Zentrum für sichere Informationstechnologie – Austria
Secure Information Technology Center – Austria

A-1040 Wien, Weyringergasse 35
Tel.: (+43 1) 503 19 63–0
Fax: (+43 1) 503 19 63–66

A-8010 Graz, Inffeldgasse 16a
Tel.: (+43 316) 873-5514
Fax: (+43 316) 873-5520

http://www.a-sit.at
E-Mail: office@a-sit.at

Kryptosysteme basierend auf Elliptischen Kurven Einsatz und Verbreitung in Standardsoftware

Wolfgang Bauer

29. Juli 2009

Zusammenfassung

Kryptosysteme basierend auf elliptischen Kurven (ECC) sind trotz ihrer herausragenden technischen Eigenschaften bis vor Kurzem kaum in Standardanwendungen zum Einsatz gekommen. In letzter Zeit scheint sich jedoch eine Trendwende abzuzeichnen. So setzt beispielsweise die US-Regierung bei Belangen mit Nationaler Sicherheit ausschließlich auf ECC bei asymmetrischer Kryptographie (bei den öffentlich publizierten Algorithmen). Dieses Dokument untersucht inwieweit elliptische Kurven heute schon in typischen Endanwenderprodukten integriert sind. Dabei werden exemplarisch Internetbrowser und E-Mail Programme unter Microsoft Windows und Linux herangezogen. In diesem Kontext wird auch die Situation der österreichischen Bürgerkarte, die in ihrer Ausprägung als e-card und als Bankomatkarte auf ECC setzt, nochmals betrachtet.

Versionshistorie

Version	Datum	Autor(en)	Änderungen
1.0	03.03.08	Wolfgang Bauer	Erste Version
1.1	22.07.09	Wolfgang Bauer	Betriebssysteme: Windows 7 hinzugefügt. SSL/TLS: Safari, Google Chrome und Opera als neue Browser aufgenommen. S/MIME: keine Änderungen.

Inhaltsverzeichnis

1 Einleitung

Österreich hat bei der Einführung der Bürgerkarte (z.B. in der Ausprägung als Bankomatkarte oder auch als e-card) schon sehr früh auf Kryptosysteme basierend auf elliptischen Kurven (ECC) gesetzt. Zum damaligen Zeitpunkt war die Verbreitung von ECC in Standardsoftware noch nicht sehr weit verbreitet. ECC scheint sich jedoch immer mehr zu etablieren, und auch in Produkten für Endanwender Einzug zu halten. Vor allem prominente amerikanische Softwarehersteller setzten verstärkt auf diese Technologie. Dieser Schritt scheint auch durch die Veröffentlichung der Suite B [6], durch die National Security Agency (NSA), motiviert zu sein.

Die Suite B beinhaltet kryptographische Algorithmen für Verschlüsselung, Signaturen, Schlüsselaustausch und Hashverfahren. Dabei setzt die NSA bei asymmetrischen Verfahren ausschließlich auf ECC. Das bedeutet dass für Belangen mit Nationaler Sicherheit (neben der nicht veröffentlichten Suite A) in den USA in Zukunft ausschließlich ECC zum Einsatz kommen wird. Die Suite B kann aber auch für andere Bereich eingesetzt werden, wie das folgende Zitat aus [6] unterstreicht:

> Because Suite B is also a subset of the cryptographic algorithms approved by the National Institute of Standards, Suite B is also suitable for use throughout government.

Diese Tatsache und die Marktdominanz der US-Softwareindustrie scheinen die Verbreitung von ECC stark voranzutreiben. Dieser Artikel analysiert die derzeitige Verbreitung von Kryptosystemen basierend auf elliptischen Kurven. Dabei wird vor allem untersucht wie ECC in gängigen Betriebssystemen und Anwendersoftware integriert ist. Vor diesem Hintergrund wird die damalige Entscheidung Österreichs, sehr früh auf ECC zu setzen, nochmals beleuchtet.

1.1 Eigenschaften von ECC

Durch die Vorreiterrolle Österreichs bei der Einführung von ECC, hat sich auch das Zentrum für sichere Informationstechnologie – Austria (A-SIT) schon intensiv mit ECC beschäftigt. Auf die technischen bzw. mathematischen Grundlagen sei daher auf die von A-SIT betreute, 2001 erstellte Studie [7] verwiesen und anschließend werden daher nur mehr die wichtigen Eigenschaften nochmals kurz zusammengefasst.

