# PKI 普及後の展望

古川 潤
日本電気株式会社
j-furukawa@ay.jp.nec.com

1

# 本チュートリアルの構成

- PKI
  - PKI の概要
  - PKI 普及の要件
  - PKIの問題と解決方法
- グループ署名
  - 基本的な暗号技術の紹介
- 放送型暗号、不正者追跡可能暗号、不正者失効可能放送型暗号
- ID-based 暗号、階層型 ID-based 暗号
- Forward Secure 暗号/署名、Key-Insulated 暗号/署名等
- ミックスネット
- ペアリング
- 展望

2

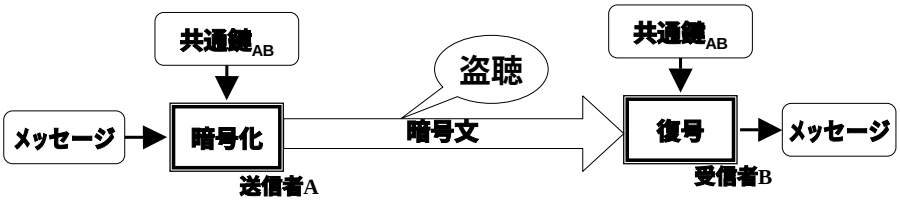
# PKI

- 共通鍵暗号と公開鍵暗号
- 公開鍵暗号の利点
- 公開鍵暗号の問題
- 電子署名とその問題
- 公開鍵暗号と署名の不足点
- PKIによる補強
- PKI普及の要件
- PKIの問題と解決方法
  - 様々な暗号プロトコルの使用

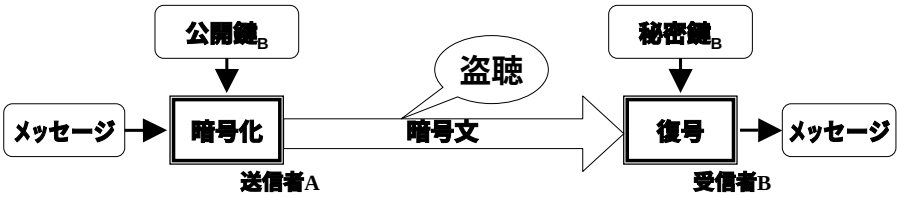
3

# 共通鍵暗号と公開鍵暗号

- **共通鍵暗号**：同じ鍵を送信者と受信者で共有




- **公開鍵暗号**：送信者は受信者の公開鍵で暗号化する



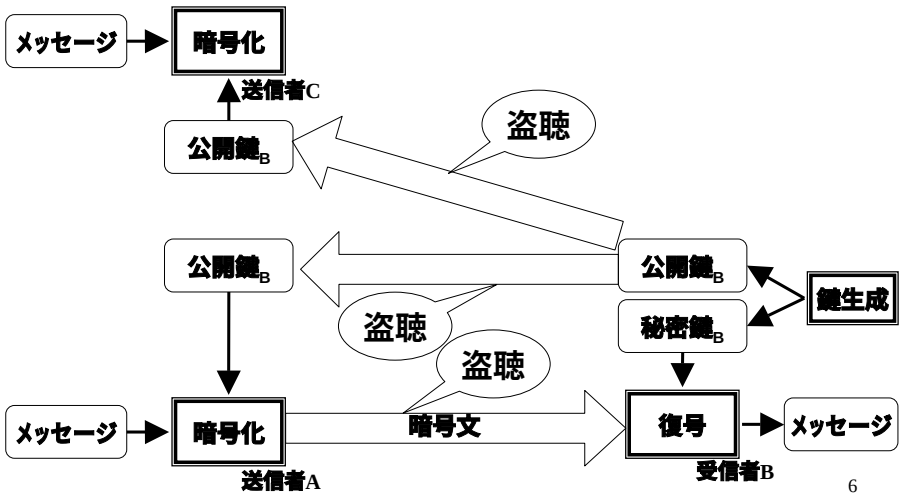

4

# 公開鍵暗号の利点

- 共通鍵暗号の問題点
  - 共通鍵暗号では、互いに通信する二人が予め同じ鍵を共有する必要がある。そもそも、両者の通信が**盗聴**される可能性がある場合に、鍵を送るなどして共有できない。
  - 通信する相手の数だけ異なる鍵を準備する必要がある。
- 公開鍵暗号の利点
  - 暗号化に使う鍵、すなわち公開鍵は、秘密にする必要はなく、秘密鍵のみを秘密にしておけばよい。ゆえに、受信者は自身の公開鍵を生成した後、これを送信者に盗聴可能な通信路を用いて送れば十分な**気がする**。
  - 各自は、秘密鍵と公開鍵の組を一組準備すれば十分。

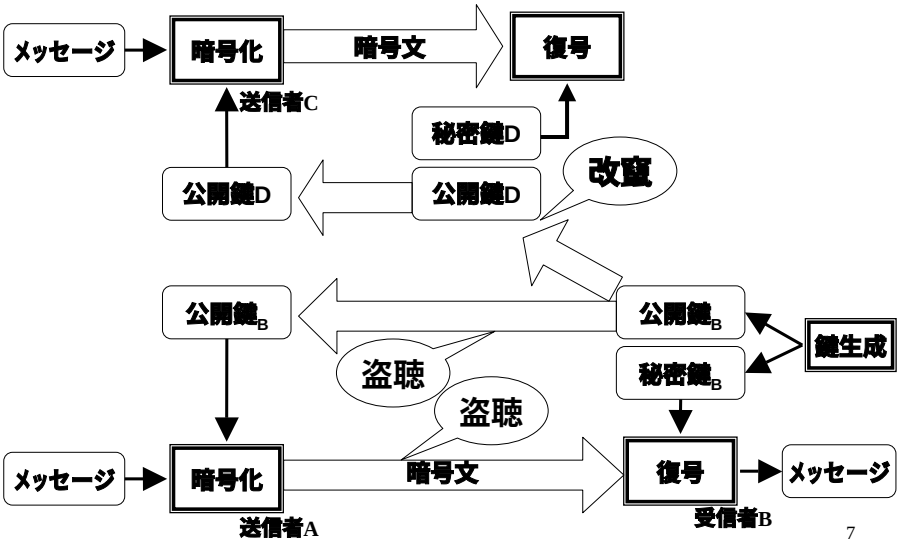
5

# 公開鍵暗号の問題




6

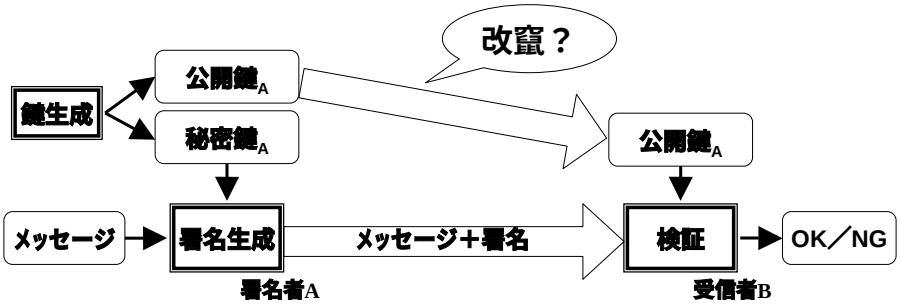
# 公開鍵暗号の問題




7

# 電子署名とその問題

- 電子署名：秘密鍵を用いてメッセージに対する署名を生成できる。この署名は、対応する公開鍵で検証可能。検証を通る署名は、秘密鍵がなければ生成できないので、本人が生成したことが保証される**気がする**。しかし、通信内容を改竄されると弱い。




8

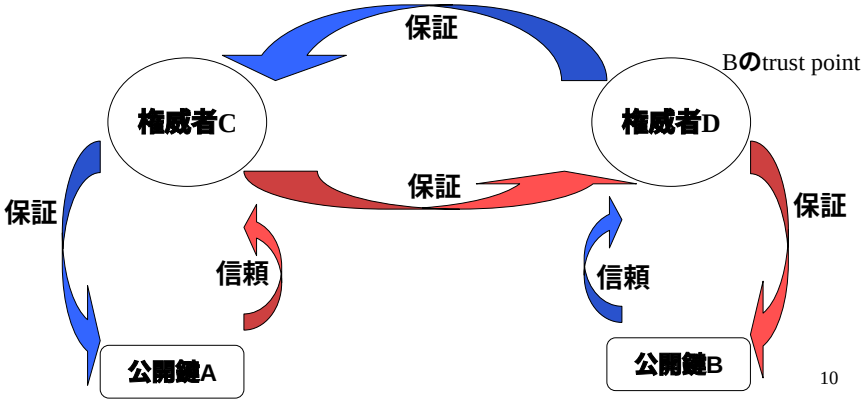
# 公開鍵暗号と署名の不足点

- 送信者が使う受信者の暗号用の公開鍵が、本当に受信者のものであれば、通信の秘密は保たれる。
- 署名検証者の使う署名用の公開鍵が、本当に署名者のものであれば、署名の検証結果を信じることができる。
- いずれの方式も、そもそも公開鍵が特定の人物あるいは機器、組織等と、対応付けられていることを信じるに足る根拠がなければ、信頼することができない。
- 公開鍵を信じるに足るものとすることを助けるのが公開鍵基盤PKI。

9

# PKIによる保証

- 権威者が信頼できるとする公開鍵に電子署名を行うことで、信頼を保証する。これ逐次繰り返すことにより、信頼の鎖を作る。利用者は自身の信頼する権威者から伸びる信頼の鎖に連なる公開鍵を**信じる**。




10

# PKI 普及の要件

- 未だ末端のネットワーク利用者では、PKIで保証された自身の公開鍵を保持している場合は少ない。
- 利益主導による普及
  - PKI を用いると、厳重な本人認証が必要なサービスを受けられる。それゆえ、サービスの拡充がPKI普及の鍵かもしれない。しかし、サービスの拡充とPKIの普及の両者の遅れが互いに他者の前進を阻んでいる可能性がある。
- 攻撃主導による普及
  - 一方、ネットワーク上では、発信者を特定することが困難であることを悪用して、スパム等が散見される。
  - このような不届きなアクセスが甚大な量となると、本人を認証できない限りサービス提供はおろか、さらなるアクセスをも拒む必要が生まれてくる。このような攻撃、乱用がPKI普及の鍵となる可能性もある。
- 利益主導よりも、攻撃主導による普及の方が、セキュリティシステムの普及の形として自然。

11

# PKIの問題と解法(1/2)

- 攻撃が主導してPKIの常時使用が**必須**となるようになった場合、すなわち、PKIを利用した認証や暗号化を行わないネットワークの使用が事実上不可能となった場合の問題点とそれらの解法を列挙する
- 主要なネットワーク活動において認証が必須となると、これらの認証情報を集めて個人の活動を調べることができ、**匿名性**が失われる。
  - **グループ署名**
- メーリングリスト、テレビ会議、有料放送等**多人数**が参加するプロトコルでは暗号や署名の扱いが煩雑。
  - **放送型暗号**
  - 不正者追跡可能暗号
  - 不正者失効可能放送型暗号

12

# PKIの問題と解法(2/2)

- 通信をする場合必ず相手の公開鍵および公開鍵証明書を入手する必要があると不便。紙に書かれたe-mailアドレスにメールを送るのにも苦労する。
  - **ID-based 暗号、階層型 ID-based 暗号**
- 個人の秘密鍵が頻繁に使われ、その重要性が増すと、漏洩や紛失の機会が増え、**漏洩紛失**が致命的になってくる。
  - Forward secure 暗号/署名/放送型暗号
  - Key insulated 暗号/署名
  - 鍵交換
- PKI だけで、ネットワーク上の活動が簡単になるわけではない。電子的な**投票**、入札、支払い等には問題が残る。
  - **ミックスネット**

13

# グループ署名

- グループ署名はネットワーク上のサービスを匿名で受けることを可能にする。

14

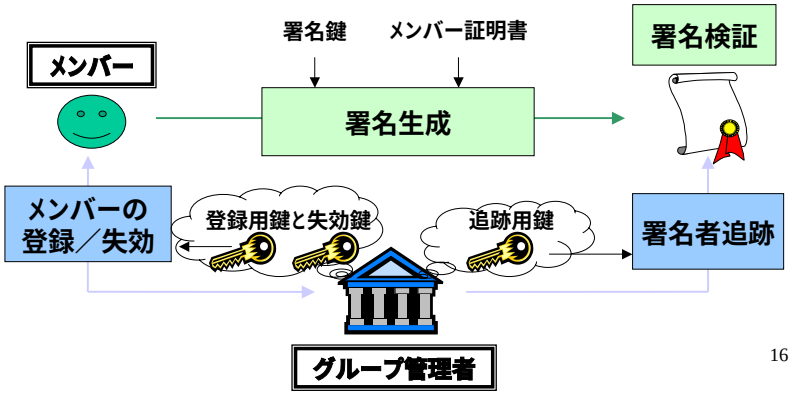
# グループ署名

- グループ署名
- グループ署名のモデル詳細
  - メンバー登録、署名生成と検証、署名者追跡、追跡失効
- グループ署名の応用例
- 匿名での活動を許すPKI
- グループ署名の歴史
- グループ署名の構成例
  - **基本的な暗号技術の紹介**
  - グループ署名の一般的な構成例
- 効率的なグループ署名, 構成例
- メンバー失効
  - アキュミュレーター
- グループ署名の課題
  - 階層型グループ署名

