1. 21世紀初頭の暗号技術

5. 鍵生成と鍵管理

岡本 栄司
筑波大学 システム情報工学研究科
okamoto@risk.tsukuba.ac.jp

鍵生成と鍵管理の役割

暗号鍵管理は，暗号システムにおいて暗号化鍵／復号鍵(単に鍵ともいう)を安全に運用する手段である．通常，鍵管理は鍵の生成，配送，保管および廃棄（無効化）からなる．このうち，生成と配送はまとめて行われることも多く，鍵更新あるいは鍵共有と呼ばれる一種のマルチパーティプロトコルである．暗号システムは，伝えたいメッセージ（データ）そのものの暗号化（以下，データ暗号化と呼ぶ）とこの暗号鍵管理からなり，全体をハイブリッド暗号と呼ぶこともある．これに対する攻撃と安全性を厳密に定義した上で，安全な暗号システムの議論がなされている．KEM（Key Encapsulation Mechanism）はその代表的な定式化で，鍵共有の安全性をデータ暗号のための共通鍵暗号を組み合わせて証明する枠組みである．

鍵生成

鍵は各暗号方式に依存した構造をしている．たとえば，共通鍵暗号ではある決まった長さの乱数を鍵として用いる．公開鍵暗号系では，通常，鍵にはいくつかの制約条件がある．たとえばRSA暗号系では法nは大きな素数の積であり，それらの素数にも条件がある．また，公開鍵eや秘密鍵dもnとある種の関係を満たしていなければならない．したがってこれらの条件を満たす範囲でできるだけランダムに鍵を生成することになる．このため，鍵生成においては必ず乱数が必要となり，しかも，エントロピーの観点からそれは一様乱数が望ましい．

乱数には物理現象などを用いた真性乱数とソフトウェア的に生成する疑似乱数があるが，現実には使いやすさから疑似乱数が用いられることが多い．

乱数生成そのものはここでは述べないが，本文では，乱数は生成器により理想的な乱数が生成されるものと仮定する．

鍵共有方式

具体的な鍵共有方式にはさまざまな方式がある．最近は公開鍵暗号系が主流であるためここでも公開鍵暗号系を用いた方式を取り上げるが，共通鍵暗号のみを用いた鍵共有方式も可能であり，実際以前にはよく用いられていた．

●公開鍵サーバ

公開鍵をサーバにおき必要に応じて各エンティティがアクセスする方式であり，公開鍵暗号系の原点となっている方式といえる．この公開鍵サーバは改ざんされないように安全に管理しなければならない．もし，攻撃者が自分の秘密鍵SKと公開鍵PKを生成し，そのPKをサーバ上の正当な公開鍵に置き換えると，PKで暗号化されたメッセージは攻撃者がSKで復号できてしまう．

●公開鍵証明書

Kohnfelderによって1978年に提案された方式であり，X.509で標準化され実用化されている．

エンティティの公開鍵に，CA（Certificate Authority）と呼ばれる鍵登録機関の署名鍵でディジタル署名を施したものを公開鍵証明書という．受信者はこの公開鍵証明書を相手に送る．相手はCAの検証公開鍵で正当性を検証し，真ならば正しい公開鍵とみなす．このとき，このCAの公開鍵の正しさを保証するためさらに上位のCA'がCAの公開鍵に署名を施すことがある．このようにしてCAの階層構造ができるが，これをPKI（Public

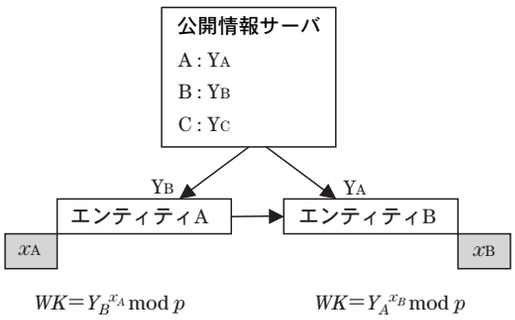

図 -1　DH 公開鍵配送方式

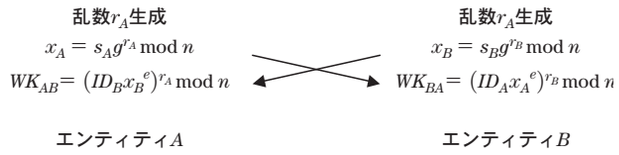

図 -2　ID-KDS

Key Infrastructure）という．送信側は，受信側の公開鍵 *p-Entity* の正当性が検証できたなら，ランダムに生成したデータ暗号化鍵を *p-Entity* で暗号化して送る．受信側は自分の秘密復号鍵 *s-Entity* で復元すれば鍵共有が可能となる．