Asymmetrische Kryptosysteme stützen sich auf schweren mathematischen Problemen. Die heutzutage relevanten Verfahren basieren entweder auf dem diskreten Logarithmusproblem, dem RSA Problem oder dem diskreten Logarithmusproblem über elliptischen Kurven (ECDLP). Das einzige dieser Probleme für das bis zum heutigen Zeitpunkt kein Algorithmus mit subexponentieller Laufzeit existiert ist das ECDLP. Deshalb kommen Kryptosysteme die auf diesem Problem beruhen bei gleicher Sicherheit mit bedeutend kürzeren Schlüssellängen aus. Um beispielsweise das Sicherheitsniveau einer symmetrischen 128 Bit AES Verschlüsselung zu erreichen, muss man einen 3072 Bit RSA Schlüssel verwenden. Bei ECC reicht hingegen ein ca. 256 Bit Schlüssel [5]. Zumindest auf Geräten mit limitiertem Speicher und Rechenleistung, wie beispielsweise Smartcards, stellen lange RSA Schlüssel ein technologisches Problem dar. Serverseitig können ECC basierende Systeme extrem performant umgesetzt werden. Dies zeigt auch eine A-SIT Studie aus dem Jahre 2002 [8]. Dabei wurde deutlich das Potential von hardwareunterstützen ECC Lösungen hinsichtlich Performance aufgezeigt.

Aus dieser Überlegung heraus sieht man, dass vor allem für zukunftsorientierte Anwendungen ECC sehr verlockende Eigenschaften besitzt. Trotzdem war diese Technologie bis vor kurzem hauptsächlich in akademischen Kreisen populär und wurde in der Praxis selten eingesetzt.

Die Entscheidung der NSA in Zukunft ausschließlich auf ECC zu setzen legt die Vermutung nahe, dass die Verbreitung dieser Technologie in Softwareprodukten stark ansteigen wird. Dieser Artikel untersucht die für Endanwender derzeit gängigen Produkte und wie weit die ECC Integration derzeit fortgeschritten ist.

In weiterer Folge werden im nächsten Abschnitt die Betriebssysteme Linux und Windows betrachtet und die Kernelintegration von ECC beurteilt. Anschließend werden Internetbrowser und E-Mail Clients analysiert und in Abschnitt 6 die Ergebnisse nochmals zusammengefasst.

2 Betriebssystemintegration

In diesem Abschnitt wird die Integration in die Betriebssysteme Microsoft Windows und Linux untersucht. Dabei werden nicht Anwendungen und deren Portierung auf die unterschiedlichen Plattformen analysiert, sondern die kryptographischen Funktionen die das Betriebssystem selbst zur Verfügung stellt.

2.1 Microsoft Windows

Bei Windows werden alle kryptographischen Verfahren im sogenannten Cryptographic Service Provider (CSP) gekapselt. Der Zugriff auf diese Funktionen erfolgt über die vom Microsoft vorgegebene Schnittstelle CAPI. Alle Algorithmen die im CSP integriert sind stehen dann sowohl Anwendungen (z.B. zum Signieren von E-Mails) als auch dem Betriebssystem (z.B. für verschlüsselte Dateisysteme, IPSec, etc.) zur Verfügung. Voraussetzung für eine ECC Funktionalität in integrierten Windows Applikationen ist daher die ECC Integration in den CSP. Dieser Schritt wurde von Microsoft mit Windows Vista getan und ist auch als Cryptography API: Next Generation (CNG) bekannt.