15

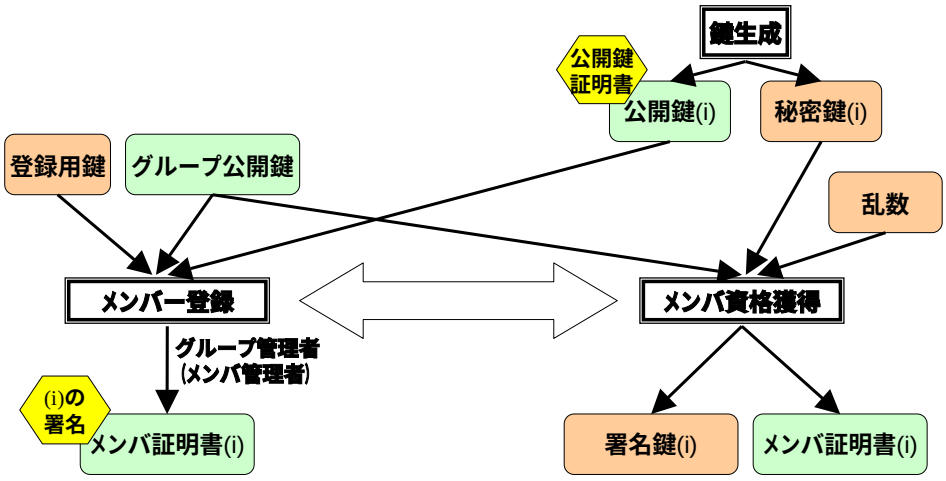
# グループ署名

- グループの**メンバーのみがグループ署名を生成**できる
- 一般の利用者は、この署名から署名者個人を**特定することは不可能**
- グループ管理者は利用者をグループのメンバーに**登録**することができる。
- グループ管理者は、グループ署名から署名者個人を**追跡**し特定することができる。
- グループ管理者は、メンバーをグループから追放 (**失効**) することができる



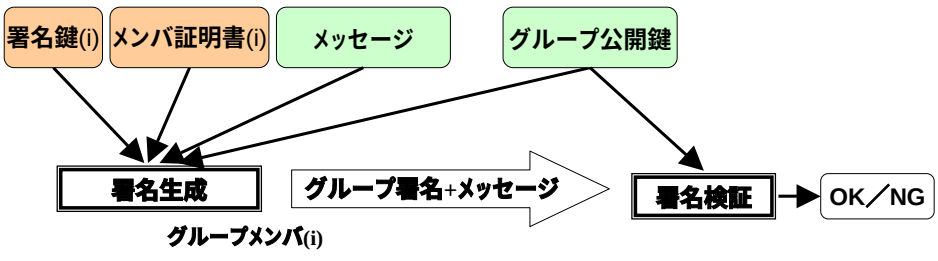

16

# グループ署名のモデル詳細 (メンバー登録)



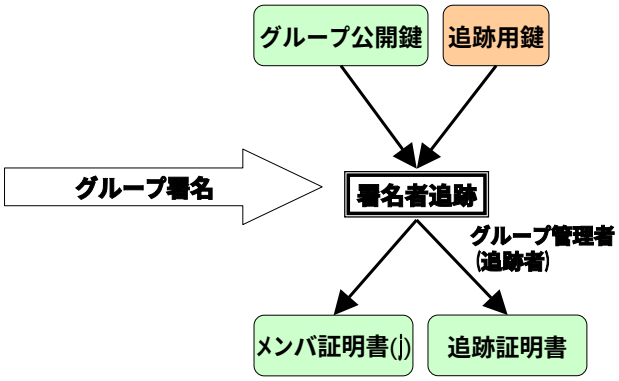

17

# グループ署名のモデル詳細 (署名生成と検証)



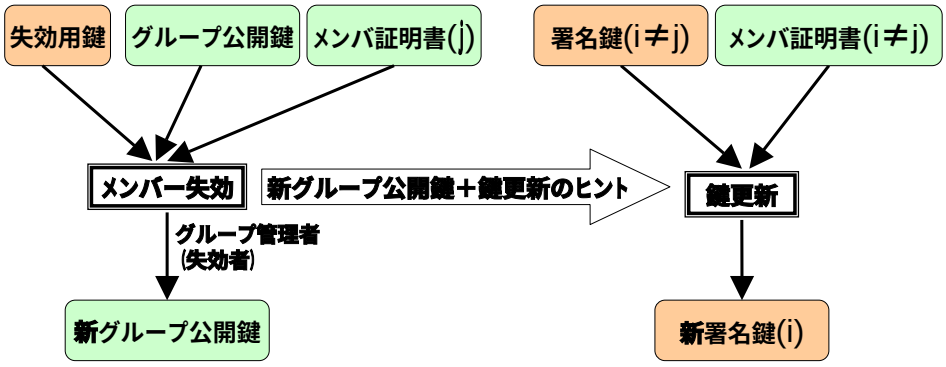

18

# グループ署名のモデル詳細
# (署名者追跡)




19

# グループ署名のモデル詳細
# (メンバー失効)




20

# グループ署名の応用例

- グループとして何を選ぶかで様々な応用が考えられる
  - 特定の権限を持つメンバのグループ
    - アクセス制限、
  - 特定の契約をしてメンバのグループ
    - 契約者のみが匿名でサービスを受けることができる
  - 特定の団体のメンバのグループ
    - 企業が、複数の社員に署名権限を与える一方、その署名から社内の組織構造を隠す。
  - その他のグループ
    - 年齢、性別、職業、国籍などのグループ
    - 非成年被後見人、非犯罪者、非税滞納者のグループ
    - PKI により認証された主体のグループ

21

# 匿名での活動を許すPKI

- グループ署名方式で、PKI で証明書が付いた公開鍵のグループを考える。
  - 各公開鍵の保有者は、併せて、グループ署名の署名鍵とメンバ証明書を保持する。この署名鍵とメンバ証明書を用いて生成したグループ署名から、グループ管理者は上記公開鍵を追跡できる。
  - 匿名性を求めるネットワーク活動、通常は個人の特定を求めないネットワーク活動では、グループ署名を用いて認証を受ける。例えば、一般的サイトの閲覧、検索活動等である。
  - サービス提供者はログとしてグループ署名を保存する。このデータからグループ管理者以外には個人を特定できないため、サービス提供者を信用する必要はない。また、サービス提供者にとってもログの管理は比較的容易である。
  - 利用者が、犯罪行為にかかわる、DoS 攻撃を行う、スパムメールを発信するなど、サービスの乱用を行った場合、グループ管理者ログとして保存されているグループ署名から個人を特定し、場合によっては該当するグループメンバーを失効する。

22

# グループ署名の歴史

| 提案年 | 提案者 | 署名長(bit) | 計算量比 |
|---|---|---|---|
| 1971 | Chaum, Heyst | | |
| 2000 | Ateniese, Camenisch, Joye, Tsudik | 23,709 | 82.5 |
| 2004 | Boneh, Boyen, Shacham | 2,057 (変種) | 1.1 |
| 2004 | Camenisch, Lysyanskaya | 5,926 | 2.3 |
| 2004 | Camenisch, Groth | 5,216 | 1.3 |
| 2004 | Nguyen, Safavi-Naini | 4,782 | 1.2 |
| 2005 | Furukawa, Imai | 1,704 | 1 |

- 通常の暗号や署名方式と同程度の効率をもつグループ署名方式が提案されたのは最近のこと。

23

# グループ署名の構成例

- 基本的な暗号技術の紹介
  - 公開鍵暗号
    - 確率暗号
    - 暗号の識別不可能性
    - ElGamal 暗号
    - 適応的選択暗号文攻撃
  - 署名
    - 適応的選択文書攻撃
  - 零知識対話証明
    - 対話証明
    - 零知識対話証明
    - 知識の証明
    - ランダムオラクルモデル
    - Fiat-Shamir 変換
  - 署名の例、暗号の例
- グループ署名の一般的な構成例

24

# 公開鍵暗号

- 公開鍵暗号は次の3個の（確率的多項式時間）アルゴリズムからなる
- 鍵生成：公開鍵と秘密鍵を生成する。
- 暗号化：メッセージと公開鍵が入力され、暗号文を出力する。
- 復号：暗号文と秘密鍵が入力され、メッセージを出力する。

- 鍵生成アルゴリズムには**セキュリティ変数**が入力されるが、この変数に関する議論は以降一切省略する。なお、「**無視できる**」、「多項式時間」という言葉はこの変数によって定義される。

25

# 確率暗号

- 公開鍵暗号において、同一のメッセージに対して一意的に暗号文が対応するとする。この様な暗号では、メッセージの候補が限られているならば、それらを順番に公開鍵で暗号化し、懸案の暗号文と一致するものを見つければ解読できる ⇒ 弱い暗号。
- 確率暗号では、一つのメッセージに対応する暗号文が大量にあり、暗号化時にそのうちの一つが確率的に選ばれる。強い安全性を持つ公開鍵暗号は確率暗号でなければならない。

26

# 暗号の識別不可能性

- 暗号が識別不可能であるとは、およそ、どのような（確率多項式時間）攻撃者A、いかなるメッセージの対 $(m_0,m_1)$ に対しても、$m_0$ の暗号文 $enc(m_0)$ が与えられた攻撃者が 1 を出力する確率と、$m_1$ の暗号文 $enc(m_1)$ が与えられた攻撃者が 1 を出力する確率の差が無視できる程であること。

$$\Sigma_{m_0,m_1} |Pr[A(m_0, m_1, enc(m_0))=1] - Pr[A(m_0, m_1, enc(m_1))=1]| < neg$$

確率は、鍵生成と攻撃者が使う乱数とに関してとる。

27

# ElGamal 暗号

識別不可能な暗号の例：

- ElGamal暗号
  - **鍵生成:** $p,q$ 大きな素数, $q|p-1$.
    $G_q:=\{h|h \in (Z/pZ)^*, h^q=1 \bmod p\}$,
    $g \in G_q, x \in_R Z/qZ, y=g^x \bmod p$.
    公開鍵$=(g, y)$, 秘密鍵$=x$
  - **暗号化:**メッセージ $m \in G_q$, 乱数 $r \in_R Z/qZ$ ,
    暗号文 $(h, v)=(g^r, m\, y^r) \bmod p$
  - **復号：** $v/h^x = m\, y^r/(g^r)^x = m\, y^r/g^{x\,r} = m\, y^r/y^r = m$

28

# ElGamal 暗号の安全性

- **Diffie-Hellman 判別問題:**
  - *集合* $R = (G_q)^4$
  - *集合* $D = \{(g, y, h, z) \mid \exists x,s \in Z/qZ\ (g,y,h,z)=(g,g^x,g^r,g^{xr})\} \subset R$

  **いかなる(確率的多項式時間)攻撃者も、集合**Rからランダムに選ばれた$(g, y, h, z)$**と集合**Dからランダムに選ばれた$(g, y, h, z)$**を無視できない確率で識別できない。**

- **公開鍵** $(g, y=g^x)$ **と暗号文** $(h,v)$ **が与えられたとき、暗号文が** $m$ **の暗号文であるならば、**$(h, z)=(h, v/m)=(g^r, my^r/m = y^r)$ **となる** $r$ **が存在する。すなわち、**$(g, y, h, z) \in D$**となる**.
- ElGamal **暗号は** Diffie-Hellman **判別問題が難しければ識別不可能である。**

29

# Diffie-Hellman 判別問題

$$\begin{pmatrix} g & \overset{a}{\longrightarrow} & g^a \\ g^b & \overset{a}{\longrightarrow} & g^{ab} \end{pmatrix} \quad , \quad \begin{pmatrix} g & \overset{a}{\longrightarrow} & g^a \\ g^b & \overset{c/b}{\longrightarrow} & g^c \end{pmatrix}$$

30

# 適応的選択暗号文攻撃

- 次のようなゲームを考える
  - 挑戦者は、最初に鍵を生成して公開鍵を攻撃者に与える。秘密鍵は挑戦者が保持する。
  - 攻撃者は任意の暗号文を生成し、挑戦者に復号を要求する。挑戦者はこれを**復号**して結果を攻撃者に与える
  - 攻撃者は、上記復号要求を行うなか、一度だけ二つの平文 $m_0, m_1$ を選び、挑戦者に送る。挑戦者は $b$ を $\{0,1\}$ の中から無作為に選び、$m_b$ の暗号文 $c^*$ を生成して攻撃者に与える。
    - 攻撃者は復号要求を続けるが、$c^*$ の復号だけは要求してはならない。
  - 攻撃者は最後に $b'$ を出力して終了する。
- いかなる（多項式時間）攻撃者に対しても、$b=b'$ である確率と $1/2$ の差が無視できる程であれば、適応的選択暗号文攻撃に対して識別不可能な暗号という。

31

# 署名

- 署名は次の3個の（確率的多項式時間）アルゴリズムからなる
- 鍵生成：公開鍵と秘密鍵を生成する。
- 署名生成：メッセージ、秘密鍵、と公開鍵が入力され、署名を出力する。
- 署名検証：メッセージと公開鍵が入力され、1(受理)または0(不受理)を出力する。

32

# 適応的選択文書攻撃

- 次のようなゲームを考える
  - 挑戦者は、最初に鍵を生成して公開鍵を攻撃者に与える。秘密鍵は挑戦者が保持する。
  - 攻撃者は任意の文書を生成し、挑戦者に署名生成を要求する。挑戦者はこの文書に対する署名を生成して攻撃者に与える。
  - 攻撃者は最後に文書と署名の対 *(m,sig)* を出力する。
    - ただし、この対は、攻撃者が挑戦者に署名を要求した文書とそれに対して返答された署名の対のいかなるものとも、対として一致してはならない。
  - 攻撃者の出力した文書と署名の組が正当なものと検証されれば攻撃に成功したとする。
- いかなる（多項式時間）攻撃者に対しても、上記攻撃が成功する確率が無視できる程であれば、適応的選択文書攻撃に対して存在的偽造不可能な署名という。

33

# 対話証明

- 対話証明
  - およそ、証明者Pと検証者Vが互いに通信することで、PがVにある事実を納得させるプロトコル。
    - 事実が成り立つならば、正直な証明者Pと正直な検証者Vが対話した後、検証者Vは高い確率で 1 (納得)を出力する
    - 事実が成り立たないならば、いかなる不正な証明者P*と対話しても、検証者Vは高い確率で 0 （不納得）を出力する
- つまらない例：組*(g, y, h, z)*が, $(h, z)=(g^r, y^r)$ なる $r$ が存在する組であることの対話証明：
  - 証明者Pは $r$ を検証者Vに送る。
  - 検証者Vは $(h, z)=(g^r, y^r)$ を確認。

