楕円曲線暗号の整備動向
＋楕円暗号の実装状況

2011年2月23日

筑波大学　金岡 晃

IPA 情報セキュリティ技術動向調査TG（タスク・グループ）

- “広範な情報セキュリティ分野において、継続的に、かつ、質の高い技術情報を収集し続けるため、半期毎に発表会形式の会合を開催し、討議をふまえて調査報告書を作成します。”
  - http://www.ipa.go.jp/security/outline/committee/isec_tech1.html
- 2010年上期で金岡が「楕円曲線暗号の整備動向」を書きました
  - http://www.ipa.go.jp/security/fy22/reports/tech1-tg/a_01.html

楕円曲線暗号整備の背景（１）

- 公開鍵暗号と言えばRSA暗号
  - 公開鍵暗号の利用されているシーンでは、現在ほぼすべてRSA暗号が使われていると言って良い
  - RSA暗号で使われる鍵のサイズは、現在1024ビットや2048ビットが主流である。
- 楕円曲線暗号
  - 楕円曲線暗号は、楕円曲線利用し、曲線上の点の演算により定義される暗号方式の総称である。
  - 楕円曲線上でDiffie-Hellman（DH）鍵共有を行う楕円DH（ECDH）方式や、楕円曲線上でDigital Signature Algorithm（DSA）を実現する楕円DSA（ECDSA）方式などがある。
  - RSA暗号と比較し、鍵サイズが小さいことが特長であり、ポストRSA暗号として注目されている

楕円曲線暗号整備の背景（２）

- NSA Suite B
  - 2005年、米国家安全保障局（NSA）は機密情報の保護に利用される暗号アルゴリズムのセットSuite Bを発表した
  - 公開鍵暗号のアルゴリズムにRSAはなく、鍵交換はECDH（256または384ビット素体）、電子署名はECDSA（256ビットまたは384ビット素体）が指定されている。
  - その後、Suite Bに合わせた仕様が策定されて来た
    - RFC 4869、5430、5008等
  - またSuite Bの実装ガイド（ECDH版, ECDSA版）も公開されている

2010年上半期の動向

- NSA Suite B Implementer ‘s Guide to FIPS 186-3
- IETFにおける3つの新規RFCと3つの改訂・更新RFC
- OpenSSL 1.0.0のリリース

NSA Suite B Implementer ‘s Guide to FIPS 186-3

- 2010年2月3日発行
- Suite Bに入っているECDSAの実装ガイド
  - NIST FIPS 186-3に定められているECDSAを中心に
- Suite BのECDSA実装に必要となる仕様がそれぞれ抜粋し構成されている
  - ECDSA仕様のうちSuite Bに関するもの
    - P-256とP-386の2つのパラメータ
  - ECDSAアルゴリズムそのもの
    - ANS X9.62
  - 公開鍵の検証
    - NIST SP 800-56A

IETFでの関連仕様（RFC）2010年以降発行

- 新規
  - RFC 5639:
    - Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation （March 2010）
  - RFC 5759
    - Suite B Certificate and Certificate Revocation List (CRL) Profile （January 2010）
  - RFC5915
    - Elliptic Curve Private Key Structure （June 2010）
  - RFC 6090
    - Fundamental Elliptic Curve Cryptography Algorithms (Feb. 2010)
- 改訂（Updates & Obsoletes）
  - RFC 5903
    - Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2 （June 2010）<旧 4753>
  - RFC 5753
    - Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS) （January 2010）<旧3278>
  - RFC 5758
    - Internet X.509 Public Key Infrastructure:Additional Algorithms and Identifiers for DSA and ECDSA （January 2010）<旧3279>