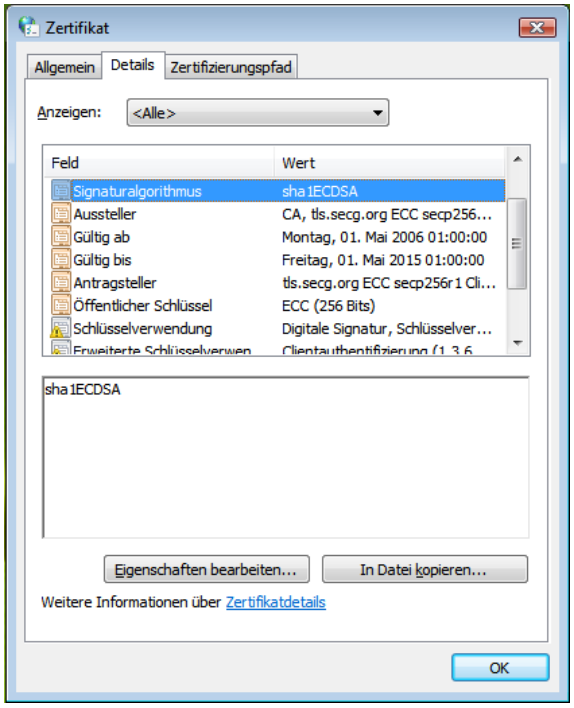

Abbildung 1: Suite B ECC Zertifikat

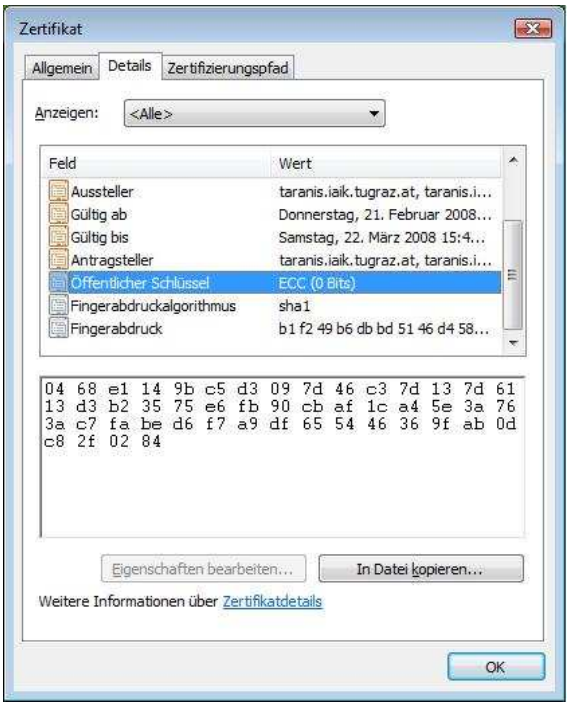

Abbildung 2: 192 Bit ECC Zertifikat

Die Abbildung 1 zeigt ein X.509 Zertifikat wie es unter Windows Vista dargestellt wird. Das Zertifikat wurde von einer Zertifizierungsstelle (CA) mit dem Elliptic Curve Digital Signature Algorithm (ECDSA) unterschrieben. Der im Zertifikat enthaltene öffentliche Schlüssel ist ein 256 Bit ECC Schlüssel aus der Suite B. Wie man erkennen kann, wird das Zertifikat akzeptiert und der Schlüssel richtig erkannt.

Durch diese Unterstützung von ECC im CSP hat Microsoft die Voraussetzung für die Integration in alle Bereiche des Betriebssystems geschaffen. Dies betrifft beispielsweise ver-

schlüsselte Dateisysteme, Virtual Private Network (VPN) Unterstützung über IPSec oder auch die sichere Windows Anmeldung über Kerberos. Diese Funktionen wurden bei dieser Studie nicht explizit evaluiert. Es ist jedoch damit zu rechnen, dass ECC bei diesen Funktionen schon jetzt unterstützt wird oder in absehbarer Zeit verfügbar sein wird.

An dieser Stelle möchten wir noch explizit darauf hinweisen, dass sich die Unterstützung von ECC in Windows Vista ausschließlich auf elliptische Kurven über Primzahlenkörpern mit mindestens 256 Bit konzentriert. Dies beinhaltet alle elliptischen Kurven aus der Suite B. Wenn man versucht ein anderes ECC Zertifikat zu importieren, wie beispielsweise das 192 Bit Zertifikat aus Abbildung 2 (oder auch elliptische Kurven über $GF(2^n)$), so wird dies nicht richtig erkannt. Das Zertifikatsformat wird zwar richtig analysiert, aber der öffentliche Schlüssel mit „ECC(0Bits)“ ausgewiesen. Dies deutet darauf hin, dass dieser Schlüssel nicht verwendet werden kann, was auch im nächsten Abschnitt bestätigt wird.

Die Ergebnisse unter Windows 7 (Evaluierungskopie Build 7100) unerscheiden sich hier nicht von denen mit Windows Vista. Auch hier werden nur Suite B Algorithmen für ECC unterstützt.