34

# 零知識対話証明

- 零知識対話証明
  - 対話証明で、検証者が得られる知識が零であるもの。
- 得られる知識が零であることの定式化。ある事実が成り立つ(集合 L に属する)ことの零知識証明とは、全ての検証者 V* (正直に動くとは限らない)に対して、シミュレータ S が存在して、どのような L の要素 $x$ に対しても、次の二つのデータ識別できないことをいう
  - $x$ が L に属することを、証明者 P が V* に証明した後に、V* が出力するデータ。
  - $x$ を与えられた S が出力するデータ
- これは、V* が P と対話証明したことによって生成できるデータは、高々、S が P と対話をせずとも生成(シミュレーション)できる程度のデータであることを意味する。

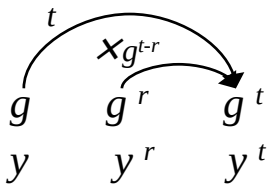
35

# 零知識対話証明の例(1/2)

- 例:組$(g, y, h, z)$が、$(h, z)=(g^r, y^r)$ なる$r$が存在する組(Diffie-Hellman 組)であることの零知識対話証明。
  - 以下の処理を十分多数回繰り返す
    - 証明者乱数 $t$ を選び、$(u, v)=(g^t, y^t)$ を生成し検証者に送る。
    - 検証者はランダムに$c=0$か$c=1$を送り返す。
    - 証明者は$c=0$ なら $s=t$ を、$c=1$ なら $s=t-r$ を検証者に送る。
    - 検証者は、$c=0$なら$(u, v)=(g^s, y^s)=(g^t, y^t)$を、
      $c=1$なら$(u/h, v/z)=(g^s, y^s)=(g^{t-r}, y^{t-r})$ を確認する。

36

# 零知識対話証明の例(2/2)




- **対話証明**: c=0, c=1 **の両方に答えることができるならば高い確率で、** $t$-$r$ **と** $t$ **の両方を知っていて、** $g^{t-r} = u/h$, $y^{t-r} = v/z$, $g^t = u$, $y^t = v$ **が成り立たねばならない。**
  **よって、** $h = u/(u/h) = g^t/g^{t-r} = g^r$, $z = v/(v/z) = y^t/y^{t-r} = y^r$ **が成り立たないと高い確率で検証者を納得させることができない。**
- **零知識性**: **各** c **の値の予想ができれば、** $(u, v)=(g^s, y^s)$ **あるいは、** $(u,v)=(g^s h, y^s z)$ **と生成すれば、** $r$ **の知識がなくとも検証が納得する対話を行える。検証者をブラックボックスと見て、何度もリセットし同じ入力を与える(巻き戻す)ことで、巻き戻されない検証者が得ることができる履歴と識別できない履歴を生成(シミュレーション)可能。すなわち、検証者が得た対話履歴は、証明者と対話せずとも、そもそも自身で生成可能な履歴であったため、この対話から知識(**$r$**など)を得ることはない。**

37

# 応用例

- **零知識対話証明は、認証プロトコルに応用できる。**
- **ランダムに与えられた組** $(g, y, h, z)$ **に対して,** $(h, z)=(g^r, y^r)$ **なる** $r$ **が存在するかどうかを知ること(**Diffie-Hellman **判別問題)は難しい。一方、** $r$ **の知識があればこれを知ることは簡単である。**
- **上記組**$(g, y, h, z)$**を公開鍵、** $r$ **を秘密鍵とする。認証プロトコルとして、公開鍵に対して、上記** $r$ **が存在することの零知識対話証明を行う。**
  - Diffie-Hellman**判別問題の難しさから、秘密鍵所持者以外が零知識対話証明で検証者を納得させるのは難しい。**
  - **零知識対話証明はなんら知識を検証者に与えないので、検証者はこの対話に参加した以降も、証明者に成りすますことはできない。**(**** $r$ **を示すつまらない対話証明であった場合、検証者は** $r$ **を用いて、以降、証明者に成りすますことができる。**)

38

# 知識の対話証明

- 先の例では、$r$ の存在を示すのみならず、証明者から $r$ を抽出することも可能であった。およそ、このように証明者から事実を示す証拠を抽出できる対話証明を知識の対話証明と呼ぶ
- 知識の対話証明の例
  - 組 $(g, y)$ に対して、$y=g^r \bmod p$ を満たす $r$ の知識の証明。$g, y$ が同じ群の元である場合は、このような $r$ が存在することは明らかなので、証明するに及ばないが、証明者が $r$ の知識を持つかは別である。先の零知識対話証明とほぼ同様のプロトコルを考えると、やはり同様に証明者から $r$ を抽出することができるので、このプロトコルは零知識な知識の証明となる。

39

# 正直検証者零知識

- 零知識対話証明では、検証者に悪意があり正しくプロトコルに従わない場合も考慮していた。検証者が正直に振舞う場合に零知識性が保たれる場合、正直検証者零知識と呼ぶ。
- 正直検証者零知識な知識の対話証明の例：
  - 対 $(g, y,)$ に対して $y = g^r \bmod p$ なる $r$ の知識を証明する。
    - 証明者乱数 $t$ を選び、$u = y^t \bmod p$ を生成し検証者に送る。
    - 検証者はランダムに$c \in Z/qZ$を選び、送り返す。
    - 証明者は $s = cr+t \bmod q$ を送る。
    - 検証者は、$g^s = y^c u \bmod p$ を確認する。
  - 知識の証明：証明者は異なるcに対しても高い確率で返答するので、同じ$t$ を用いて二回証明させると

    $s' = c'r+t$ , $s = cr+t$ が得られる。$r=(s-s')/(c'-c)$
  - 零知識: $s, c$ をランダムに選び、$u = g^s y^{-c}$ とシミュレートする。

40

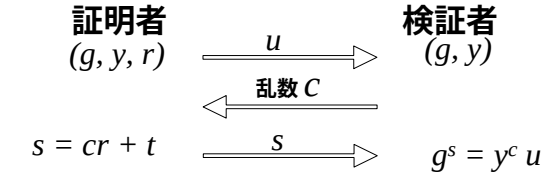
# ランダムオラクルモデル

- ランダムオラクルは、任意の文字列が入力されると、ある定められた地域（例えばZ/qZ）の値を出力するオラクルで、次の様なもの。
  - 入力が初めての値であったならば、その値域からランダムに値を選んで出力する。
  - 新たな入力が過去に入力された値と同じであれば、その時に出力した値と同じ値を出力する。
- ランダムオラクルは、文字列を入力してみなければその出力が全く予期できないハッシュ関数をモデルしている。
- 「ランダムオラクルモデルで」と言う場合、ハッシュ関数をランダムオラクルに置き換えた場合を考えている。

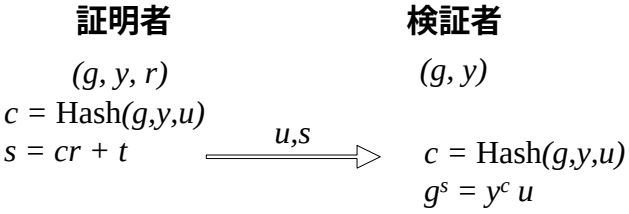
41

# Fiat-Shamir 変換

- 先の正直検証者零知識な知識の対話証明は次の動作をする。




- そこで、検証者に乱数 $c$ を請う代わりに、暗号学的ハッシュ関数の出力で代用すると、ランダムオラクルモデルで、非対話に知識の証明が行える。




42

- この手法を用いると、非対話零知識証明も可能となる

# 署名への応用例

- Fiat-Shamir 変換は適応的選択文書攻撃に対して存在的偽造不可能な署名の構成に応用できる。
- ランダムに与えられた組 $(g, y)$ に対して $y = g^r$ なる $r$ を求めること(離散対数問題)は、難しい。上記組 $(g, y)$ を公開鍵、$r$ を秘密鍵とする。メッセージ $m$ に対する署名として、$m$ と関係付けられた、$r$ の知識の証明のFiat-Shamir 変換を生成する。
- $m$ に対する署名は、$(s, u)=(r\ Hash(g, y, g^t, m)+t, g^t)$.
  - $m$ がハッシュ関数の入力に含まれているところが差分。
  - 離散問題の難しさから、秘密鍵所持者以外がこの様な署名を生成するのは難しい。
  - 正直零知識な知識の証明をFiat-Shamir変換したものなので、およそ、攻撃者は署名から新たな署名の偽造に使える知識を得ることはない。

43

# 暗号への応用例

- Fiat-Shamir 変換は適応的選択暗号文攻撃に対して識別不可能な暗号の構成に応用できる。
- 鍵生成:秘密鍵を$x$、公開鍵$(g, y=g^x)$
- 暗号化:メッセージ $m$, と公開鍵 $(g, y)$ が与えらる。$r,s \in Z/qZ$を選び、

  $z=m\ y^r$, $u = g^r \mod p$, $w = g^s \mod p$, $g'=Hash\ (z, u, w)$,
  $u'= g'^r \mod p$ $w'= g'^s \mod p$, $c =Hash'(g',u',w')$
  $s = s+rc \mod q$

  暗号文は、$(z, u, w, u', w', s)$
- 復号:$g'= Hash(z, u, w)$, $e = Hash'(g', u', w')$を生成し、
  $g^s = u^c w \mod p$, $g'^s = u'^c w' \mod p$ が成り立てば、
  $m = z / u^x$ を復号結果とする。

44

# グループ署名の構成例

- **セットアップ**：グループ管理者は署名用と暗号用にそれぞれ公開鍵秘密鍵ペア *(pk(s), sk(s)), (pk(e), sk(e))* を準備し、
  - *(pk(s), pk(e))* を公開。
- **メンバ登録**：メンバー*(i)*は署名用の公開鍵秘密鍵ペア *(pk(i), sk(i))* を生成。*pk(i)* に、PKIで証明された公開鍵 *(pk'(i))* で検証可能な署名 *sig'(i)* を生成し、グループ管理者に送る。グループ管理者は、*pk(i)* に *sk(s)* で署名 *sig(s,i)* を打ち、メンバーに送る。
  - 署名鍵は *sk(i)*、メンバー証明書は*(pk(i), sig(s,i)).*
- **署名生成**：メッセージ*m*の*sk(i)*による署名*sig(i,m)*を生成。
  ***次に***、*(i, pk(i), sig(s,i), sig(i,m))*の*pk(e)*による暗号文 *enc(misc)*を生成し、上記処理が行われたことの非対話零知識証明 *nizk(misc)* を生成。
  - *(enc(misc), nizk(misc))* を *m* に対するグループ署名とする。
- 署名検証：enc(misc)に対するnizk(misc) をpk(s)を用いて検証する。
- **追跡**：*sk(e)* を用いて *enc(misc)* を復号すると *(i, sig(i,m))* が現れ、*i* 番目のメンバが署名したことが証明される。

45

# 効率的なグループ署名の構成

- 前述の一般的なグループ署名の構成では *nizk(misc)* が非常に非効率。上記構成例と同等のことができる効率的な特殊例を見つけることが重要。
- 様々な方式が提案されているが、現在のところペアリングを利用したものが最も効率的である。

46

# 効率的な方式の例(Furukawa-Imai)

- いい加減な記述:
  - 登録用鍵: $w \in Z/pZ$
  - 追跡用鍵: $(s, t) \in (Z/pZ)^2$
  - グループ公開鍵: $(y, e, f) = (g_2^w, g^s, g^t) \in G_2 \times G^2$
  - 署名鍵: $(x, b, a) \in (Z/pZ)^2 \times G_2$
  - メンバー証明書: $(z, g^x) \in (Z/pZ)^3$
- これらは次の式を満たす。

$$a^{(w+b)} = g_1 h^x k^z$$

- 署名は以下を満たす、$(x, b, a, z, r)$ の零知識証明を、$m$ を入れて Fiat-Shamir 変換したもの。

$$a^{(w+b)} = g_1 h^x k^z$$

$$u = g^x g^r, \quad v = e^r, w = f^r$$

47

# メンバー失効

- グループ署名は匿名で署名をするので、失効者リストの公表ではメンバーの資格を失効できない。
- メンバーが失効する度にグループ公開鍵を全く別のものに変え、新たに各メンバーに新しいメンバー証明書を発行する方式では、実際の運営には使えない。
- 2002年Camenisch, Lysyanskaya が提案したアキュミュレータを用いたメンバー失効方式では、グループ管理者のすべきことは、グループ公開鍵を更新して、この公開鍵と小さいデータである鍵更新のヒントを公表するだけである。各メンバーは、公表された新しいグループ公開鍵とヒントと、自身のメンバ証明書と署名鍵から、新たな署名鍵を生成できる。失効されたメンバーはこの処理を行うことが不可能になっている。

48

# アキュミュレータ

- セットアップ：安全な素数 $p, q, n=pq, u \in QR_n$ を選び、$(n, u)$ を公開。
- メンバー$(i)$の追加：ある範囲の大きさの素数 $p(i)$ を選び、$w(i)=u^{1/p(i)} \bmod n$ を生成し、$(p(i), w(i))$をメンバーに渡す。
- メンバーの証明：メンバーは、$u$ に対して

$$w(i)^{p(i)}=u \bmod n$$

なる $(w(i), p(i))$ を保持することを零知識で知識の証明することがメンバーである証明として付加的に要求される。

- メンバー$(i)$の失効：$u \rightarrow w(i)$ と更新し、更新のヒントを $p(i)$ として公表する。
- 各メンバ$(j \neq i)$は、新しい $u$ に対して、$w'(j)^{p(j)}=u=w(i)$となる $w'(j)$を次のようにして生成する。

$a\ p(i)+b\ p(j)=1$ なる$a, b$を生成し、

$w'(j)=w(j)^a\ w(i)^b$



# アキュミュレータ

- $w'(j)=w(j)^a\ w(i)^b$

$= u^{a/p(j)} u^{b/p(i)}$

$= u^{a/p(j)+b/p(i)}$

$= u^{a\ p(i)\ /\ p(i)p(j)\ +\ b\ p(j)\ /\ p(i)p(j)}$

$=u^{(a\ p(i)+b\ p(j))\ /\ p(i)p(j)}$

$=u^{1\ /\ p(i)p(j)}$ ($a\ p(i)+b\ p(j)=1$ より)