SSL（Secure Socket Layer）はこの範疇に入る方式である．

●DH 公開鍵配送方式とその拡張方式

Diffie と Hellman は，RSA 暗号以前に，公開鍵を用いた鍵配送方式を提案しており，DH 公開鍵配送方式と呼ばれている．有限体 F_p 上で構成する方式と，楕円曲線上 $E(q)$ に構成する方式がある．ここでは，有限体 F_p 上の DH 公開鍵配送方式を示すが，楕円曲線上でもほぼ同様に構成できる．

図 -1 に F_p 上の DH 公開鍵配送方式の基本形を示す．図において，p は大きな素数であり，Y_i と x_i の間には $Y_i = \alpha^{x_i} \bmod p$ という関係がある．α は有限体 F_p における原始根である．素数 p が大きいと，α と Y_i から x_i を求めるのは離散対数問題を解くこととなり，困難である．したがって，Y_i を公開情報としても x_i を求めることができず秘密情報とすることができる．

エンティティ A とエンティティ B が暗号通信をする場合は，エンティティ A は公開情報サーバ等から相手の Y_B を入手し，$WK = Y_B{}^{x_A} \bmod p$ を計算する．$Y_B = \alpha^{x_B} \bmod p$ なので，これは $WK = Y_B{}^{x_A} \bmod p = \alpha^{x_B x_A} \bmod p$ となる．これは，A と B に関して対称形なので，エンティティ B が同様に計算した $Y_A{}^{x_B} \bmod p$ に等しくなる．

エンティティ A とエンティティ B 以外は WK を計算することが困難である．正確に言えば，$\alpha^{x_A} \bmod p$ と $\alpha^{x_B} \bmod p$ から $\alpha^{x_B x_A} \bmod p$ を求める問題は Diffie-Hellman 問題として定式化されており，明らかに離散的対数問題より困難ではない．

DH 公開鍵配送方式を基本にした代表例には，Key Exchange Algorithm（KEA）[1]，IEEE P1363-2000[2]，あるいは IKE（Internet Key Exchange）などがある．

●ID に基づく鍵配送方式

ID-KDS（ID-based Key Distribution System），KPS（Key Predistribution System），IBS（ID Based System）など，提案者により呼称がさまざまであるが，ID 情報として公開情報を用いることが特徴である．従来の公開鍵暗号系では公開鍵を自由な値にできず，ランダムな数字になることが多かった．このため，公開鍵の管理が必要となっていた．それを任意の値に設定できるようにしたのが，ID に基づく鍵配送方式である．ただし，鍵生成センタがコアとなる重要な秘密情報を持つことになるので，中小規模のシステム向きである．ここでは，3 つの方式を示す．

ID-KDS（ID-based Key Distribution System）

DH 公開鍵配送方式と RSA 公開鍵暗号系を組み合わせた方式である．データ暗号用鍵が共有できるまでに双方向の予備通信が必要となるが，シンプルな方式である[3]．その概略図を**図 -2** に示す．図において，$s_X = ID_X{}^{-d} \bmod n$ は信頼できる鍵生成センタが生成するエンティティ X の秘密鍵である．その鍵生成センタは RSA 暗号の公開鍵 (e, n) および秘密鍵 (d, n) を持っていて，それで s_X を計算している．このとき $s_X{}^e ID_X \bmod n = 1$ が成り立つため，

$$WK_{AB} = (ID_B x_B{}^e)^{r_A} \bmod n = (ID_B (s_B g^{r_B})^e)^{r_A} \bmod n$$
$$= (ID_B s_B{}^e g^{r_B e})^{r_A} \bmod n = g^{e r_A r_B} \bmod n = WK_{BA}$$

からデータ暗号化鍵が共有できていることが分かる．

KPS（Key Predistribution System）

予備通信を必要としない ID 情報を用いた鍵共有方式である[4]．結託閾値が存在するが，その閾値以下の結託には情報理論的に安全性が保証できる．その代表例が次に示す SKGS（Symmetric key Generation System）である[5]．信頼できる鍵生成センタは秘密の $k \times k$ 次対称行列 G を持ち，各エンティティには秘密の k 次行ベクトル

$s_X = ID_X G$をあらかじめ渡しておく．ここで，ID_XはエンティティXのID情報を表すk次行ベクトルで，ID情報から誰でも計算できる．実用上はID情報に公開の一方向性で変換した結果をID_Xとすることが多い．

