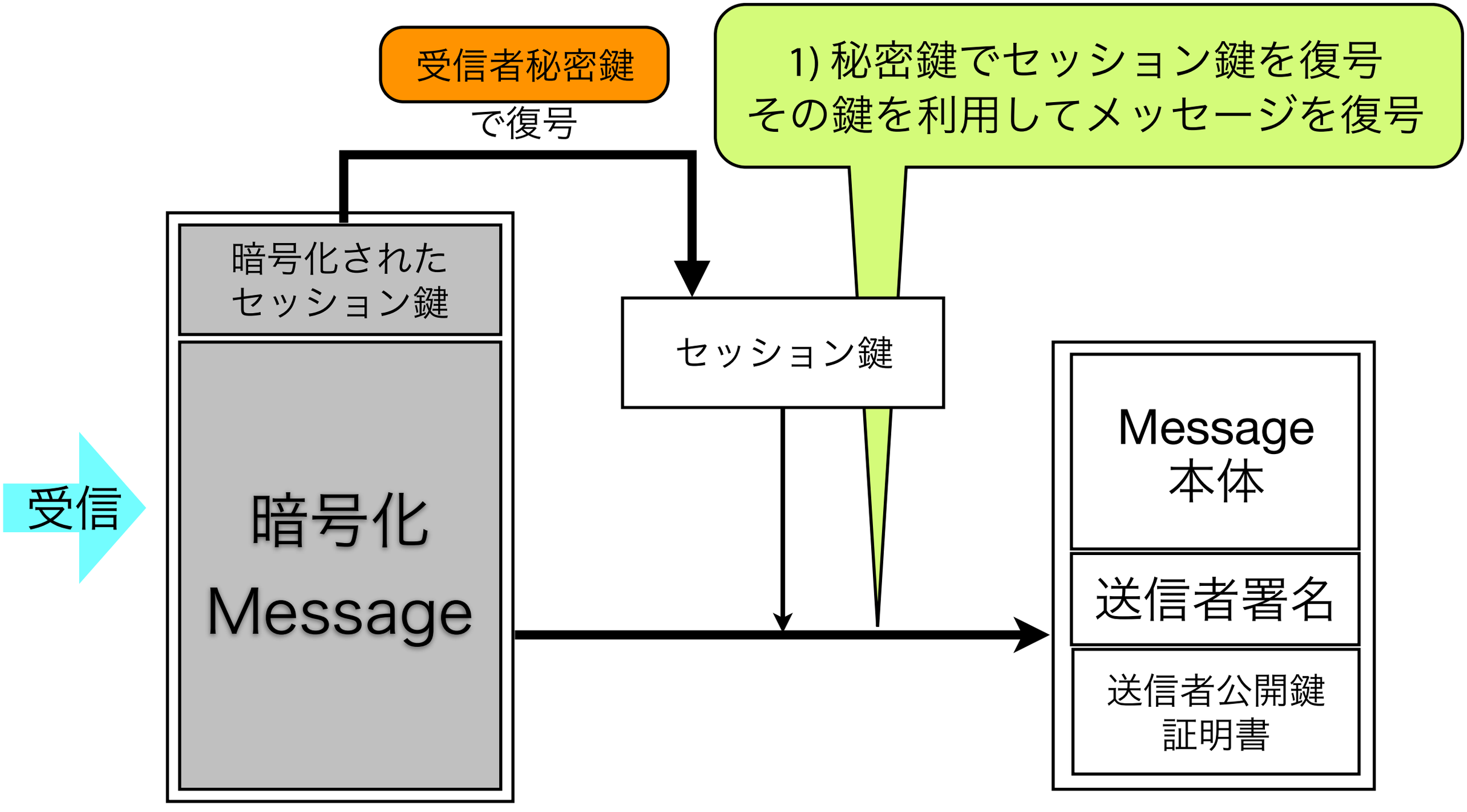
安全なMessage交換 (送信側)