$= w(i)^{1\ /\ p(j)}$

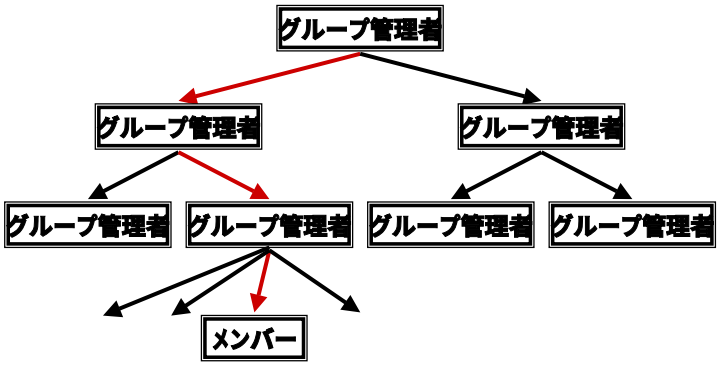
- 各メンバー$(j, j \neq i)$は、ヒントを用いて自分で自分の秘密を更新できる。しかし、メンバー$(i)$は更新できない。

# グループ署名の課題

- メンバの階層的管理ができないと、PKI と融合しにくい。
  - PKI では、各 certificate authority が独自に公開鍵証明書を発行することができるが、グループ署名方式においては単一のグループにメンバーを加えることのできるグループ管理者はただ一人しか存在しない。各 certificate authority がメンバを加えることができるようにすると、現在の方式では、各 certificate authority ごとのグループを形成するしかない。この場合、匿名性が若干下がる。
- 階層的グループ署名。
  - 提案されている方式はあるが、通常の使用に耐える効率を達成していない。

51

# 階層的グループ署名




- 各層のグループ管理者は、その直下のグループ管理者あるいは、メンバーを追加することができる。
- メンバーのグループ署名から、その先祖に当たる管理者のみが、署名者が自身のどの直下の子孫の子孫であるかのみを知ることができる。
- メンバーのみが、自身に関連付けられる署名を生成できる。

52

# 前半のまとめ

- PKIの必要性と手法を概観し、その問題点を提示し、問題を解決するための技術の名前として、グループ署名等を列挙した。
- 公開鍵暗号、署名、対話証明、零知識、知識の証明、ランダムオラクルモデルとFiat-Shamir 変換等の、暗号技術の基本的な要素を説明した。
- 暗号の基本的な要素を使ったグループ署名の一般的な構成方法と効率的な具体例を示した。さらに、アキュミュレータによる効率的なメンバーの失効方法の概要を説明した。

53

# 後半

- PKIで解決しきれない問題を解決する技術として、放送型暗号、(ブラックボックス不正者追跡暗号、ブラックボックス不正者失効可能放送型暗号)、ID-based 暗号、階層型 ID-based 暗号、forward secure 暗号と署名、Key-insulated 暗号と署名、forward secure 放送型暗号、第三者検証可能ミックスネットとは何かを説明する。
- また、これらの多くの技術で使われているペアリングと呼ばれる技術を紹介する。

54

# 放送型暗号

- 大規模なメーリングリストで各メールを、その受信者全員が復号できるように普通の方式で暗号化するのではコストが大きい。
- 放送型暗号は、有料放送やDVDの暗号化にも有効である。

55

# 放送型暗号

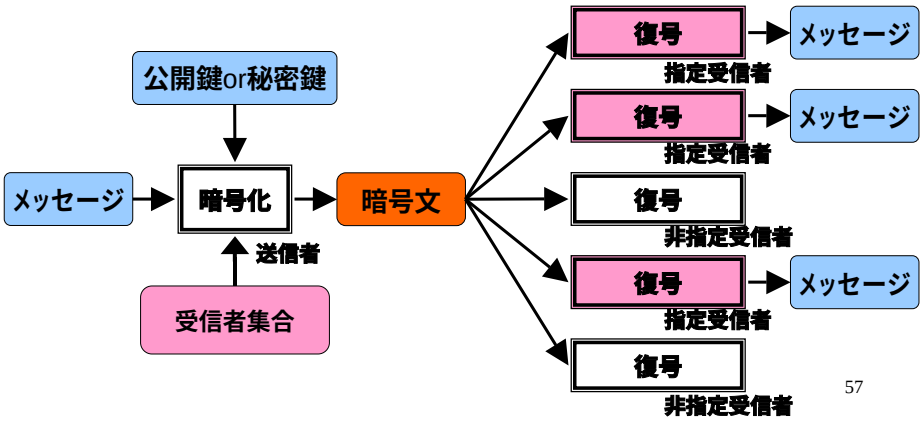
- 放送型暗号
- 放送型暗号の歴史
- 木構造を用いた放送型暗号
- インフラとしての放送型暗号
- 多人数との秘密通信を容易にするインフラ
- Boneh-Gentry-Waters 放送型暗号
- 放送型暗号をDVD等に応用した場合の問題
- ブラックボックス不正者追跡暗号
- ブラックボックス不正者追跡暗号の問題

- （ブラックボックス不正者失効可能放送型暗号に続く）

56

# 放送型暗号

- 公開鍵放送型暗号方式は、送信者が指定した受信者のみが復号できる暗号文を生成できる方式。単純には受信者の数だけ暗号文を並べれば実現可能だが、これよりも効率的に実現することが目標。




57

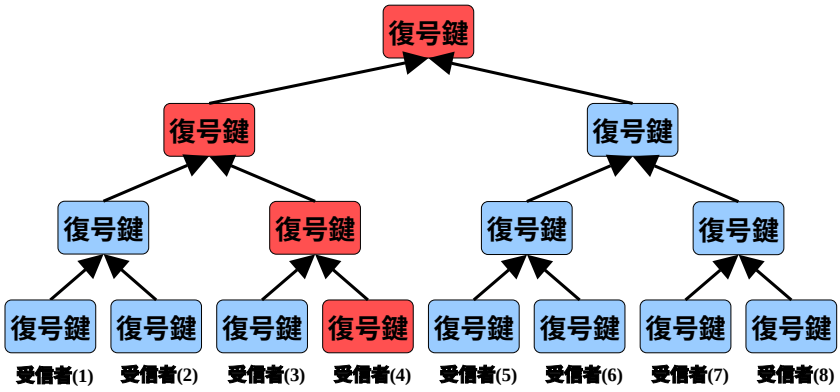
# 放送型暗号の歴史

| 提案年 | 提案者 | 暗号文長 | 方法 |
|---|---|---|---|
| 1993 | Fiat, M. Naor | | |
| 2001 | D. Naor, M. Naor, Lotspiech | 失効者数依存 | 木構造利用 (SD) |
| | ..... | | |
| 2005 | Boneh, Gentry, Waters | 定数長 (短い) | 双線形写像利用 |

58

# 木構造を用いた放送型暗号(1/2)

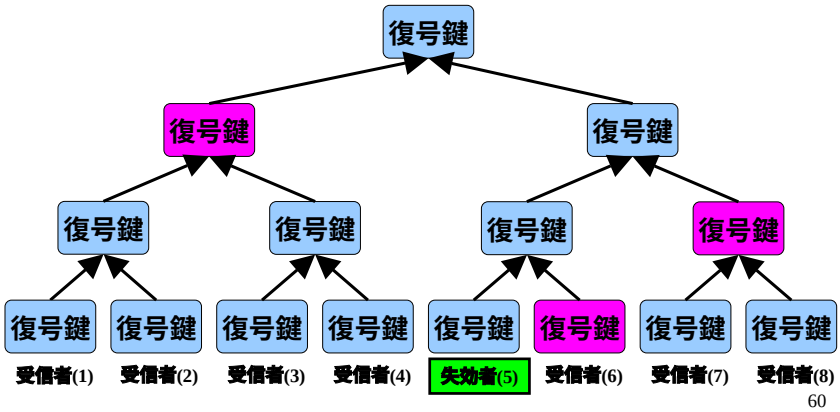
- 各受信者はルートに繋がる鍵を保持
  - (受信者(4)の所持する復号鍵の例)
- 暗号化は失効者の知らない鍵を用いて暗号化
- 暗号文は失効者の数に依存する。失効者が少ない時に有効




59

# 木構造を用いた放送型暗号(2/2)

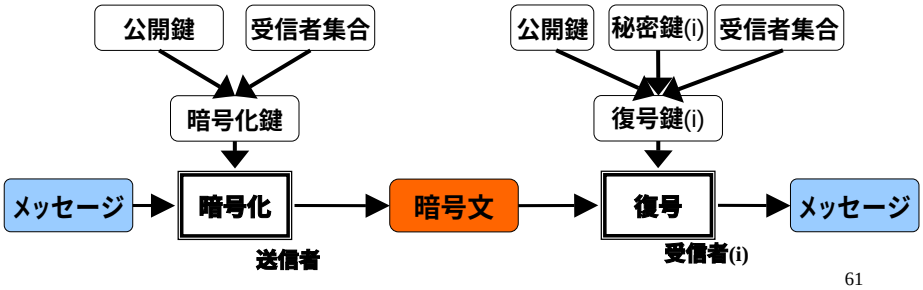
- 各受信者はルートに繋がる鍵を保持
  - (受信者(4)の所持する復号鍵の例)
- 暗号化は失効者の知らない鍵を用いて暗号化
- 暗号文は失効者の数に依存する。失効者が少ない時にのみ有効




60

# インフラとしての放送型暗号

- 放送型暗号方式をインフラとして用いるにはいくつか望まれる性質がある。
  - 暗号化は公開鍵のみで可能な（公開鍵放送型暗号）。
  - 任意の受信者集合に対しても、効率的な方式（木構造型では無理）。
  - 受信者の増減があり、受信者集合が変わった場合に必要な追加的な演算が小さい。




61

# 新しい放送型暗号方式

- 2005年 Boneh, Gentry, Waters
- 暗号文長は、システムの利用者数や受信者集合に依らず一定である
  - 例、350bit.
- 秘密鍵の長さは、システムの利用者数に依らず一定
  - 例、176bit
- 暗号化と復号に必要な計算量は、受信者集合が変化しない限り、システムの利用者数や受信者数に依らずに一定。
  - 例、ペアリング演算二回
- 受信者集合の増減があった場合に追加的に発生する計算量は、増あるいは減した受信者一人当たり楕円加算一回。
- 双線形写像（ペアリング）を利用する

62

## 多人数との秘密通信を容易にするインフラ

- PKI で証明書が付いた公開鍵と放送型暗号方式における受信者番号とを関連付ける。
  - 各公開鍵の保有者は、併せて、放送型暗号方式の受信者番号が割り当てられ、この番号に対する証明書および秘密鍵を保持する。
  - メーリングリストを構成、あるいは多人数にメッセージを送る場合は、送付先集団 (自身を含むことも可) に対応する暗号化鍵を、公開鍵より生成。各自、自身の秘密鍵と公開鍵から復号鍵を生成して保存。入会者あるいは退会者が現れれば、暗号化鍵と各自の復号鍵を更新。
  - 有料放送に加入すれば、コンテンツあるいはチャンネルに対応して、受信可能な集団に対応する暗号鍵を生成。各受信者は対応する秘密鍵を生成。
  - DVDへの応用では、復号装置に秘密鍵を格納
  - インフラに登録されている鍵を使えば受信者が受信者全員に分かってしまうのが問題。若干の工夫で回避可能。
  - DVD 等での応用では海賊版への対応が重要になる。

63

## Boneh-Gentry-Waters 放送型暗号(1/3)

- ペアリング
  - $G$, $G_T$ を素数位数$p$の加法的巡回群、$g$ を $G$ の生成元とする。$g$ の $a$ 倍を $[a]g$ と表す。
  - 双線形写像 $e$ を $e: G \times G \rightarrow G_T$で、全ての $u,v \in G, a, b \in Z$ に対して、
    - $e([a]u, [b]v) = e(u, v)^{ab}$
    - $e(g, g) \neq 1$
  - $e$ *は効率的に計算可能*
  - 超特異楕円曲線などで実現
- ハイブリッド暗号:
  - 放送型暗号を使って共通鍵暗号の共通鍵を配布し、メッセージ本体はこの共通鍵で暗号化したものを添付する。

64

## Boneh-Gentry-Waters **放送型暗号**(2/3)

**セットアップ:** ランダムに $\alpha, \gamma \in \mathbb{Z}/p\mathbb{Z}$ を選び,
$g_i = [\alpha^i]g, v = [\gamma]g, d_i = [\gamma]g_i$ を生成する.
公開鍵を $(v, g, g_1, \ldots, g_\ell,\ g_{\ell+2}, \ldots, g_{2\ell})$ とする.
$i$ 番目の受信者の秘密鍵を $d_i$ とする.

**暗号化:** ランダムに $\tau \in \mathbb{Z}/p\mathbb{Z}$ を選ぶ.
共通鍵を $c_0 = e(g_1, g_\ell)^\tau = e(g, g_{\ell+1})^\tau$ とし, その暗号文を
$C_{\mathcal{S}} = (C_1, C_2) = \left([\tau]g,\ [\tau]\left(v + \sum_{j\in\mathcal{S}} g_{\ell+1-j}\right)\right)$ とする.