エンティティAとエンティティBが鍵を共有する場合，エンティティAは$WK_{AB} = s_A ID_B{}^T$により，鍵を計算する．同様にエンティティBは$WK_{BA} = s_B ID_A{}^T$により，鍵を計算する．これらは

$$WK_{AB} = s_A ID_B{}^T = ID_A G \cdot ID_B{}^T = (ID_A G \cdot ID_B{}^T)^T$$
$$= ID_B G^T \cdot ID_A{}^T = WK_{BA}$$

より，等しい．

この方式では，kエンティティが各自の秘密ベクトルを持ち寄ると，Gが計算できてしまう．これは，$s_X = ID_X G$がk個得られるため，連立1次方程式を解くことにより，Gが計算できるからである．この意味で，kは結託閾値となっている．k未満ならば安全性は情報理論的に保証される．この結託閾値を実質的にあげる改良案がいくつか提案されている．

IBS（ID Based System）

楕円曲線上のPairingを利用した方式である[6]．閾値が存在せず，予備通信も必要としない．信頼できる鍵生成センタは秘密の整数dを持ち，各エンティティにはあらかじめ秘密に楕円曲線$E(F_q)$上の点$S_X = dP_X$を渡しておく．ここで，P_Xは楕円曲線$E(F_q)$上のID情報に依存する点で，ID情報から誰でも計算できる．

エンティティAとエンティティBが鍵を共有する場合，エンティティAは$WK_{AB} = e(S_A, P_B)$を計算する．エンティティBは$WK_{BA} = e(P_A, S_B)$を計算する．ここで，$e(\ ,\ )$はPairingと呼ばれる$G_1 \times G_2$からμ_nへの双線形関数である．G_1とG_2は楕円曲線の部分加法群であり，μ_nはF_qの拡大体の部分乗法群である．

双線形関数は各変数に関して線形となるため，$e(aP, bQ) = e(P,Q)^{ab}$が成立する．したがって，データ暗号化鍵は等しくなる：

$$WK_{AB} = e(S_A, P_B) = e(dP_A, P_B) = e(P_A, P_B)^d$$
$$= e(P_A, dP_B) = e(P_A, S_B) = WK_{BA}$$

鍵無効化技術

鍵管理において，無効鍵の廃棄は重要な技術である．特に問題となるのは，放送番組を配送するようなときに，受信資格のあるユーザだけにコンテンツを配送するために暗号を使う場合である．この場合，資格がなくなったユーザの鍵を無効化する必要がある．また，PKIにおける公開鍵証明書も使えなくなった場合に無効化する必要がある．いずれの場合にしてもそれらの鍵がユーザ側にあるようなローカルなかたちの鍵管理方式において問題となる．一方，センタに鍵をその都度問い合わせる方式ならば，センタが変更／削除しておけばよいので問題ない．

鍵無効化は応用依存性の高い技術であるが，一般的には，無効鍵が生じたときには，それを何らかのかたちでローカルユーザに通知しなければならなくなる．その方法として

ブラックリスト－無効化鍵

ホワイトリスト－有効な鍵

のどちらかを配布する方法が一般的である．しかし，ネットワークが大きくなるとこれらを配布するのは容易ではないため，効率的に実現する方法がいろいろ提案されている．たとえば，前回のリストからの差分を配布する差分法は，通常リスト全体を配布するより伝送量が少なくて済む．ただ，エラーが累積されるので，定期的に全体リストを送るメカニズムは必要である．

参考文献

1）National Security Agencey: SKIPJACK and KEA Algorithm Specification（1998），http://csrc.nist.gov/CryptoToolkit/skipjack/skipjack.pdf

2）IEEE P 1363 Project: Standard Specifications for Public Key Cryptography, http://grouper.ieee.org/groups/1363/P1363/index.html

3）Okamoto, E. and Tanaka, K.: Key Distribution System Based on Identification Information, Journal on Selected Areas in Communication, The Institute of Electrical and Electronics Engineers, Vol.7, No.4, pp.481-485（1989）.

4）Matsumoto, T. and Imai, H. : On the Key Predistribution System: A Practical Solution to the Key Distribution Problem, In Advances in Cryptology-Crypro'87, Lecture Notes in Computer Science 293, pp.185-193（1984）.

5）Blom, R. : An Optimal Class Symmetric Key Generation Systems, In Advances in Cryptology-Eurocrypt'84, pp.335-338（1984）.

6）Sakai, R., Ohgishi, K. and Kasahara, M.: Cryptosystems Based on Pairing, Proc. of SCIS2000, SCIS2000-C20（2000）.

（平成16年9月30日受付）