2.2 Linux

Unter Linux sieht die Situation etwas anders aus. Dort war lange Zeit kein vergleichbares CSP Konzept im Kernel integriert. Erst mit der Version 2.4.22 wurde ein vergleichbares Konzept, die Cryptographic API, offiziell in den Kernel aufgenommen. Zur Zeit bietet jedoch der Linux Kernel keine Unterstützung für ECC an.

Obwohl Linux keine Kernelunterstützung für ECC bietet, gibt es natürlich trotzdem Anwendungen die diese Algorithmen unterstützen. Die nächsten Abschnitte analysieren die gängigsten davon.

3 TLS/SSL Unterstützung

Secure Socket Layer (SSL) und der Nachfolger Transport Layer Security (TLS) bieten Methoden, um vertraulich und authentifiziert über das Internet zu kommunizieren. So kann beispielsweise HTTP auf SSL aufsetzen (HTTPS), um so sichere Web-Anwendungen zu ermöglichen.

Zu Beginn einer SSL Verbindung authentifizieren sich der Server und (optional) der Client gegenseitig. Dabei werden X.509 Zertifikate verwendet. Abhängig von diesen Zertifikaten wird ein entsprechendes Schlüsselaustauschprotokoll initiiert und ein gemeinsamer symmetrischer Schlüssel ausgehandelt.

Die nächsten Abschnitte analysieren die ECC Unterstützung für die gängigen Browser (Microsoft Internet Explorer, Mozilla Firefox, Safari und Opera).

3.1 Microsoft Internet Explorer

Microsoft Internet Explorer ist der am häufigsten verwendete Browser und steht nur für das Windows Betriebssystem zur Verfügung. Da der Internet Explorer für die kryptographischen Verfahren den im Betriebssystem integrierten CSP nutzt (siehe dazu 2.1) steht ECC Unterstützung für HTTPS erst ab Windows Vista zur Verfügung.