65

## Boneh-Gentry-Waters **放送型暗号**(3/3)

**復号:**

$$
\begin{aligned}
c_0 &= \frac{e(g_i, C_2)}{e(C_1, d_i + \sum_{j\in\mathcal{S}, j\neq i} g_{\ell+1-j+i})} \\
&= \frac{e([\alpha^i]g, [\tau](v + \sum_{j\in\mathcal{S}} g_{\ell+1-j}))}{e([\tau]g, d_i + \sum_{j\in\mathcal{S}, j\neq i} g_{\ell+1-j+i})} \\
&= \frac{e([\alpha^i]g, [\tau]([\gamma]g + \sum_{j\in\mathcal{S}} [\alpha^{\ell+1-j}]g))}{e([\tau]g, [\alpha^i\gamma]g + \sum_{j\in\mathcal{S}, j\neq i} [\alpha^{\ell+1-j+i}]g)} \\
&= \frac{e([\tau\alpha^i]g, [\gamma]g + \sum_{j\in\mathcal{S}} [\alpha^{\ell+1-j}]g)}{e([\tau]g, [\alpha^i\gamma]g + \sum_{j\in\mathcal{S}, j\neq i} [\alpha^{\ell+1-j+i}]g)} \\
&= \frac{e([\tau]g, [\alpha^i\gamma]g + \sum_{j\in\mathcal{S}, j\neq i} [\alpha^{\ell+1-j+i}]g + [\alpha^{\ell+1}]g)}{e([\tau]g, [\alpha^i\gamma]g + \sum_{j\in\mathcal{S}, j\neq i} [\alpha^{\ell+1-j+i}]g)} \\
&= e([\tau]g, [\alpha^{\ell+1}]g) \\
&= e(g_1, [\alpha^\ell]g)^\tau
\end{aligned}
$$

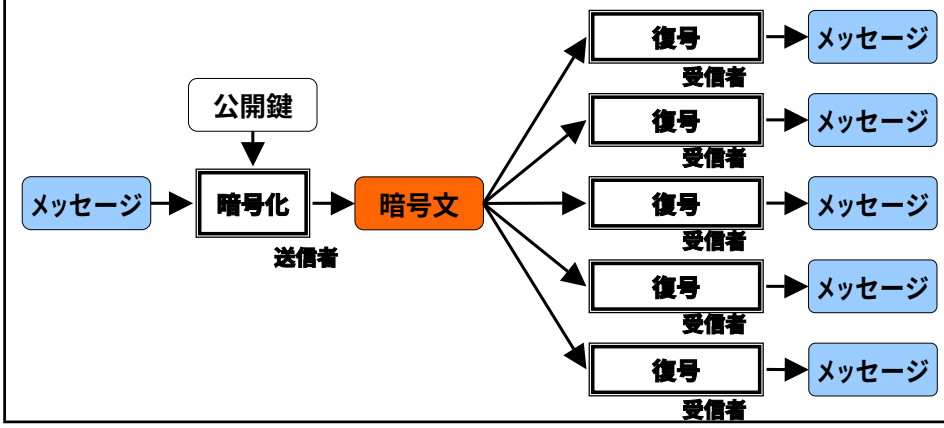
# 放送型暗号をDVD等に応用した場合の問題

- 放送型暗号方式は有料放送やDVDの暗号化にも利用価値が高い。
  - 有料放送に応用する場合は、契約者に秘密鍵を配布して使う。契約の更新時や、不払い者、受信装置の海賊版が発生すると受信者集合を変更して利用する。
  - DVDに応用する場合は、DVDの復号装置に秘密鍵を格納して利用する。復号装置の海賊版が発見されたら、以降、これに格納されている秘密鍵を受信者集合から除外する。
- 受信装置や復号装置の海賊版への対策は、これらアプリケーションでは重大な課題である。しかし、受信/復号装置の海賊版から使われている秘密鍵を特定できるかどうか分からない。なぜなら、海賊版に書かれているコードは難読化されているかもしれない。

67

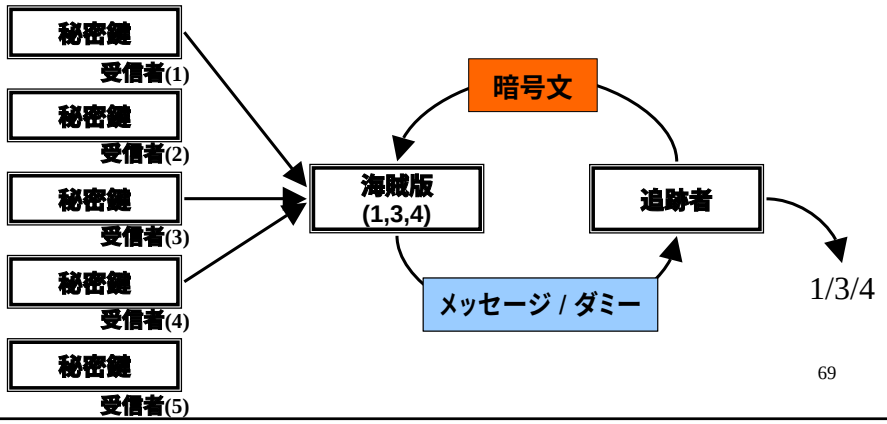
# ブラックボックス不正者追跡暗号

- ブラックボックス不正者追跡暗号方式は、受信者が複数いる暗号方式で、受信者の海賊版が存在したとき、その海賊版をブラックボックス解析することで、海賊版の生成にかかわった不正者のうち、少なくとも一人は特定できる方法。海賊版には暗号文を投げて、その結果を見るという通常の動作処理しか行わないため、海賊版のコードを解析する必要がない。



# ブラックボックス不正者追跡暗号

- ブラックボックス不正者追跡暗号方式は、受信者が複数いる暗号方式で、受信者の海賊版が存在したとき、その海賊版をブラックボックス解析することで、海賊版の生成にかかわった不正者のうち、少なくとも一人は特定できる方法。海賊版には暗号文を投げて、その結果を見るという通常の動作処理しか行わないため、海賊版のコードを解析する必要がない。




69

# ブラックボックス不正者追跡暗号の問題

- 不正者追跡暗号は、海賊版から、これの製作にかかわった不正者の少なくとも一人は特定することができる。しかし、もし海賊版がインターネットで広く配布されたような場合にはあまり有効ではない。なぜなら、不正者を特定して罰を課すことはできても、一旦配布された海賊版を、そのほかの利用者が使うことを制限できないからである。
  - 放送型暗号方式は、不正者をブラックボックス追跡できないが、もし不正者が見つかれば、これを失効できる。
  - ブラックボックス不正者追跡暗号は、不正者を知っていても失効できないが、不正者をブラックボックス追跡できる。
- 両方の特徴を併せ持つ方式が必要

70

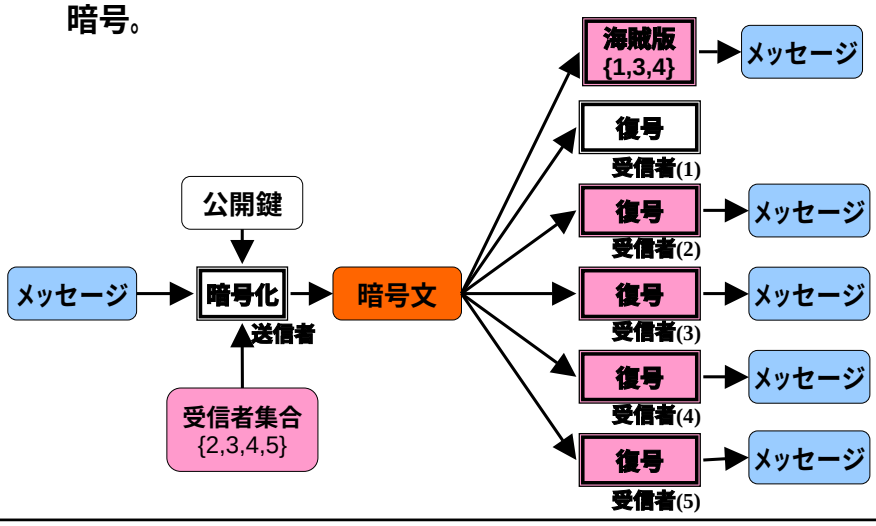
# ブラックボックス不正者失効可能放送型暗号

- ブラックボックス不正者失効可能放送型暗号
- 非自明性
- ブラックボックス不正者失効可能放送型暗号の歴史
- Boneh-Waters 方式での失効方法
- 失効方法図解
- 失効方法の性質
- ブラックボックス不正者
  失効可能暗号の構成

71

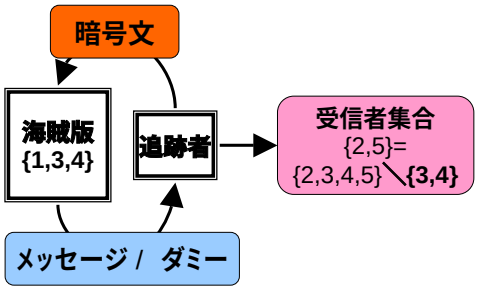
# ブラックボックス不正者失効可能放送型暗号

- 追跡者は、受信者集合に属する全ての不正者を追跡することができ、追跡された不正者を失効することができる放送型暗号。



# ブラックボックス不正者失効可能放送型暗号

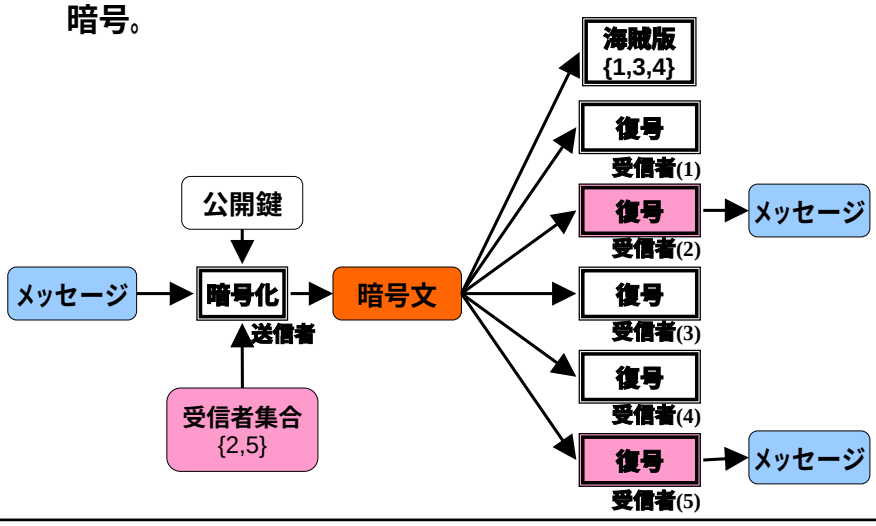
- 追跡者は、受信者集合に属する全ての不正者を追跡することができ、追跡された不正者を失効することができる放送型暗号。




73

# ブラックボックス不正者失効可能放送型暗号

- 追跡者は、受信者集合に属する全ての不正者を追跡することができ、追跡された不正者を失効することができる放送型暗号。



# 非自明性(1/2)

- 放送型暗号(BE)とブラックボックス不正者追跡暗号(TT)を重ねて用いれば、ブラックボックス不正者失効可能暗号が構成可能であるように見えるが、これは間違いである。
- 例えば、配布したい共有鍵 $Key_{TR}$ を、二つの鍵 $Key_{BE}$ と $Key_{TT}$ の和としたとする。すなわち、

$$Key_{TR}=Key_{BE}+Key_{TT}$$

$Key_{BE}$ を放送型暗号方式で暗号化して配り、$Key_{TT}$ をブラックボックス不正者追跡暗号で暗号化して配るとする。受信者は、両システムの二つの鍵を所有しており、それぞれ復号し、結果を足し合わせて $Key_{TR}$ を得るとする。

75

# 非自明性(2/2)

- 前記方法は不正者が一人ならば有効である。TTで不正者を追跡し、BEでこれを失効すればよい。
- しかし、不正者が二人いると、永遠に失効できない海賊版を作ることができる。二人の不正者をA,Bとする。$Key_{BE}$ を不正者Aの放送型暗号の秘密鍵で復号し、$Key_{TT}$ を不正者Bのブラックボックス不正者追跡暗号の秘密鍵で復号するとする。
- このばあい、追跡者は不正者Bが海賊版にかかわっていることを知ることができるので、Bを放送型暗号方式で失効する。しかし、この海賊版は放送型暗号に関してはAの鍵を持っているので、引き続き暗号文を復号することができる。

$$Key_{BE}=\text{復号}(A\text{の}BE\text{鍵},ciph_{BE})$$
$$Key_{TT}=\text{復号}(B\text{の}TT\text{鍵},ciph_{TT})$$

76

## ブラックボックス不正者失効可能放送型暗号の歴史

- 提案年　提案者　　許容不正者数
- 2001 D. Naor, M. Naor, Lotspiech 不正者が多いと失効は困難

  .....
- 2006 Boneh, Waters 任意数の不正者を失効可能
- 2006 Furukawa, Attrapadung 任意数の不正者を失効可能

- 2006 年Boneh, Water が初めて、任意の数の不正者により作成された海賊版の復号装置を、ブラックボックス失効できる放送型暗号方式を提案した。

77

## Boneh-Waters 方式での失効方法(1/2)

- 海賊版が与えられており、この海賊版は受信者集合 $S$ に対する暗号文を復号できるものとする。システムの利用者数を $n$ とする。
- 追跡用の暗号文の集合 $\{E(S,i)\}_{i=1,\ldots,n}$ を次の様なものとする。
  - $j \in \{i,\ldots,n\} \cap S$ なる受信者$(j)$は、正常の復号処理により $E(S,j)$ の復号に成功する。
  - 複数の秘密鍵を保持している攻撃者が、二つの暗号文 $E(S,i)$ と $E(S,i+1)$ を識別できるのは以下の場合に限られる。
    - $i \in S$ かつ受信者$(i)$の秘密鍵を保持している。
  - $E(S,n)$ は、メッセージの情報を完全になくした暗号文である。

78

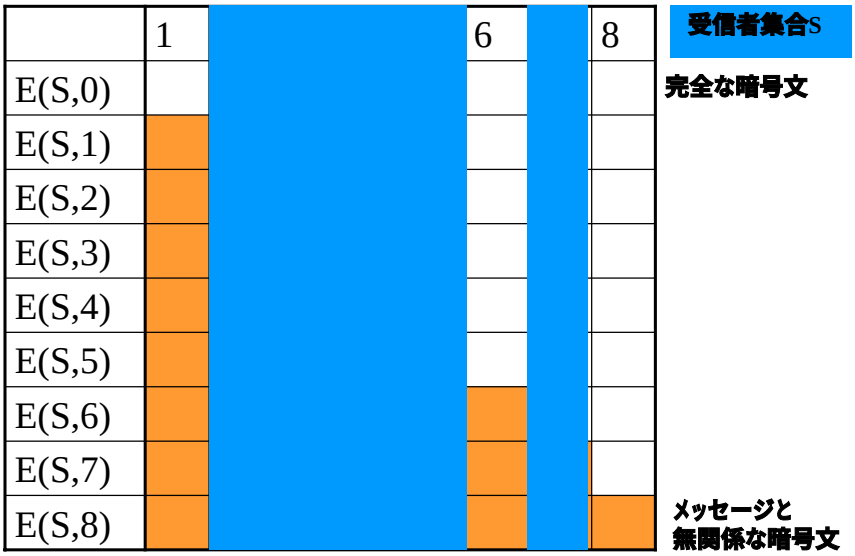
# 失効方法(2/2)

