Methoden für das DLP und DDH

Gruppenordnung $\ell = c\ell_0$, ℓ_0 größter Primfaktor von ℓ.

Generische Methoden (Shanks BSGS, Pollard rho, ...):

- Laufzeit Theta($\sqrt{l_0}$), also exponentiell in $\log(\ell_0)$.

Index Calculus Methoden für $\mathbb{F}_q^\times$ mit $\ell \mid (q-1)$:

- Laufzeit $L_q(1/2, c)$, c Konstante.
- In speziellen (und den wichtigsten) Fällen $L_q(1/3, d)$, d Konstante.

Daher Sicherheit im Verhältnis zur Effizienz bei $\mathbb{F}_q$ nicht wesentlich besser als bei RSA ...

Elliptische Kurven

Sei $K = \mathbb{F}_q$ mit $q = p^r$ und $p > 3$.
Eine elliptische Kurve wird durch eine Gleichung gegeben:
$E : Y^2 = X^3 + aX + b$ mit $a, b \in K$ und $4a^2 + 27b^3 \neq 0$.

Menge der Punkte über K: $E(K) = \{(x,y) \in K^2 \mid y^2 = x^3 + ax + b\} \cup \{O\}$.
O ist formal der Punkt „im unendlichen“.

Hasse-Weil: $\#E(K) = q + 1 - t$, wobei $|t| \leq 2\sqrt{q}$.

Man kann $E(K)$ in eine abelsche Gruppe mit neutralem Element O machen. Gruppengesetz wird üblicherweise additiv geschrieben. Es gibt spezielle Formeln, mit der die Punkte „addiert“ werden.

Elliptische Kurven

Man geht davon aus, daß das effizienteste Verfahren für das DDH in einer Untergruppe G von großer Primzahlordnung der Punktgruppe $E(\mathbb{F}_q)$ einer elliptischen Kurve mit zufällig gewählten a, b das Pollard rho Verfahren ist.

Eine elliptische Kurve bietet somit maximal mögliche Sicherheit im Rahmen der gruppenbasierten Kryptographie.

Probleme/Fragen:

- Ordnung von $\#E(K)$ ($\Rightarrow$ Punkte zählen, Kurven konstruieren).
- Spezialfälle, in denen $E(K)$ unsicher ist.
- Optimierungen in Bandbreite und Rechnen (z.B. Punktkompression).

Punkte zählen

Zum Rechnen und wegen Pohlig-Hellman möchten wir $E(K)$ kennen.
Wissen nur $\#E(K) = q + 1 - t$, und $|t| \leq 2\sqrt{q}$.

Es gibt einige Algorithmen zum Punktezählen. Aktuelles Forschungsgebiet.

Laufzeit:

- Kleines p, also $p = O(1)$: $O^\sim(\log(q)^2)$.
- Großes p: $O^\sim(\log(q)^4)$.

Ist $\#E(K)$ berechnet, so kann man kleine Faktoren durch Probedivision herausdividieren und auf den Kofaktor dann einen Primzahltest (Miller-Rabin) anwenden.

Kurven konstruieren

Ein anderer Ansatz ist, elliptische Kurven so zu konstruieren, daß $\#E(K)$ a priori bekannt ist.

Subfield Kurven:

- Ist E über $\mathbb{F}_q$ definiert und $\#E(\mathbb{F}_q)$ bekannt, so kann man leicht $\#E(\mathbb{F}_{q^n})$ für alle n ausrechnen.

Komplexe Multiplikation:

- Mit weitergehender Mathematik kann man zu vorgegebener Punktanzahl direkt eine Kurve E konstruieren.

Etwas nachteilig ist hier - nur aus philosophischer Sicht -, daß die Kurven nicht zufällig gewählt werden. Dies könnte Möglichkeiten für spezielle Angriffe eröffnen (nichts wesentliches bekannt).

Unsichere Spezialfälle

Multiplikativer Transfer:

- auch Frey-Rück Reduktion (Menezes-Okamoto-Vanstone Angriff).
- Seien $\gcd\{\ell, q\} = 1$ und μ_ℓ die ℓ-ten Einheitswurzeln in $\mathbb{F}_{q^k}$ mit $\ell \mid (q^k - 1)$ und k minimal. $G = E(\mathbb{F}_q)[\ell]$ Untergruppe der Ordnung ℓ.
- Mit Hilfe der Tate-Paarung kann man einen Isomorphismus $E(\mathbb{F}_q)[\ell] \to \mu_\ell$ definieren, der in Zeit poly$(\log(q))$ berechnet werden kann.
- Man kann also ein DLP von $E(\mathbb{F}_q)[\ell]$ nach $\mathbb{F}_{q^k}^\times$ transferieren und dort subexponentiell lösen.
- Für von q unabhängiges, zufälliges ℓ ist k meist von der Größenordnung wie ℓ, somit der Angriff nicht durchführbar.
- Speziell für supersinguläre Kurven ($t = 0 \bmod p$) kann man jedoch immer $k \leq 6$ erreichen.
- Man sollte immer prüfen, ob zu q und ℓ der Exponent $k \geq 20$ ist.