新規RFC

- RFC 6090: Fundamental Elliptic Curve Cryptography Algorithms（Feb. 2011）
  - 基本的な楕円曲線暗号のアルゴリズム
  - ECDH、EC ElGamal署名（KT-I）
- RFC 5915: Elliptic Curve Private Key Structure （June 2010）
  - 楕円曲線暗号のPrivate鍵フォーマット
- RFC 5639: Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation （March 2010）
  - 楕円曲線パラメータの仕様
- RFC 5759: Suite B Certificate and Certificate Revocation List (CRL) Profile （January 2010）
  - Suite B対応のX.509v3証明書プロファイルとX.509v2 CRLプロファイル

更新・改訂RFC （Updates & Obsoletes）

- RFC 5903: Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2 （June 2010）
  - Obsoletes: 4753
  - IPsecで利用されるIKEとIKEv2用の楕円曲線群
  - erratの修正
- RFC 5753: Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS) （January 2010）
  - Obsoletes: 3278
  - CMSでのECCの利用
  - 参照仕様の変更、利用可能ハッシュ関数の拡大（SHA2対応）
- RFC 5758: Internet X.509 Public Key Infrastructure:Additional Algorithms and Identifiers for DSA and ECDSA （January 2010）
  - Updates: 3279
  - SHA2系をつかったDSAとECDSAのアルゴリズムのOID仕様

OpenSSL 1.0.0のリリースとECC関連

- 2010年3月29日にリリース
- ECC自体の鍵生成や証明書発行は0.9.8でも可能だった
- ECC関連の暗号スイートはALLで呼んでもリストアップされず”ECCdraft”を付ける必要があった
  - 1.0.0からはALLで呼ばれるようになり、ECCdraftはなくなった

実装の公開鍵証明書の取り扱い

- 鍵生成と証明書発行
  - OpenSSL
  - Windows Server
- 証明書の利用
  - OpenSSL s_server
  - ブラウザ

パラメータの種類：NIST

- 楕円曲線のパラメータ
  - 体の構成（Prime,Binary）、サイズ、曲線
- NIST推奨パラメータ
  - FIPS 186-3 Appendix D
  - ちなみにECDSA 。ECCはない。

Curves over Prime Fields
P-192
P-224
P-256
P-384
P-521

Curves over Binary Fields	
K-163	K-409
B-163	B-409
K-233	K-571
B-233	B-571
K-283	
B-283	

パラメータの種類：SECG

- SEC2: Recommended Elliptic Curve Domain Parameters
  - www.secg.org/download/aid-784/sec2-v2.pdf

Curve over Fp
secp192k1
secp192r1
secp224k1
secp224r1
secp256k1
secp256r1
secp384r1
secp521r1

Curve over F_2^m	
sect163k1	sect283k1
sect163r1	sect283r1
sect163r2	sect409k1
sect233k1	sect409r1
sect233r1	sect571k1
sect239k1	sect571r1

RFC 4492によるマッピング

SECG	ANSI X9.62	NIST
sect163k1		NIST K-163
sect163r1		
sect163r2		NIST B-163
sect193r1		
sect193r2		
sect233k1		NIST K-233
sect233r1		NIST B-233
sect239k1		
sect283k1		NIST K-283
sect283r1		NIST B-283
sect409k1		NIST K-409
sect409r1		NIST B-409
sect571k1		NIST K-571
sect571r1		NIST B-571
secp160k1		
secp160r1		
secp160r2		
secp192k1		
secp192r1	prime192v1	NIST P-192
secp224k1		
secp224r1		NIST P-224
secp256k1		
secp256r1	prime256v1	NIST P-256
secp384r1		NIST P-384
secp521r1		NIST P-521

利用可能なパラメータ種類

- CNGは3種類
  - NISTパラメータのP-256、P-384、P-521
- OpenSSLがとにかく多い
  - openssl ecparam -list_curves
  - 67種類 （別のファイルで）