- 失効者は $i=1,\ldots,n$ に対して次の操作を行う
  - $E(S,i), E(S,i+1)$ の二種類暗号文のうち、無作為に片方を選んで海賊版に与え正しく復号するかを調べる作業を十分な回数行う。
  - 両者に差があった場合、少なくとも受信者$(i)$が海賊版の作成にかかわっているとし、これを $S$ から取り除く

    $S \rightarrow S \setminus \{i\}$
- 最大 $n$ 回繰り返して最終的に得られた $S$ に対する暗号文を、海賊版は復号できない。

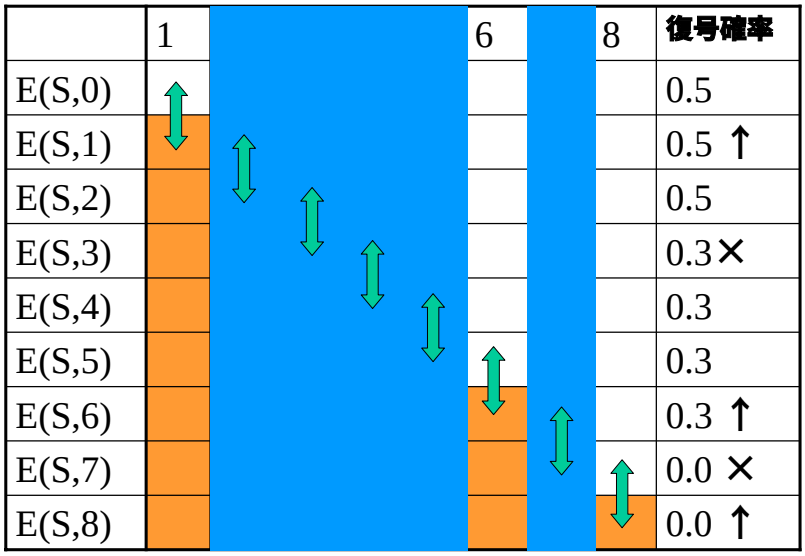
79

# 失効方法図解(1/4)




80

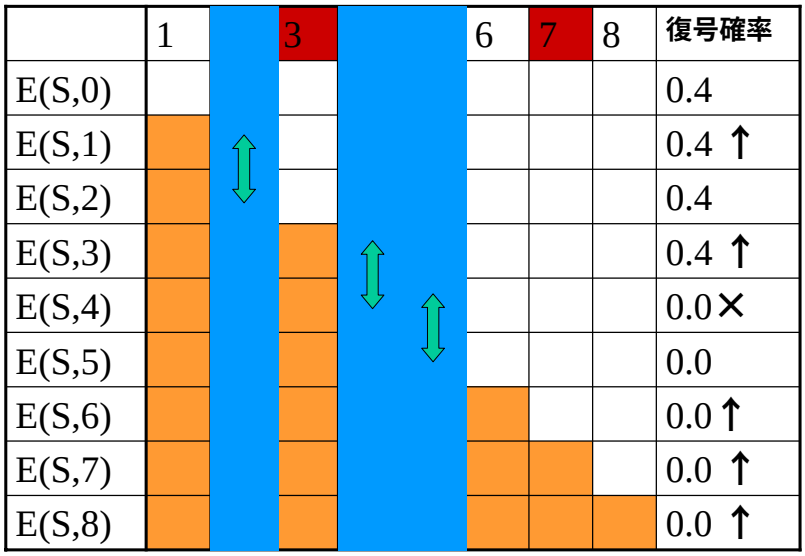
# 失効方法図解(2/4)



| | 1 | 6 | 8 | 復号確率 |
|---|---|---|---|---|
| E(S,0) | | | | 0.5 |
| E(S,1) | | | | 0.5 ↑ |
| E(S,2) | | | | 0.5 |
| E(S,3) | | | | 0.3 × |
| E(S,4) | | | | 0.3 |
| E(S,5) | | | | 0.3 |
| E(S,6) | | | | 0.3 ↑ |
| E(S,7) | | | | 0.0 × |
| E(S,8) | | | | 0.0 ↑ |

81

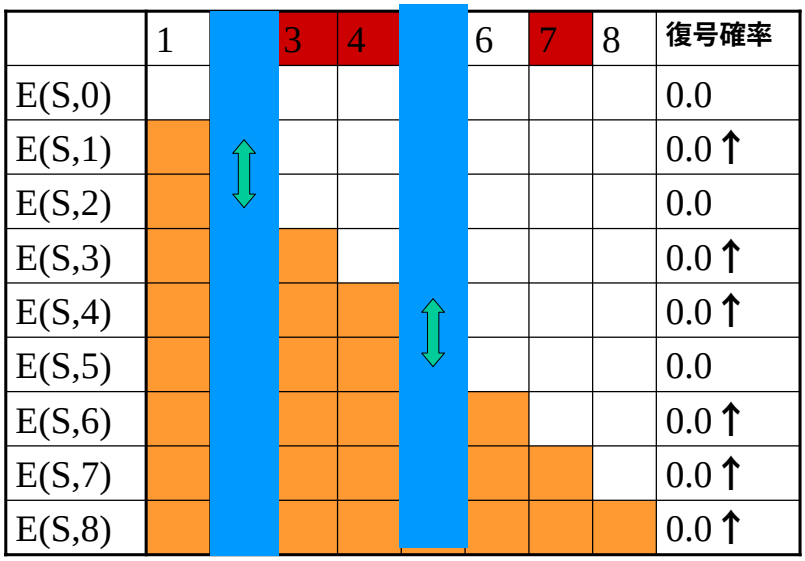
# 失効方法図解(3/4)



| | 1 | 3 | 6 | 7 | 8 | 復号確率 |
|---|---|---|---|---|---|---|
| E(S,0) | | | | | | 0.4 |
| E(S,1) | | | | | | 0.4 ↑ |
| E(S,2) | | | | | | 0.4 |
| E(S,3) | | | | | | 0.4 ↑ |
| E(S,4) | | | | | | 0.0 × |
| E(S,5) | | | | | | 0.0 |
| E(S,6) | | | | | | 0.0 ↑ |
| E(S,7) | | | | | | 0.0 ↑ |
| E(S,8) | | | | | | 0.0 ↑ |

82

# 失効方法図解(4/4)



| | 1 | 3 | 4 | 6 | 7 | 8 | 復号確率 |
|---|---|---|---|---|---|---|---|
| E(S,0) | | | | | | | 0.0 |
| E(S,1) | ■ | | | | | | 0.0 ↑ |
| E(S,2) | ■ | | | | | | 0.0 |
| E(S,3) | ■ | ■ | | | | | 0.0 ↑ |
| E(S,4) | ■ | ■ | ■ | | | | 0.0 ↑ |
| E(S,5) | ■ | ■ | ■ | | | | 0.0 |
| E(S,6) | ■ | ■ | ■ | ■ | | | 0.0 ↑ |
| E(S,7) | ■ | ■ | ■ | ■ | ■ | | 0.0 ↑ |
| E(S,8) | ■ | ■ | ■ | ■ | ■ | ■ | 0.0 ↑ |

83

# 失効方法の性質

- 失効は、決して正直な受信者を失効してはならない。前述の失効方法では、正直な受信者*(i)*の鍵がなければ、が *E(S,i)* と *E(i+1)* を識別し得ないので、**正直な受信者は失効されない**。
- 不正者の数に依らず、海賊版が無効となる受信者集合を求めることができる。
- 海賊版は、状況によって用いる秘密鍵を使い分ける可能性がある。よって、あるある S に対して復号できなくなっても、$S' \subset S$ なる $S'$ に対して復号が可能である場合もある。このため、新たな受信者集合 S を用いるときは、全ての既存の海賊版が復号できるか改めて確かめる必要がある。

84

# ブラックボックス不正者<br>失効可能暗号の構成

- 位数が合成数 $n=pq$ のペアリングを持つ群を利用。ここで $p, q$ は素数。$G, G_T$ を素数位数 $n=pq$ の加法的巡回群とする。
  - $G, G_T$ の元で位数が $p$ のもののなす部分群をそれぞれ $G', G'_T$ とする。
  - $G, G_T$ の元で位数が $q$ のもののなす部分群をそれぞれ $G'', G''_T$ とする。
  - $G, G', G''$ の生成元をそれぞれ $g, g', g''$ とする。
- この時、$e(g',g'') = e(g,g)^{0=1}$ が成り立つ。このことを利用すれば、$(G, G_T)$ の世界に、$(G', G'_T)$ の世界の暗号と、$(G'', G''_T)$ の世界の暗号を二重に作りんだむことができる。これを利用して、$E(S,i)$ の様な形で部分的に破壊された暗号文を $\sqrt{n}$ の大きさで生成できる。
- 構成の詳細は複雑なため省略。その効率は、暗号文の長さが $\sqrt{n}$ であるなど、放送型暗号に比べると見劣りする。

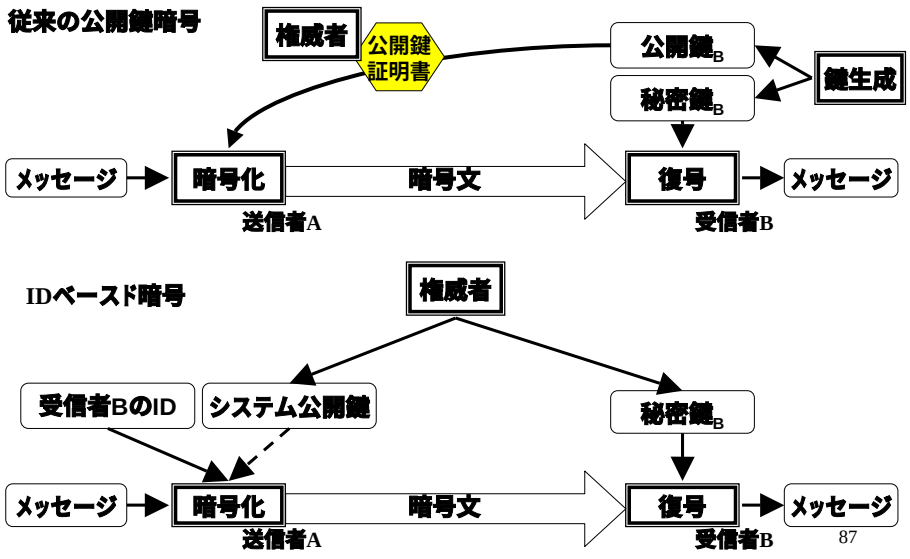
85

# ID-Based 暗号

- 名刺などの印刷物に書かれたemail アドレスにメールを出したい。従来の公開鍵暗号でこれを行う為には公開鍵を別途入手する必要があり、不便。
  - 双方向通信を行う必要がある。
- ID-based 暗号は、任意の文字列を公開鍵として暗号通信が行える公開鍵暗号方式。

86

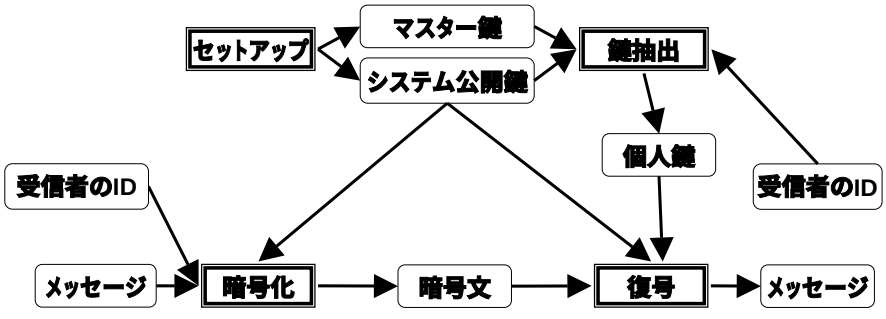
# 従来の公開鍵暗号と ID-based 暗号






# ID-based 暗号のモデル

- **セットアップ**：システム公開鍵とマスター鍵を生成する
- **鍵抽出**：ID とマスター鍵が与えられたら、対応する個人鍵を生成する
- **暗号化**：暗号文、ID、システム公開鍵が与えられたら暗号文を生成する。
- **復号**：暗号文、個人鍵、システム公開鍵が与えられたらメッセージを出力する。



# ID-based 暗号の歴史

| 提案年 | 提案者 | |
|---|---|---|
| • 1984 | Shamir | 概念を提案 |
| • 2001 | Boneh-Franklin | 初めての実用的な方法<br>（ペアリングを利用） |
| • 2004 | Boneh-Boyen | ランダムオラクルによらない方法 |
| • 2005 | Waters | 同上。効率的になった |
| • 2006 | Gentry | さらに効率的な方法 |

89

# Boneh-Franklin ID-based 暗号

$G, G_T$を位数 $p$ の群で、双線形写像 $e: G \times G \rightarrow G_T$ を持つとする。$H$ を値域が $G$ のハッシュ関数とする

- **セットアップ**：ランダムに $g \in G, s \in Z/pZ$ を選ぶ。$y=[s]g$ とする。システム公開鍵を $(g, y)$ 、マスター鍵を $s$ とする。
- **鍵抽出**：ある ID が与えられたら、個人鍵を $d_{ID}=[s]H(\mathrm{ID})$ とする。
- **暗号化**：暗号文 $M \in G_T$ の暗号文は次の通り。ランダムに $r \in Z/pZ$ を選び、

  $(c_1,C_2)=([r]g, M \cdot e(H(\mathrm{ID}), y)^r)$
- **復号**：$M=C_2 / e(d_{ID}, c_1)$