Unsichere Spezialfälle

Additiver Transfer (für anomale Kurven, also $t = 1$ bzw. $\#E(\mathbb{F}_q) = q$):

- auch Rück oder SmartASS Angriff, additive Version der FR Reduktion. Hier $\ell = p$ und meistens $q = p$.
- Es gibt einen Isomorphismus $E(\mathbb{F}_q)[p] \to \mathbb{F}_p^+$, der in poly$(\log(q))$ berechnet werden kann.
- Man kann also ein DLP aus $E(\mathbb{F}_q)[p]$ nach $\mathbb{F}_p^+$ transferieren und dort in Polynomzeit lösen. Durch Iterieren erhält man auch diskrete Logarithmen in $E(\mathbb{F}_q)$ (wie bei Pohlig-Hellman).

Anomale Kurven sind also besonders unsicher, während zum Beispiel supersinguläre Kurven nur eine reduzierte Sicherheit (subexponentiell) bieten und damit verwendbar bleiben. Siehe identitätsbasierte Systeme ...

Unsichere Spezialfälle

Weil Abstieg Techniken.

- Treffen im wesentlichen nur auf $q = 2^n$ und n mit kleinen Faktoren und/oder für spezielle elliptische Kurven zu.
- Konstruieren eine höher-geschlechtige Kurve, die aber über einem Teilkörper von $\mathbb{F}_q$ definiert ist.
- Das DLP kann in die Picardgruppe dieser Kurve transferiert werden. Hierin kann man einen Index-Calculus Angriff durchführen.
- Allgemeine Abhilfe: n prim wählen, zufällige Kurven verwenden.

Zusammenfassung Sicherheit

Kriterien für die Sicherheit einer elliptischen Kurve:

1. Punktanzahl $\#E(\mathbb{F}_q)$ enthält einen ausreichend großen Primfaktor ℓ, um gegen Pohlig-Hellman und Pollard rho zu schützen.
2. Es gilt nicht $\#E(\mathbb{F}_q) = q$, um vor Rück bzw. SmartASS Angriff zu schützen.
3. Das kleinste k mit $\ell \mid (q^k - 1)$ erfüllt $k \geq 20$, um gegen FR-Reduktion bzw. MOV-Angriff zu schützen.
4. Mit $q = p^n$ sollte n prim oder 1 sein, wegen Weil Abstieg Angriff.

Normalerweise wird nur $q = 2^n$ oder $q = p$ verwendet.

Für identitätsbasierte Systeme verzichtet man auf 3. (und 4.) und erhält zusätzliche Funktionalität bei reduzierten Sicherheits-Effizienz Verhältnis (hier häufig supersingulär und $q = 3^n$, da dann $k = 6$ möglich).

Zusammenfassung Sicherheit

Will man „besonders sicher“ gehen, sollte man die elliptischen Kurven mit zufälligem a, b erzeugen, um keine „speziellen“ Eigenschaften zu erzeugen.

Wie kann man jemanden davon überzeugen, daß man eine Kurve zufällig erzeugt hat? Kurve beweisbar zufällig erzeugen ...

- Man nimmt z.B. $a =$ SHA-1(Zahl-1) und $b =$ SHA-1(Zahl-2), wobei die Zahlen zu variieren sind, bis die resultierende Kurve einen großen Primfaktor enthält.
- Man veröffentlicht dann die Zahlen 1, 2 zur Berechnung von a, b.

In Standards werden die zu verwendenden Kurven häufig speziell vorgegeben (IPSec). Obiger „Sicherheitstest“ bei RSA nicht möglich.

Optimierungen

Man kann im Hinblick auf die Effizienz der erforderlichen Rechnungen in $E(\mathbb{F}_q)$ eine ganze Reihe von Optimierungen durchführen.