OpenSSLで鍵をつくる

- 利用したOpenSSLは1.0.0
  - 最新版ではないです（当時は最新版でした）
- コマンド例
  - RSA鍵
    - openssl req　-x509 -nodes -days 365　-newkey rsa:4096 -sha256 -keyout rsaroot.key -out rsaroot.pem
  - ECC
    - openssl ecparam -out eckey.ecparam -name secp384r1
    - openssl req -new -509 -nodes -days 3650 -newkey ec:eckey.ecparam -sha256 -keyout ecroot.key -out ecroot.pem
- 結果
  - 67種類のうち、2種類のみ失敗
    - Oakley-EC2N-3
    - Oakley-EC2N-4
- つくった鍵
  - ここに全部置きました
  - http://www.cipher.risk.tsukuba.ac.jp/~kanaoka/ecctest/keys/

OpenSSLでCSRを作る

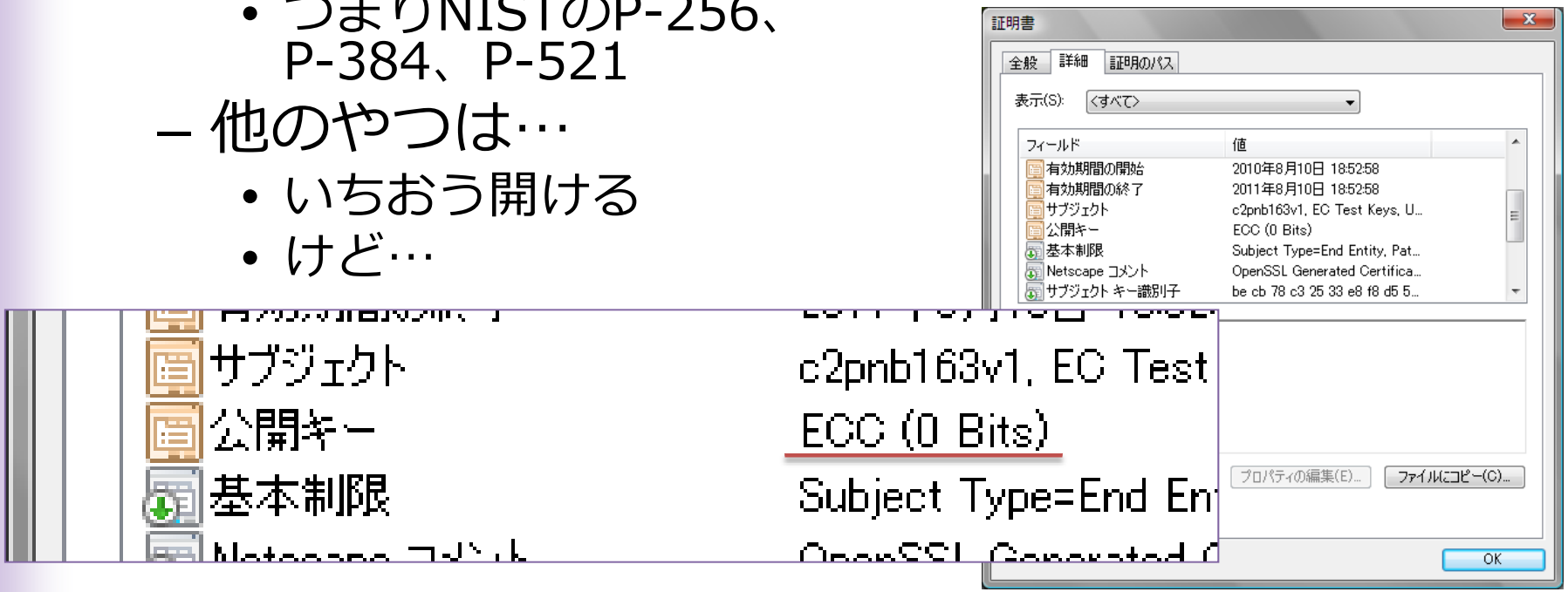
- コマンド例
  - openssl ecparam -out eckey.ecparam -name secp256k1
  - openssl req -nodes -new -newkey ec:eckey.ecparam -keyout ecMCA_byrsa.key -out ecMCA_byrsa.csr
- 結果
  - 67種類のうち、2種類のみ失敗
    - Oakley-EC2N-3
    - Oakley-EC2N-4
- つくったCSR
  - ここに全部置きました
  - http://www.cipher.risk.tsukuba.ac.jp/~kanaoka/ecctest/csrs/