  $=M\, e(H(\mathrm{ID}), y)^r / e([s]H(\mathrm{ID}),[r]g)$

  $= M\, e(H(\mathrm{ID}), [s]g)^r / e([s]H(\mathrm{ID}),[r]g)$

  $= M\, e(H(\mathrm{ID}), g)^{rs} / e(H(\mathrm{ID}),g)^{rs}$

  $=M$

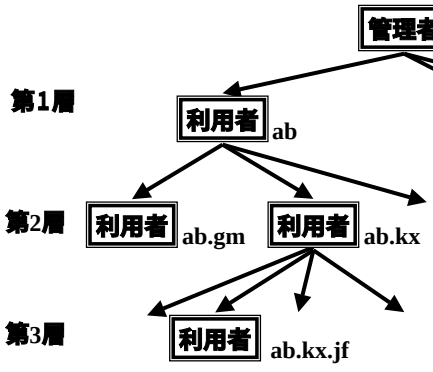
90

# ID-based 暗号のインフラ

- ID-based 暗号のインフラを構築した場合、唯一つの権威者が鍵を発行することになる。このような権威者を運営することが現実的であるか疑問である。
- PKIと同様に階層的な鍵発行を行う方式として、階層的ID-based 暗号方式がある。

91

# 階層的ID-based 暗号




- 各利用者のIDは、その親のIDに新たな番号を連結したものである。
- 最上位にいる管理者は第1層の利用者に対して個人を発行できる。
- 各利用者は、自身の子となる下位の利用者に個人鍵を発行できる。

- 非常に効率的な階層的ID-based暗号が存在する(Boneh-Boyen-Goh 2005)

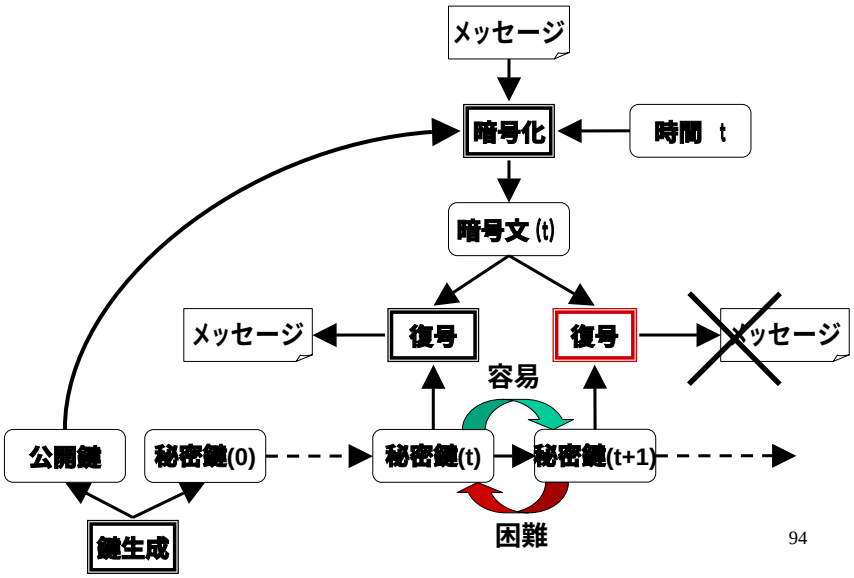
92

# 未来安全(Forward Secure)暗号/署名

- 秘密鍵は単なるデータであるため、漏洩すると回収できない。
- 秘密鍵を携帯電話の様な機器に格納して使っていて、携帯電話を紛失すると、他人に秘密鍵を使われる可能性がある。
- 秘密鍵はネットワーク上では自身の認識手段であるため、人格の紛失とも言える。
- Forward secure 暗号は、秘密鍵を更新し続けるが、対応する公開鍵は同一であり続ける方式で、現在の秘密鍵では過去の暗号文を復号することができない方式。
- Forward secure 署名は、秘密鍵を更新し続けるが、対応する公開鍵は同一であり続ける方式で、現在の秘密鍵では過去の日付の署名を生成することができない方式。
- 両方式とも、過去の鍵は消去して利用する。また、現在の鍵から未来の鍵は生成できるが、過去の鍵は生成できない。よって、現在の鍵を紛失しても、過去の暗号文を読まれたり、過去の日付の署名を生成されることはない。未来に関しては不安が残るが、直ちに公開鍵の失効手続きをとれば、被害は軽減される。

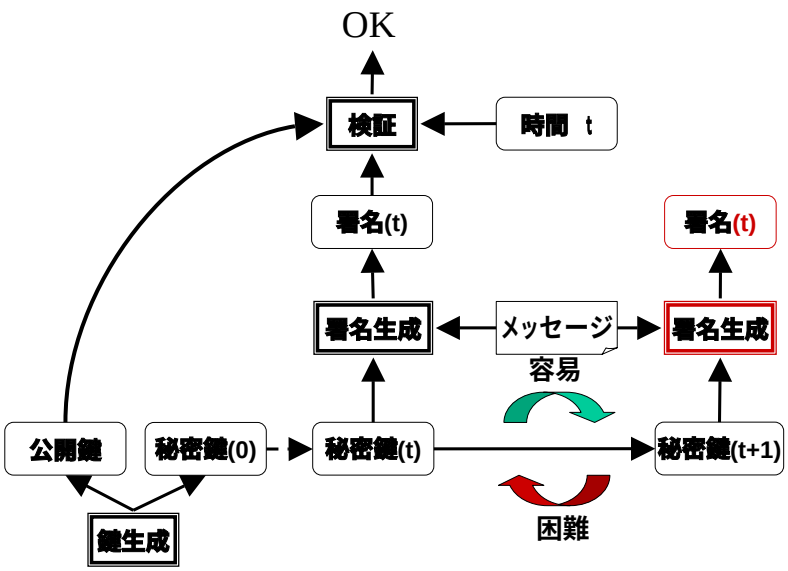
93

# Forward Secure 暗号



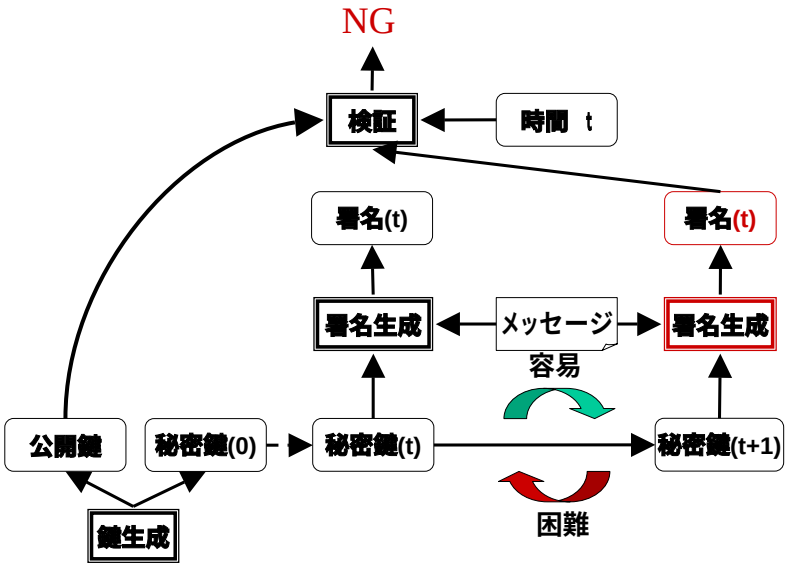

94

# Forward Secure 署名




95

# Forward Secure 署名




96

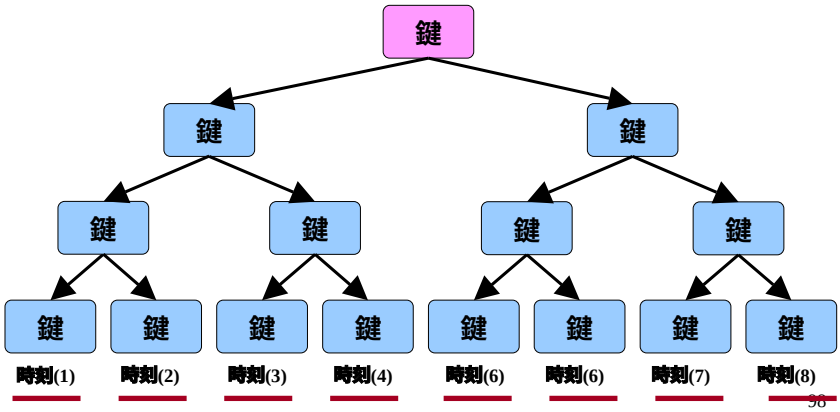
# Forward Secure 暗号/署名の構成

- Forward Secure 暗号は階層的 ID-based 暗号から構成可能である。そして、効率的な階層的 ID-based 暗号が存在するので、効率的な構成可能である。しかし、秘密鍵の長さは、時間間隔の数の対数に比例する。
- ID-based 暗号から署名を構成する方法が知られている。階層型ID-based 暗号も ID-based 暗号を含んでいるため、同様のことができる。これにより、Forward Secure 署名も同様に構成可能。

97

# 階層型ID-based 暗号を用いた Forward Secure 暗号の構成

- 各時刻に対応する暗号文は、階層型暗号の各最下層のノードに対する暗号文が対応する。
- 時刻 $t$ においては、時刻$t$以前の鍵は抽出できないけれど、時刻 $t$ 以降の鍵を生成できるノードの鍵を効率的に保持する。



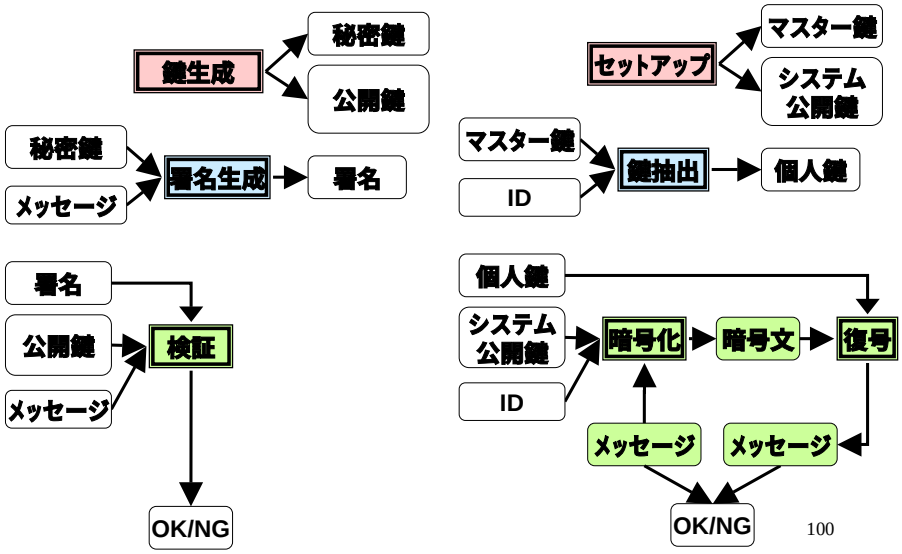

98

# Forward Secure 署名の構成

- ID-based 暗号から署名が構成できる
  - 鍵生成：ID-based 暗号のセットアップを利用。署名の公開鍵はセットアップの出力するシステム公開鍵。署名の秘密鍵はマスター鍵。
  - 署名：ID-based 暗号の鍵抽出を利用。署名するメッセージは鍵抽出する対象のID。署名は、個人鍵。
  - 検証： ID-based 暗号の暗号化と復号を利用。ランダムに選んだ文字列を署名対象のメッセージであるIDで暗号化し、これを署名である個人鍵で復号して同じ文字列に戻るかを検査する。

99

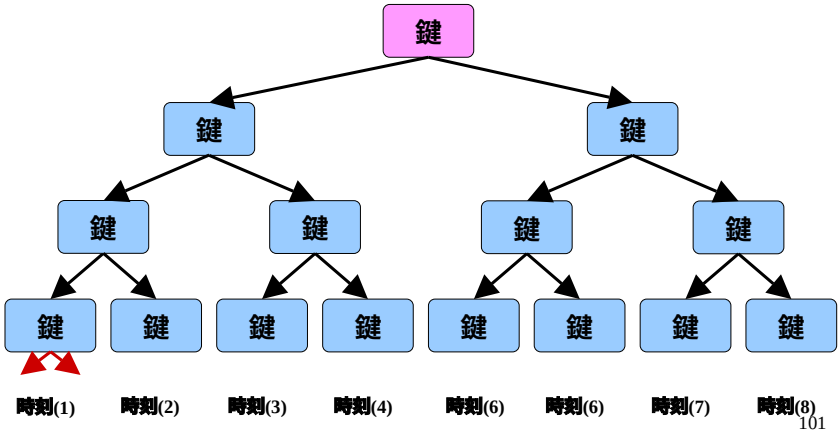
# ID-Based 暗号から署名の構成




100

# 階層型ID-based 暗号を用いた Forward Secure 署名の構成

- 階層型 ID-based 暗号の最下層はID-based暗号になっている
- 最下層にあるID-based 暗号を署名に変換すればよい。



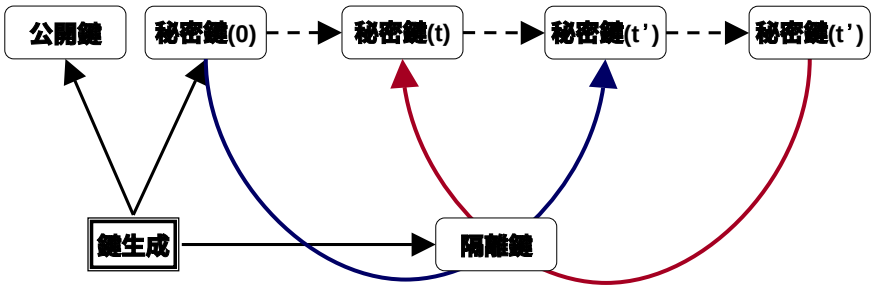

101

# 隔離鍵(key-insulated)暗号／署名

- Forward security の問題
  - forward secure 暗号 では過去の暗号文を自分も復号できない。
  - forward secure 暗号や署名では、秘密鍵を漏洩したことに気がつかなかったら、未来の暗号文を復号されたり、メッセージに対して未来に署名を生成される。
- Key-insulated 暗号と署名
  - 秘密鍵の他に、隔離されたより安全なデバイス上に隔離鍵を持つ。隔離鍵を用いると任意の時刻の秘密鍵が生成できる。
  - 鍵の更新が一日一回である場合に、秘密鍵を携帯電話に格納して使う場合等には、自宅の充電器に接続すると鍵が更新されるといった形で使う。
  - 鍵の更新パターンは様々なものが考えられている。例えば、職場と自宅、二つの充電器に交互に接続しないと鍵を更新できないようにして、充電器ごとの盗難に備える等。