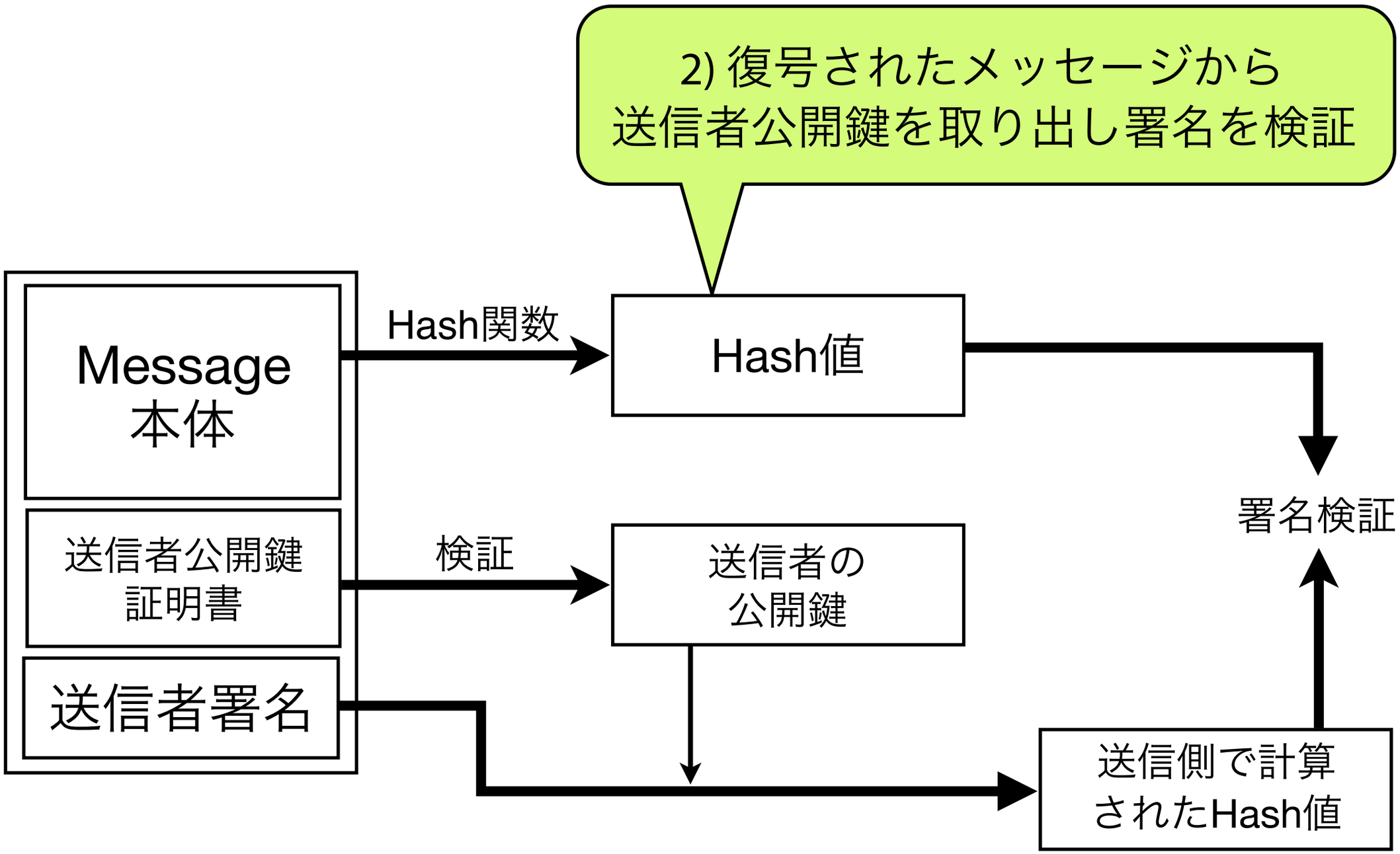
安全なMessage交換 (送信側)



安全なMessage交換 (受信側)



安全なMessage交換 (受信側)