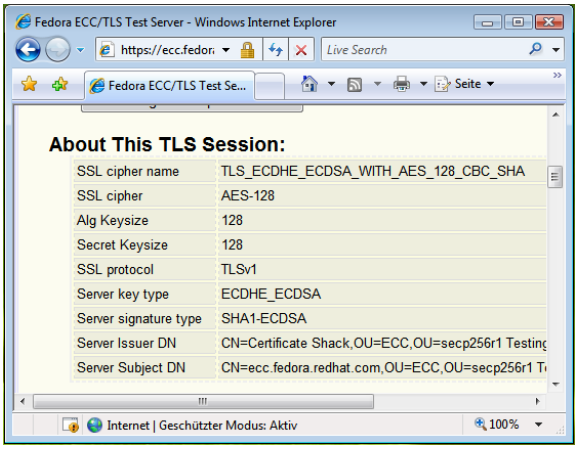

Abbildung 3: HTTPS Verbindung

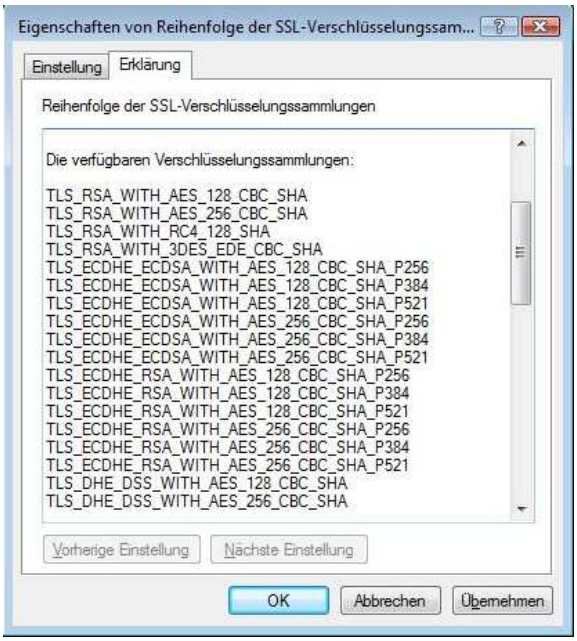

Abbildung 4: Cipher Suites

Die Abbildung 3 zeigt eine sichere HTTPS Verbindung mit dem öffentlichen Testserver `http://ecc.fedora.redhat.com/`. Wie der Screenshot auch zeigt, wurde für die Aushandlung des Schlüssels die Diffie Hellman Variante für elliptische Kurven und der ECDSA als Signaturalgorithmus verwendet. Die Abbildung 4 daneben zeigt die Liste der verfügbaren Cipher Suites unter Vista. Wie man auch aus dieser Liste ablesen kann, werden ausschließlich 256 Bit oder stärkere Mechanismen für ECC angeboten. Der Internet Explorer unterstützt auch HTTPS (SSL) Verbindungen mit Clientauthentifizierung. Damit diese getestet werden kann, muss zuerst ein entsprechendes Clientzertifikat importiert werden. Dies kann beispielsweise am öffentlichen Testserver `http://tls.secg.org` überprüft werden, wo auch entsprechende Clientzertifikate bereitgestellt werden.

Um das Verhalten des Internet Explorers bei nicht Suite B konformen Cipher Suites zu testen und wurden auch stichprobenartig andere Konfigurationen ausprobiert. Die Ergebnisse sind am Ende dieses Artikels in Tabelle 3 für die untersuchten Browser angeführt.

Alle diese Tests zeigen, dass die ECC Integration der Suite B Algorithmen sehr gut in TLS integriert sind. Auch der komplexeste Fall mit zertifikatsbasierender Clientauthentifizierung funktioniert.

3.2 Chrome

Für das Windows Betriebssystem stehen seit kurzem auch der Google Chrome Browser zur Verfügung. Dieser verwendet auch die Unterstützung des Betriebssystem für die cryptographischen Algorithmen und die Zertifikatsverwaltung. Daher ist es auch nicht verwunderlich, dass Google Chrome die selben SSL Cipher Suites wie Microsofts Internet Explorer unterstützt. Zertifikatsbasierte Clientauthentifizierung wird jedoch derzeit nicht unterstützt. Dies ist kein ECC spezifisches Problem sondern betrifft die gesamte SSL Implementierung. Dies ist eine bekannt Einschränkung von Google Chrome.

3.3 Opera

Der Opera Browser verwendet nicht die Betriebssystemunterstützung für SSL sondern setzt auf eine eigene Implementierung. Derzeit sind dort jedoch keine ECC Algorithmen implementiert und daher unterstütz Opera derzeit keine ECC SSL Cipher Suites.

3.4 Safari

Der Safari Browser läuft unter Mac OS und Windows. Da in dieser Studie Mac OS nicht näher analysiert wurde gehen wir in diesem Abschnitt auch ausschließlich auf die Windows Variante von Safari ein. Wie auch der Chrome Browser von Google greift Safari auf die kryptographischen Verfahren vom Windows Betriebssystem zu und unterstützt daher seit Windows Vista die selben Cipher Suites wie der Internet Explorer. Auch Clientauthentifizierung funktioniert mit dem Safari Browser.

3.5 Mozilla Firefox

Ein weiterer populärer Browser ist Mozilla Firefox der Versionen für Linux und Windows bereitstellt. Dieser Browser nutzt eine eigene kryptographische Softwarebibliothek (Network Security Services NSS) und ist daher nicht angewiesen auf die ECC Unterstützung durch das Betriebssystem. Getestet wurde die Version 3.5 dieses Browsers.