OpenSSLで証明書発行（１）

- OpenSSLで作成した65種類のCSRに対して実行
- ルートCAと中間CAを用意し、中間CAから証明書発行
  - ルートCA、中間CAともにprime256v1の証明書
- 結果
  - 65種類とも成功
- つくった証明書
  - ここに全部置きました
  - http://www.cipher.risk.tsukuba.ac.jp/~kanaoka/ecctest/certs1/

OpenSSLで証明書発行（２）

- 65種類の証明書をWindows（Vista）で見てみると
  - 3種のパラメータで問題なく見える
    - prime256v1, secp384r1, secp521r1
    - つまりNISTのP-256、P-384、P-521
  - 他のやつは…
    - いちおう開ける
    - けど…



OpenSSLで証明書発行（３）

- CAと中間CAの暗号アルゴリズムを変えて、パターンを複数作る
  - CA証明書：ECC or RSA
  - 中間CA証明書：ECC or RSA
  - クライアント証明書：ECC or RSA
- ECCの証明書はprime256v1を利用
- Windowsで証明書ファイルを見てみる
  - いずれも問題なく見える
- つくった証明書
  - ここに全部置きました
  - http://www.cipher.risk.tsukuba.ac.jp/~kanaoka/ecctest/certs2/
  - 命名ルール
    - *root.cer はルート証明書
    - *MCA*.cer は中間CA証明書
    - XXX_byYZ.cerがクライアント証明書
      - XXX：rsa or ec、クライアントの鍵の暗号アルゴリズム
      - Y： r or e、中間CAの鍵の暗号アルゴリズム
      - Z： r or e、ルートCAの鍵の暗号アルゴリズム
      - 例：ec_byre.cer →　ルートCAがECC鍵、中間CAがRSA鍵、クライアントがECC鍵

Windows Serverでの鍵生成・CSR生成・証明書発行

- 鍵（だけ）とCSRを作る
  - Windows Server上のWebアクセスツール（？）を利用
  - 作成できず。プロバイダにECCがない
- 自己署名証明書なら作成できた
- 証明書発行する
  - OpenSSLでつくったCSR群
  - 限られたやつだけ発行できた
    - prime256v1, secp384r1, secp521r1
    - つまりNISTのP-256、P-384、P-521

クライアント側：ブラウザ（１）

- サーバはOpenSSLのs_serverという手もあるが…実際に動いているWebサーバへ。
  - https://comodoecccertificationauthority-ev.comodoca.com/
  - パラメータ
    - たぶんsecp384r1 (P-384)
  - ＃自環境ではapacheで動かなかった…
- 結果
  - IE8 （8.0.6001.18928）
    - O.K.
  - Chrome 5.0.375.86
    - O.K.
  - Firefox 3.5.10
    - O.K.
  - Opera 10.54 （Win32 Windows NT 6.0）
    - アウト。クライアントのスイートに入っていない。
  - Safari 5.0（7533.16）
    - O.K.

クライアント側：ブラウザ（２）

- こんどはOpenSSLのs_serverを利用
- 利用する曲線はOpenSSLで利用可能かつ証明書発行が可能だった65種類
- 結果
  - Operaは全部アウト（スイートに入ってない）
  - IE、Firefox、Chrome、Safariは以下の曲線を使った鍵（の証明書）だとO.K.だった
    - prime256v1
    - secp384r1
    - secp521r1

証明書発行サービス

- たぶんない？