1. Gruppengesetz:
   - Verschiedene Koordinatensysteme für Punkte und optimierte Formel für das Gruppengesetz.

2. Punktvielfache:
   - Die Operation λP ist die teuerste Operation beim Verschlüsseln.
   - Man verwendet Varianten von Double-und-Add.
   - $-P$ kann besonders schnell ausgerechnet werden. Daher sollte man beim Double-und-Add auch Subtraktionen in Betracht ziehen.

Optimierungen

3. Punktkompression:
   - Für P ist y_P Nullstelle einer quadratischen Gleichung, die nur von E und x_P abhängt. Da es stets nur zwei solche Nullstellen gibt, genügt ein Bit zur Auswahl der Nullstelle.
   - Speicher- u. Kommunkationsbedarf für einen Punkt halbiert sich.

Gehen im folgenden exemplarisch etwas näher auf 2. und 3. ein.

Punktvielfache

Double-and-Add als Hornerschema:

- $m = \sum_{i=0}^{r} m_i 2^i, \ \ m_i \in \{0,1\}$ binäre Entwicklung.
- $[m]P = 2\big(\cdots \big(2\big(2[m_r]P + [m_{r-1}]P\big) + [m_{r-2}]P\big) \cdots + [m_1]P\big) + [m_0]P.$

Invertieren effizient, verwende daher allgemeiner $m_i \in \{0, \pm 1\}$.
$\Rightarrow$ Non-adjacent form (NAF), $m_i m_{i+1} = 0$ möglich.

Berechnung der NAF (im i-ten Schritt):

- Wenn $m \equiv 0 \bmod 4$, dann bereits $m_i = m_{i+1} = 0$. Wenn $m \equiv 2 \bmod 4$ dann $m_i = 0$ und $m_{i+1} \neq 0$.
- Ansonsten wähle $m_i \in \{-1, 1\}$ mit $m - m_i \equiv 0 \bmod 4$. Dann $m_{i+1} = 0$.

NAF höchstens ein Bit länger als binäre Entwicklung.
Durchschnittliche Dichte der binären Entwicklung $1/2$, der NAF $1/3$.
Binäre Entwicklung $(r/2)$ Adds + r Doubles, NAF $(r/3)A + rD$.

Punktvielfache

$E(\mathbb{F}_q)[\ell]$ zyklisch der Ordnung ℓ, $\ \phi \in$ End(E) durch algebraische Formeln auf Koordinaten definiert, liefert Element von End$(E(\mathbb{F}_q))$.

$\Rightarrow$ Es gibt $\lambda \in \mathbb{Z}$ mit $\phi(P) = [\lambda]P$ für alle $P \in E(\mathbb{F}_q)[\ell]$.

Schreibe $m = \sum_{i=0}^{r-1} m_i \lambda^i \bmod \ell$. Dann

$$\begin{aligned}[m]P = [\sum_{i=0}^{r-1} m_i \lambda^i]P &= \sum_{i=0}^{r-1} m_i [\lambda]^i(P) = \sum_{i=0}^{r-1} m_i \phi^i(P) \\ &= [m_0]P + [m_1]\phi(P) + \cdots + [m_{r-1}]\phi^{r-1}(P) \\ &= \phi\big(\cdots \big(\phi\big(\phi([m_{r-1}]P) + [m_{r-2}]P\big) + [m_{r-3}]P\big) \cdots + [m_1]P\big) + [m_0]P.\end{aligned}$$

Punktvielfache

Finde geeignete, effizient berechenbare ϕ:

- Frobeniusendomorphismus $(x_P, y_P) \mapsto (x_P^q, y_P^q)$ für Subfieldkurven über $\mathbb{F}_{q^n}$.
- Endomorphismen durch komplexe Multiplkation (siehe Konstruktion).

Dann zwei Hauptfälle:

- $r = 2$, $m_i = O(\sqrt{\ell}) \Rightarrow$ Benutze simultane Multiexponentiation.
- $r \approx \log_d(\ell)$, $|m_i| \lesssim d$, $d = O(1) \Rightarrow$ Benutze Horner Schema wie vorige Folie.

Liefert sogenannte ϕ-Entwicklungen. Weitere Methoden:

- ϕ-NAF, Sliding Window, ...

Punktkompression

Sei $P = (x_P, y_P)$ mit $y_P^2 = x_P^3 + a x_P + b$ in $\mathbb{F}_q$ (p ungerade).
Dann ist $Q = (x_P, -y_P)$ ebenfalls ein Punkt auf E.
Es gibt keine weiteren Punkte mit derselben x-Koordinate x_P.