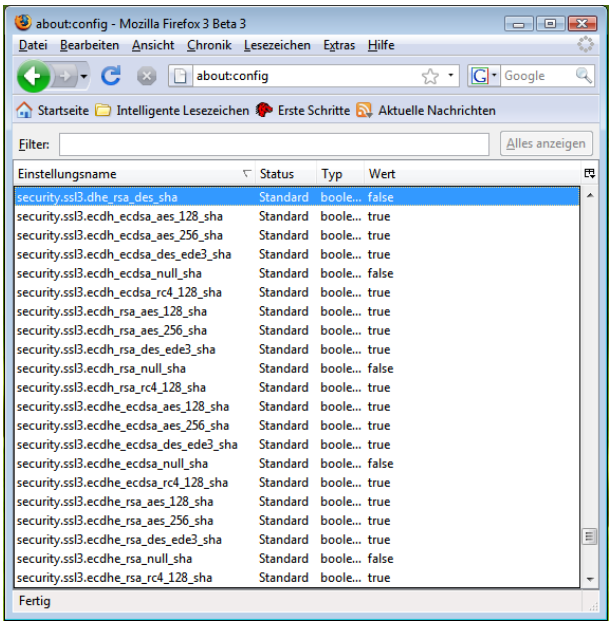

Abbildung 5: Firefox Cipher Suites

Schon die Konfiguration von Firefox (siehe Abbildung 5) zeigt, dass Firefox alle nicht anonymen Cipher Suites beherrscht. In der Praxis wurden die gleichen Tests wie mit dem Internet Explorer durchgeführt. Auch im Firefox sind die ECC Cipher Suites gut integriert und eine TLS Verbindung mit Clientauthentifizierung problemlos möglich. Wie die Tests aber auch gezeigt haben, beschränkt sich die Firefox Unterstützung ausschließlich auf elliptische Kurven über Primzahlenkörpern mit 256 Bit oder darüber. Das bedeutet, dass Kurven mit kürzeren Schlüssellängen (z.B. 192 Bit) auch nicht unterstützt werden. In Tabelle 3 sind die stichprobenartigen Testergebnisse zusammengefasst. Während der Internet Explorer ECC ausschließlich mit AES kombiniert anbietet, also Suite B konform, werden im Firefox alle standardisierten ECC Cipher Suites angeboten. Die Tabelle 2 gibt einen Überblick über alle standardisierten Protokollvarianten und welche Mechanismen von den Browsern unterstützt werden.

Zusammenfassend kann festgestellt werden, dass zumindest die Suite B Algorithmen sehr gut in den gängigen Webbrowsern integriert sind. Auf der Serverseite sieht die Situation ebenfalls gut aus. Gängige Webserver (z.B. Apache mit mod_ssl, IIS 7 und Sun Java

System Web Server 7.0) unterstützen ECC Cipher Suites. Detaillierte Interoperabilitätsergebnisse dieser Server mit den gängigen Browsern findet man unter `http://ecc.fedora.redhat.com/interop/testresults/`. Ein weiteres wichtiges und weit verbreitetes Thema ist die Sicherheit von E-Mails. Dies wird im nächsten Abschnitt analysiert.

4 E-Mail Sicherheit

Um E-Mails sicher, das heißt sowohl authentisch als auch geheim, versenden zu können, kommen in der Praxis zwei Standards zum Einsatz. Auf der einen Seite PGP, das keine hierarchische Public Key Infrastruktur (PKI) einsetzt und hier nicht näher betrachtet wird. Auf der anderen Seite das auf X.509 Zertifikaten aufbauende S/MIME. S/MIME wird von fast allen gängigen E-Mail Clients unterstützt. In diesem Dokument werden zwei weit verbreitete Mailprogramme, Microsoft Outlook und Mozilla Thunderbird analysiert.

4.1 Microsoft Outlook

Microsoft Outlook setzt für die kryptographischen Funktionen wiederum auf den CSP von Windows auf. Daher wird ECC erst ab Windows Vista unterstützt. Der mit Windows Vista ausgelieferte E-Mail Client „Windows Mail“ (der Nachfolger von Outlook Express) basiert auch auf dem CSP und scheint den gleichen Grad an ECC Funktionalität wie Outlook zu bieten. Dies wurde jedoch nicht ausführlich getestet und wird daher auch nicht weiter analysiert. Untersucht wurde Outlook 2007. Wiederum können nur Zertifikate mit 256 Bit [1] oder stärker für diese Untersuchungen verwendet werden. Alle anderen Zertifikate und der dazugehörige private Schlüssel können gar nicht importiert werden.

4.1.1 S/MIME Signaturen

Für diese Tests wurde je ein 521 Bit ECC und ein 256 Bit Zertifikat erstellt und in Windows importiert. Anschließend wurde Outlook so konfiguriert, dass ausgehende E-Mails mit ECDSA/SHA-512 signiert werden.

In Abbildung 6 sieht man den Konfigurationsdialog von Outlook 2007. Hier können auch die Verschlüsselungsoptionen (siehe Abschnitt 4.1.2) eingestellt werden. Die Nachricht kann mit diesen Einstellungen signiert und versendet werden. Auch das Empfangen und Validieren der Signatur funktioniert wie erwartet. Abbildung 7 zeigt die Signaturdetails auf Empfängerseite.