データ暗号から通信路暗号へ

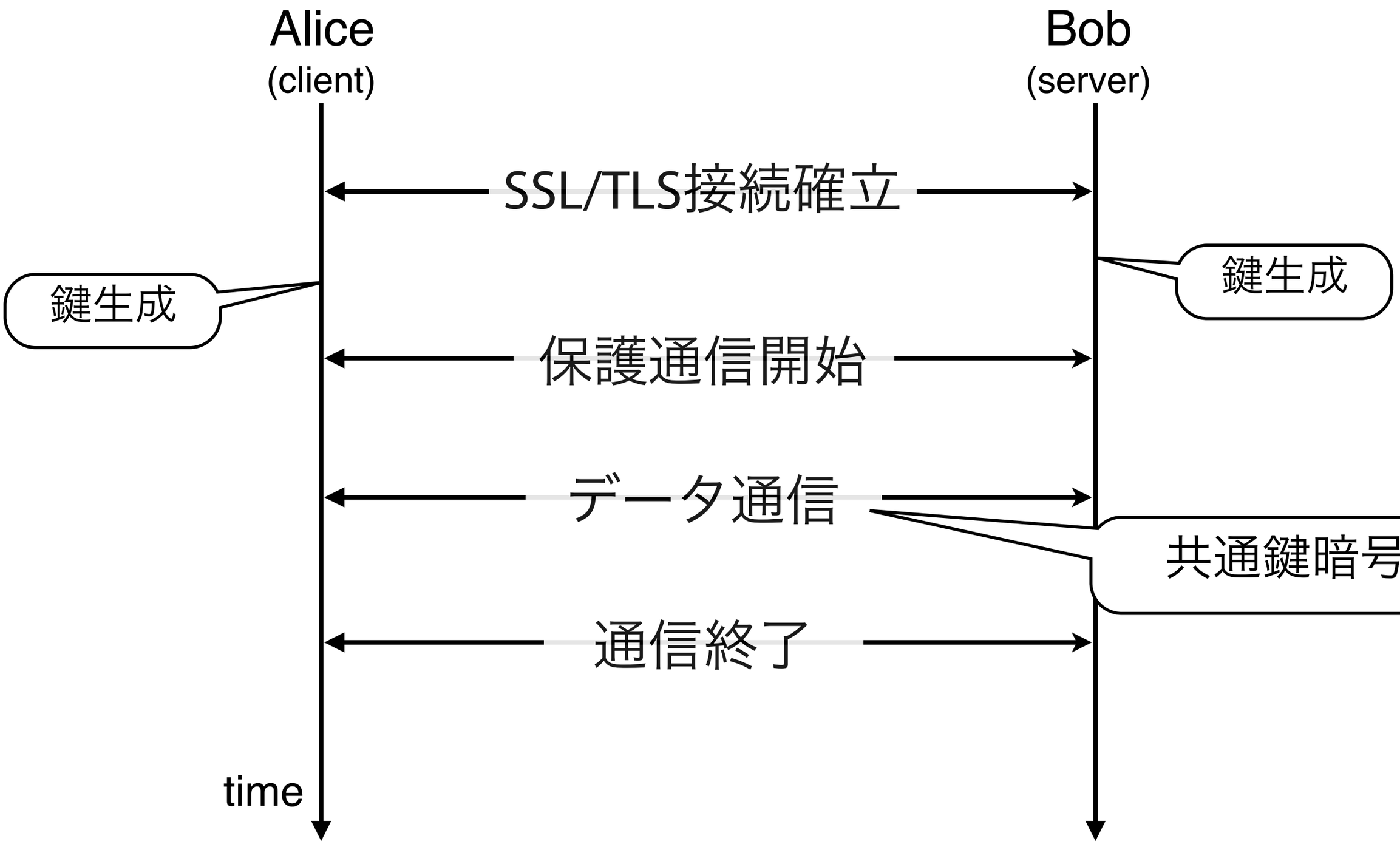
- 前述の手法はバッチ処理に対応
  - ⇒ 送信データが送信処理前に「揃っている」 (E-mail)
  - 送信先も事前に決定済み
- ⇔ 任意のデータを交換する「通信路」化の課題
  - 任意の通信相手を対象に
  - 通信路の確立と鍵の共有
  - 任意の量のデータを転送
  - 通信終了処理