Gegeben x_P, wie zwischen y_P und $-y_P$ mit Hilfe eines Bits unterscheiden? Verschiedene Möglichkeiten:

- Bit gibt an, ob y-Koordinate ein Quadrat oder nicht ist (Jacobi Symbol).
- Bit gibt an, ob lexikographisch größere oder kleinere y-Koordinate zu wählen ist.
- Praxis: Für $q = p$ wählt man als Bit das LSB(y_P) (least signifikant bit). Für $0 \le y_P < p$ wird $-y_P$ durch $-y_P + p$ repräsentiert. LSB$(y_P) = 0 \Leftrightarrow$ LSB$(-y_P) = 1$, da p ungerade. Daher klappt's.

Speicher- und Übertragungsersparnis von ca. 50% (Patentgeschützt).

Parametervergleich

NIST Tabelle, Schlüsselgrößen bei ungefähr gleicher Sicherheit:

Block Chiffre Schlüsselgröße	Beispiel Block Chiffre	ECC Schlüsselgröße	RSA / $\mathbb{F}_q^\times$ Schlüsselgröße
80	SKIPJACK	163	1024
128	AES (klein)	283	3072
192	AES (mittel)	409	7680
256	AES (groß)	571	15360

ECC mit 517 praktikabel, RSA / $\mathbb{F}_q^\times$ mit 15360 nicht.

Orientierung

Haben bisher im Public-Key Bereich nur Verschlüsselung betrachtet.

Haben dafür geeignete mathematische Strukturen und ihre Eigenschaften diskutiert.

- RSA, Rabin: Restklassenringe modulo n, Potenzieren als Einwegfunktionen mit Falltür ($x \mapsto x^e$).
- Zyklische Gruppen von Primzahlordnung, Exponieren als Einwegfunktion ohne Falltür ($x \mapsto g^x$).

Diese Strukturen und Einwegfunktionen können nun auch für weitere kryptographische Aufgabenstellungen verwendet werden.

- Digitale Unterschriften.
- Schlüsselaustausch (von symmetrischen Schlüsseln).
- ...

Digitale Unterschriften

Auch digitale Signaturen genannt.

Nachrichten aus Nachrichtenraum: $M \in \mathcal{M}$.
Signaturen aus Signaturenraum: $\sigma \in S$.
Schlüssel sind aus Schlüsselräumen: $d \in K_1$, $e \in K_2$.

Signierungsverfahren $\mathcal{S} : K_1 \times \mathcal{M} \rightsquigarrow S$.
Verifizierungsverfahren $\mathcal{V} : K_2 \times \mathcal{M} \times S \to \{0,1\}$.

Signatur von Nachricht m mit Schlüssel d: $\sigma \leftarrow \mathcal{S}(d,M)$.
Verifizierung von M,σ mit Schlüssel e: $f \leftarrow \mathcal{V}(e,M,\sigma)$.

e öffentlicher Schlüssel, d privater Schlüssel.

Bemerkungen

$\mathcal{S}$ und $\mathcal{V}$ sind effiziente Verfahren (z.B. Programme).
$\mathcal{S}$ und $\mathcal{V}$ dürfen probabilistisch sein (dürfen den Zufall verwenden).
$\mathcal{S}$ kann mehrdeutig sein ($\rightsquigarrow$).

Im allgemeinen wird nur ein Hashwert $H(M)$ und nicht M selbt signiert.

- Effizienter, da $H(M)$ viel kürzer als M ist.
- Beweisbare Sicherheit von in der Praxis relevanten Verfahren (allerdings im Zufallsorakelmodell, RO).

Offenbar muß H kollisionsfrei sein, man kann keine zwei Nachrichten M_1, M_2 mit $H(M_1) = H(M_2)$ berechnen.

Signatur mit Einwegfunktion mit Falltür

Seien $f,h : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^m$ derart, daß es für jedes (zufällige) $e \in \{0,1\}^n$ ein $d \in \{0,1\}^n$ gibt, so daß die Funktion $f(e,\cdot)$ eine Einwegfunktion mit Falltür d und inverser Funktion $h(d,\cdot)$ ist.

Beispiel: RSA Funktionen $x \mapsto x^e$, $y \mapsto y^d$.

Dann für $M \in \{0,1\}^m$:

- Signieren: $\sigma = \mathcal{S}(d,M) = h(d,M)$, Signatur ist (M,σ).
- Verifizieren: $\mathcal{V}(e,M,\sigma) = 1$ genau dann, wenn $f(e,\sigma) = M$.

„Sicherheit" basiert informell also darauf, daß niemand $\sigma = h(d,M)$ ausrechnen kann, ohne d zu kennen.

Signierer beweist, daß er die geheime Information kennt.