4.1.2 S/MIME Verschlüsselung

Anders als beim Signieren von E-Mails, treten beim Verschlüsselung von E-Mail Probleme auf. Mit unseren Testzertifikaten konnte das E-Mail zwar verschlüsselt und abgeschickt werden, auf der Empfängerseite aber nicht mehr entschlüsselt. Als Gegenprobe wurde das E-Mail mit dem CMS-S/MIME Tool [1] analysiert und konnte dort erfolgreich entschlüsselt werden. Das legt die Vermutung nahe, dass die ECC-S/MIME Integration prinzipiell richtig arbeitet. Die genaue Fehlerursache konnte jedoch nicht ermittelt werden.

[1] Wobei nur Kurven über Primzahlenkörpern unterstützt werden

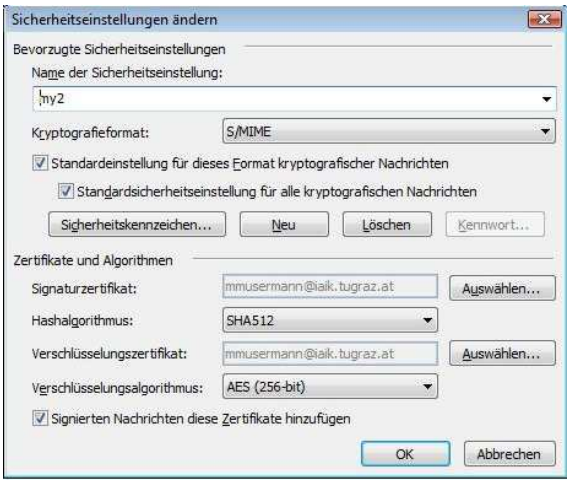

Abbildung 6: S/MIME Konfiguration

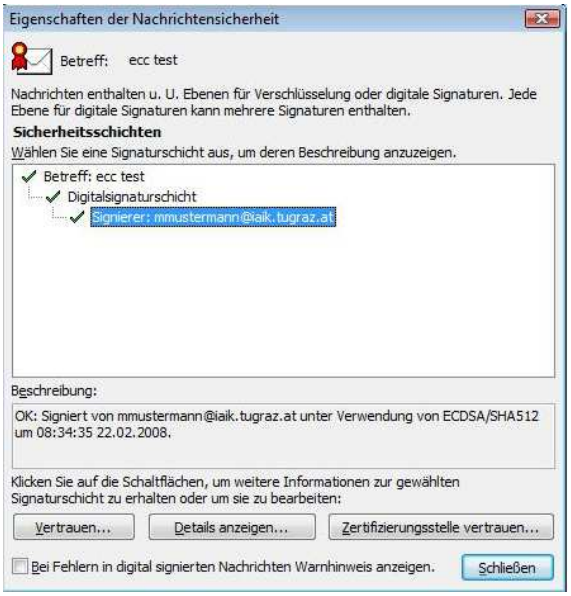

Abbildung 7: S/MIME Signatur

4.2 Mozilla Thunderbird

Dieser Mailclient (getestet wurde die Version 2.0.0.22) greift, gleich wie der Firefox Browser, auf die NSS Bibliothek für die kryptographischen Verfahren zurück. Das erklärt auch, dass ECC Zertifikate problemlos in das Mailprogramm importiert werden können. Die S/MIME Integration ist jedoch noch nicht abgeschlossen und es ist nicht möglich ein signiertes oder verschlüsseltes E-Mail zu senden. Beim Empfang eines ECC verschlüsselten E-Mails kommt eine entsprechende Fehlermeldung. ECC Signaturen werden von Thunderbird offenbar ignorieret und das signierte E-Mail gleich wie ein unsigniertes dargestellt.

Zusammenfassend kann man sagen, dass Thunderbird und Outlook S/MIME und ECC beherrschen. Bei der Kombination dieser beiden Technologien können jedoch, aufgrund der oben geschilderten Schwierigkeiten, Interoperabilitätsproblemen nicht ausgeschlossen werden.

5 Die Österreichische Bürgerkarte

Das Konzept der österreichischen Bürgerkarte ist unabhängig von der konkret eingesetzten Signaturerstellungseinheit definiert (siehe dazu auch [3]). Die derzeit am meisten verbreiten Ausprägungen sind die e-card und die Bankomatkarten. Beide Varianten setzen auf ECC für die Erstellung digitaler Signaturen. Bei der Wahl der konkreten elliptischen Kurve, viel damals die Entscheidung auf die vom National Institute of Standards and Technology (NIST) standardisierte Kurve P-192 [4]. Diese ist leider nicht in der Suite B, enthalten.