通信路化のために

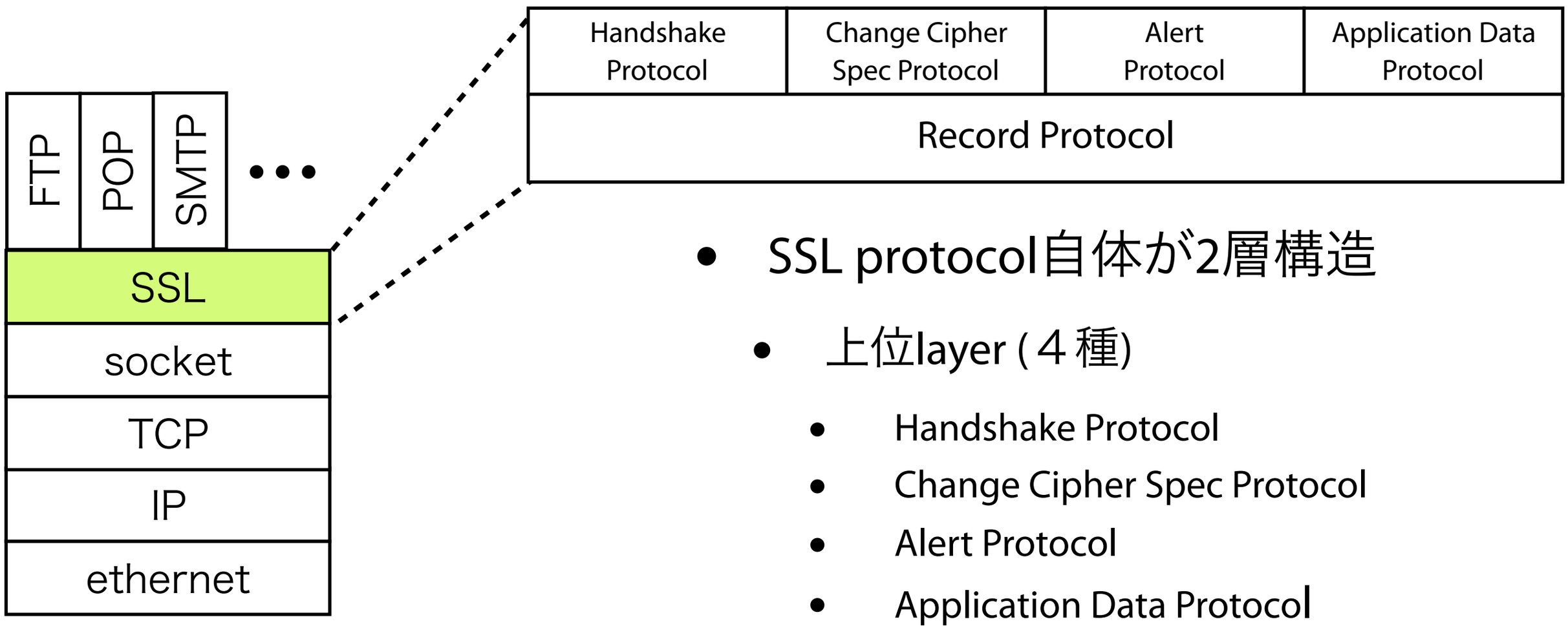
- ハンドシェイク(Handshake)
  - 相互認証、秘密の共有(=鍵生成の材料)
- 鍵の生成
  - 共有秘密から鍵を生成
- データ転送
  - データをレコードに分割、個々のレコードを保護
- 接続終了
  - 専用Msg.により通信を安全に終了.
  - 偽装Msg.により転送中のデータを切り捨てられる危険を防止