102

# Key-Insulated 暗号




103

# Key-Insulated 暗号/署名の構成

- 単純な型の鍵管理を行う場合は、Key-Insulated 暗号は ID-based 暗号から直ぐに構成可能で、Key-Insulated 署名は単なる署名から構成可能ある。その他の亜種の多くも、階層的 ID-based 暗号から構成可能である。

104

## Forward Secure, Key-Insulated 暗号、署名方式の特徴とPKI

- Forward secure 暗号、署名は秘密鍵が長くなる。
  - forward secure 暗号 では過去の暗号文を自分も復号できない。
  - forward secure 暗号や署名では、秘密鍵を漏洩したことに気がつかなかったら、未来の暗号文を復号されたり、メッセージに対して未来に署名を生成される。
- Key-insulated 暗号と署名
  - 秘密鍵の他に、隔離されたより安全なデバイス上に隔離鍵を持つ。隔離鍵を用いると任意の時刻の秘密鍵が生成できる。
  - 鍵の更新が一日一回である場合に、秘密鍵を携帯電話に格納して使う場合等には、自宅の充電器に接続すると鍵が更新されるといった形で使う。
  - 鍵の更新パターンは様々なものが考えられている。例えば、職場と自宅、二つの充電器に交互に接続しないと鍵を更新できないようにして、充電器ごとの盗難に備える[Hanaoka05]等。
- PKIで保証される鍵がforward secure である、あるいはkey-insulated となっていると安全性が高まる。

105

## 鍵交換

- 暗号は送信者から受信者へ一方向の通信である。これは、相手がオンラインであるとは限らない場合に有効な通信方法である。
- 相手が常にオンラインである場合は、鍵交換を用いた対話的な暗号通信が有効である。鍵交換はforward security を簡単に達成することができる。しかし、key-insulated の様なことはできない。

106

# Forward Secure 放送型暗号

- 放送型暗号においても、通常の公開鍵暗号と同様に鍵の紛失に備える必要がある。
- 階層型ID-based 暗号を用いた forward secure 暗号の方式と、Boneh-Gentry-Waters の放送型暗号をうまく組み合わせた、効率的な forward secure な放送型暗号が提案されている。
  - 2006年 Attrapadung, Furukawa, Imai

107

# ミックスネット

- PKIが個人に普及したときには、様々な社会的サービスをネットワーク上で受けることができるようになることが期待される。しかし、信頼できる公開鍵に基づく暗号と署名が使えても、実現が難しいサービスが多く、サービスに特化した暗号プロトコルを構成する必要がある。
- 集計の正当性と投票の匿名性という、一見矛盾する要件を満たさなければならない電子投票が、ミックスネットで実現できることを説明する。

108

# ミックスネット

- 電子投票の要件
- 電子投票の問題
- ミックスネットによる電子投票システムの構造
- 第三者検証可能ミックスネット
- ElGamal シャッフル
- ElGamal ミックスネット
- 第三者検証可能ミックスネットの効率
- シャッフルの零知識証明方法の歴史
- シャッフルの零知識証明
- 電子投票のその他の問題

109

# 電子投票の要件

- 電子投票では、次の要件が守られなければならない
  - 集計の正当性の保証
    - 有権者による投票であり、二重投票がない。
    - 表の水増し、改竄がない。
  - 投票の匿名性の保証
    - 投票内容の秘密が守られる。

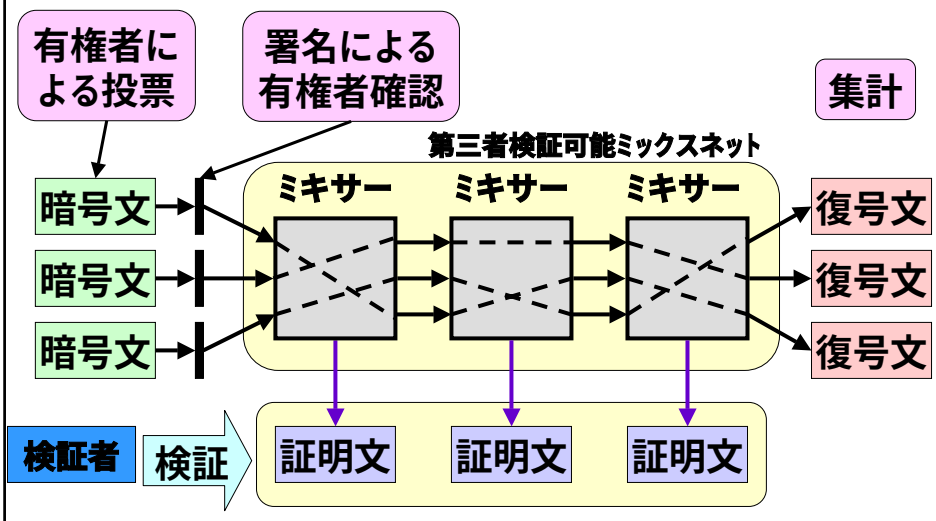
110

# 電子投票の問題

- 単純な方式では集計の正当性と投票の匿名性の両方を同時に保証する事はできない。
  - 有権者の投票であることを保証するためには、各票には署名をつける必要がある。この投票内容は暗号によって隠されている必要がある。
  - 暗号化された投票内容を復号したものを集計した結果が、投票の結果である。
    - この復号や集計の処理を、復号の秘密鍵を示して行うと、匿名性が守られない。
    - 結果だけを公表すると、偽りの集計結果があり、集計の正当性が保証されない。
- 第三者検証可能ミックスネットを用いて解決する。

111

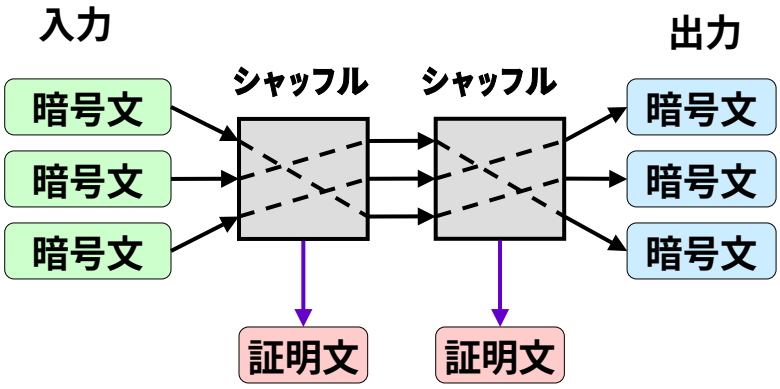
# ミックスネットによる電子投票システムの構造




Mix-netと匿名性

票を攪拌しつつ復号する．
誰にも全体の攪拌は分からない

# 第三者検証可能ミックスネット




113

# ElGamal シャッフル

- ElGamal 暗号文の再暗号化：
  - 公開鍵 $(g, y)$ と暗号文 $(g_i, h_i)=(g^r, m_i y^r)$ が与えられたとする。
  - 乱数 $s \in Z/qZ$ を選び、$(g_i, h_i)$ の再暗号化
    $(g'_i, h'_i)= (g_i\ g^s, m_i\ y^s)= (g^{r+s}, m_i\ y^{r+s})$ を計算する。
- ElGamal 暗号文のシャッフル：
  - $n$ 個のElGamal 暗号文の列 $((g_1, h_1),\ldots, (g_n, h_n))$ が与えられたとする。
  - 列の暗号文の順番をランダムに入れ替え(*置換* $\pi$)た後、各ElGamal暗号文をそれぞれ独立に選んだ乱数 $s(i)\in Z/qZ$ で再暗号化した結果
    $((g'_1, h'_1),\ldots, (g'_n, h'_n))$ を出力する。ただし、
    $(g'_i, h'_i)= (g_{\pi(i)}\ g^{s(i)}, h_{\pi(i)}\ y^{s(i)})$
    $= (g^{r(\pi(i))+s(i)}, m_{\pi(i)} y^{r(\pi(i))+s(i)})$

114

# ElGamal シャッフルの安全性

- ElGamal 暗号文のシャッフルの入力と出力が与えられても、対応関係は分からない。
  - 入力された列のi 番目の要素 $(g'_i, h'_i)$ が $(g'_j, h'_j)$ に対応しているかを判別することは、
    $(g'_i/g_j, h'_i/h_j)=(g^{s(i)}, y^{s(i)})$
    なる $s(i)$ が存在するかの判定である。
  - Diffie-Hellman 判別問題が難しい限り、ElGamal 暗号文のシャッフルの入力と出力の対応関係は分からない。

115

# ElGamal ミックスネット

- ElGamal シャッフルを連続して行うミックスネットを ElGamal ミックスネットと呼ぶ。
- 各シャッフルにおいて、入力と出力との対応関係は、置換πの生成者しか分からないので、ミックスネット全体での置換を知るには、各シャッフルを行った者全てが共謀しない限りわからない。
- 各シャッフルが、暗号文の入れ替えや改竄などを行わずに、正しく行われた事を零知識証明すれば、ミックスネット全体として暗号文の入れ替え等が行われなかったことが証明できる。
- 電子投票に応用した場合、
  - 全てのシャッフル者が共謀しない限り投票の匿名性が保証される。
  - たとえ全てのシャッフル者が共謀しても、集計の正当性は保証される。
- (復号の問題は省略して考えている)

116

# 第三者検証可能ミックスネットの効率

- 第三者検証可能ミックスネットを用いて電子投票を現実的な時間で実効できるかは、シャッフルの正当性の零知識証明が高速にできるかによる。

117

# シャッフルの零知識証明方法の歴史

| 提案年 | 提案者 | 備考 |
|---|---|---|
| 1981 | Chaum | ミックスネットを提案 |
| 1995 | Sako-Kilian | シャッフルの零知識証明を提案 |
| 2001 | Furukawa-Sako | 同上 (初めての実用的な方法) |
| 2001 | Neff | 同上 |
| 2003 | Groth | 同上 |
| 2004 | Furukawa | 同上 |
| 2005 | Peng | 同上 |

118

# シャッフルの零知識証明 (1/4)
(Furukwa 04)

- 置換行列:
  各行各列に唯一つ 1 が存在し、そのほかの要素は 0 の行列

$$\begin{pmatrix} b \\ a \\ c \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

119

# シャッフルの零知識証明 (2/4)

- シャッフルの行列表現:置換行列 $(\pi_{ji})$ を用いると、シャッフルは次のように表現できる

$$(g_i', h_i') = (g^{r_i} \prod_{j=1}^{n} g_j^{\pi_{ji}}, y^{r_i} \prod_{j=1}^{n} h_j^{\pi_{ji}})$$

- シャッフルであることは、 $g, y, ((g_i, h_i))_{i=1,\ldots,n}, ((g_i', h_i'))_{i=1,\ldots,n}$ に対して、次の二つのが成り立つことと同値
  - 上の式を満たす $r_i, (\pi_{ji})$ が存在する。
  - $(\pi_{ji})$ は、置換行列である。

120

# シャッフルの零知識証明 (3/4)

- **置換行列の性質：**
  $(\pi_{ji})$ **が置換行列である**

$$\Leftrightarrow \quad \sum_{h=1}^{n} \pi_{hi}\pi_{hj} = \delta_{ij}, \quad \sum_{h=1}^{n} \pi_{hi}\pi_{hj}\pi_{hl} = \delta_{ijl}$$

$$\left[ \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{その他} \end{cases}, \delta_{ijl} = \delta_{il}\delta_{jl} \right]$$

121

- **ランダムに選ばれた** $c_j$ **に対して次の式が成り立つとする**

$$c_i^* = \sum_{j=1}^{k} c_j \pi_{ji}$$

**この時、次の式が無視できない確率で成り立つためには、**

$$\sum_{i=1}^{k} (c_i)^3 - \sum_{i=1}^{k} (c_j^*)^3 = 0$$

,

$$\Leftrightarrow \sum_{i,j,l}^{k} (\sum_{h=1}^{k} \pi_{hi}\pi_{hj}\pi_{hl} - \delta_{ijl}) c_i c_j c_l = 0$$

**次の式が成り立っていなければならない。**

$$\sum_{h=1}^{k} \pi_{hi}\pi_{hj}\pi_{hl} = \delta_{ijl}$$

**これは、** $(\pi_{ij})$ **が置換行列の時である。**

122

# 電子投票のその他の問題

- ミックスネットを利用した電子投票は、在宅投票も可能にする。しかし、予め指定された暗号文を投票することで、票の売買が容易になる。
- 投票する暗号文は確率暗号であるが、投票したいメッセージに対する多数の暗号文の中から自由に選べない様にし、さらに、選ばれた暗号文がどのメッセージに対応するか投票者は確認できるが、これを誰にも証明できない様にすることで、票の売買を困難にするレシートフリーと呼ばれる方法がいくつか提案されている。

123

# ペアリング

- ここまでに紹介された様々な暗号技術では、ペアリングと呼ばれる技術が利用されているものが多い。
- 従来は、概念は存在しても実用に耐える具体例がなかった方式が、ペアリングの利用により実用的な方式が具体的に構成されるようになった。
- ペアリングは2001年のID-based暗号から積極的に使われるようになった、その利用が比較的新しい技術である。
- ペアリングは特殊な楕円曲線上で実現されるが、このような曲線として様々なもの提案されている。これらは、主にペアリングの計算速度を上げることが目的である。
- 当初、ペアリングの計算は非常に時間がかかるものであったが、最近はPCで数ミリ秒で可能となっており、暗号では益々有効かつ重要な技術となってきている。
- ペアリングのさらなる高速な実装を行うことは現在重要な課題である。

124

# まとめ

- 脅威が主導してPKIの常時的使用が必須となる場合を考え、将来有効となる暗号プロトコルを紹介した。
- 署名や暗号と言った単純な方式ではないこれらの暗号プロトコルは、従来実用に耐える効率を実現していなかったが、いずれも21世紀に入ってから効率的なものが次々と提案されたプロトコルである。これはペアリングの利用によるところが大きい。

125

# 質問

126