Msg. = Message

SSL/TLSの概要

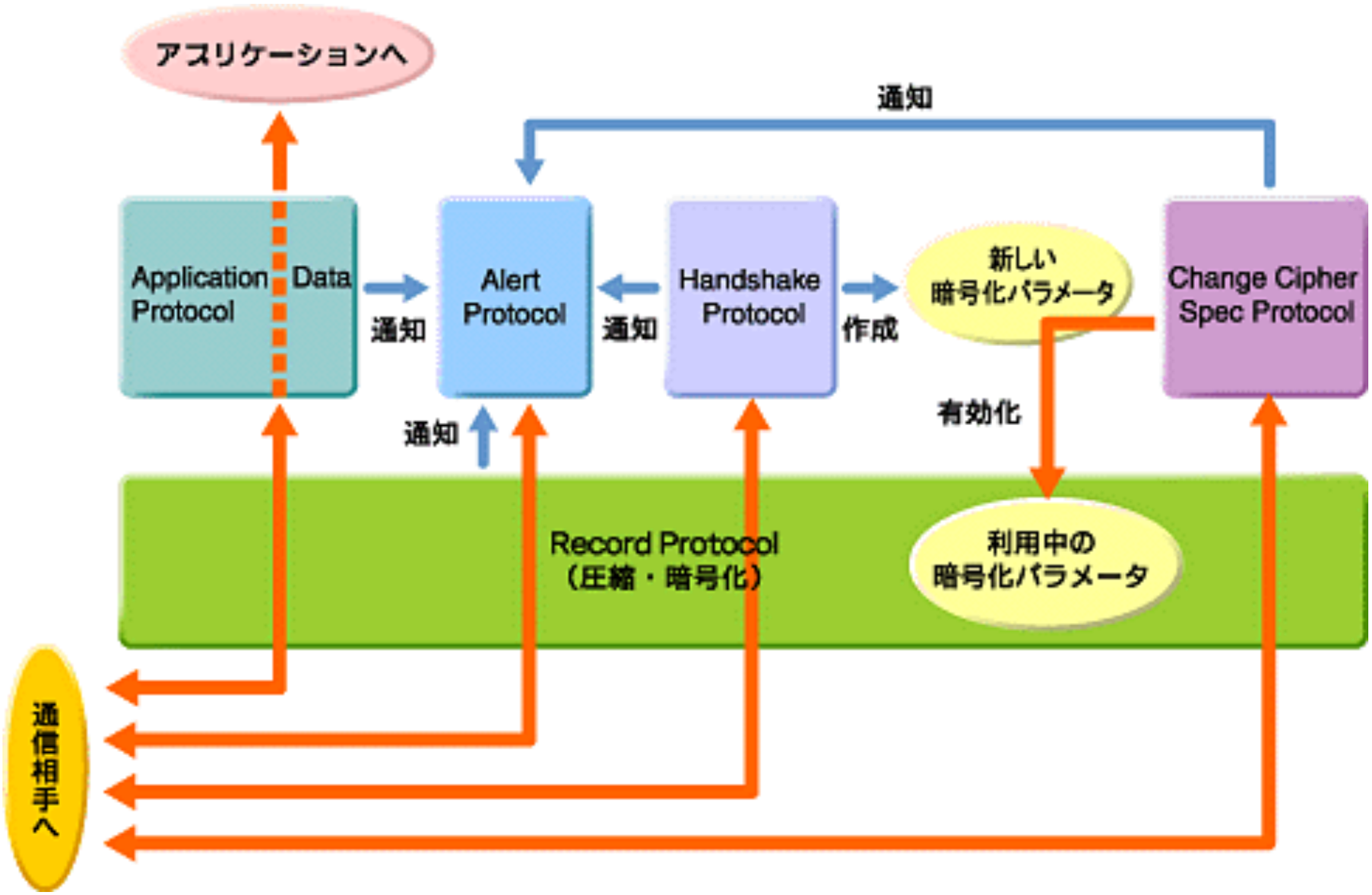

Alice
(client)
Bob
(server)
SSL/TLS接続確立
鍵生成
鍵生成
保護通信開始
データ通信
共通鍵暗号
通信終了
time

SSL プロトコル層



- SSL protocol自体が2層構造
- 上位layer (４種)
  - Handshake Protocol
  - Change Cipher Spec Protocol
  - Alert Protocol
  - Application Data Protocol
- 下位layer
  - Record Protocol

5つの内部Protocol間の関係



図引用: http://www.atmarkit.co.jp/fnetwork/rensai/cell02/ssl3.html