Wie die letzten Abschnitte gezeigt haben, konzentrieren sich derzeitige Implementierungen hauptsächlich auf Suite B Kurven und daher ist die P-192 derzeit nicht in Betriebssysteme und Anwendungen integriert. Die weitere Entwicklung dieser ECC Integration ist aus heutiger Sicht nicht abschätzbar.

Auf der anderen Seite ist jedoch davon auszugehen, dass bei zukünftigen Kartengenerationen längere kryptographische Schlüssel (256 Bit und darüber) eingesetzt werden. Dies geschieht nicht zuletzt auch aus der Notwendigkeit heraus, die Parameter und Algorithmen für qualifizierte Signaturen laufend anzupassen. Das bedeutet, dass in absehbarer Zeit eine Migration zu Suite B konformen Kurven notwendig und möglich ist.

6 Zusammenfassung

Für die Verbreitung und den Einsatz von ECC in zukünftigen Anwendungen sind eine entsprechende Standardisierung und die Verfügbarkeit in Softwarebibliotheken notwendige Voraussetzungen. Die Standardisierung ist in vielen Bereichen schon abgeschlossen und die meisten kryptographischen Standardbibliotheken bieten Unterstützung für ECC an. Dies sind beispielsweise:

für C/C++: OpenSSL (`http://www.openssl.org/`) oder Crypto++(`http://www.cryptopp.com/`)

für Java: steht im JCA/JCE Framework eine standardisierte Schnittstelle für ECC Kryptographie zur Verfügung. ECC Implementierungen existieren von den gängigen Java Security Providern.

Damit können Programmierer, unabhängig von der Unterstützung im jeweiligen Betriebssystem, ECC in Anwendungen unterstützen. Dies dürfte vor allem für Produkte mit Smartcardintegration in naher Zukunft an Bedeutung gewinnen.

Wie auch die vorangegangenen Abschnitte gezeigt haben, wird ECC heute zunehmend auch in Standardanwendungen integriert. Teilweise ist dieser Prozess, zumindest bei Suite B Algorithmen, schon sehr weit fortgeschritten. Vor diesem Hintergrund scheint die Entscheidung, bei österreichischen Bürgerkarten (in den Ausprägungen als e-card und Bankomatkarte) auf ECC zu setzen, ein Schritt in die richtige Richtung zu sein. Es ist durchaus zu erwarten, dass die Verbreitung von ECC und die Interoperabilität in den nächsten Jahren zunehmen wird. Derzeit beschränkt sich die Unterstützung jedoch ausschließlich auf elliptische Kurven über Primzahlenkörper mit mindestens 256 Bit (z.B. Kurven aus der Suite B).

Literatur

[1] Iaik-cms with s/mimev3. http://jce.iaik.tugraz.at/sic/products/communication_messaging_security/cms_s_mime, 2007.

[2] S. Blake-Wilson, N. Bolyard, V. Gupta, C. Hawk, and B. Möller. Elliptic curve cryptography (ecc) cipher suites for transport layer security (tls) (rfc 4492), May 2006.

[3] A. Hollosi and G. Karlinger. Einführung in die österreichische bürgerkarte. http://www.buergerkarte.at/konzept/securitylayer/spezifikation/aktuell/introduction/Introduction.html, 2004.

[4] National Institute of Standards and Technology. *FIPS PUB 186-2: Digital Signature Standard (DSS)*. National Institute for Standards and Technology, Gaithersburg, MD, USA, Jan. 2000.

[5] NIST. Recommendation for key management — part 1: General. http://csrc.nist.gov/groups/ST/toolkit/documents/SP800-57Part1_3-8-07.pdf, May 2006.

[6] NSA. Fact sheet nsa suite b cryptography. http://www.nsa.gov/ia/industry/crypto_suite_b.cfm, 2005.

[7] E. Oswald. Einsatz und bedeutung elliptischer kurven für die elektronische signatur. http://www.ecc-brainpool.org/ElliptischeKurven_und_Signatur_Studie.pdf, 2001.

[8] J. Wolkerstorfer and W. Bauer. A pci-card for accelerating elliptic curve cryptography, 2002.

A ECC Cipher Suites

Die folgende Tabelle 2 gibt einen Überblick über die ECC Cipher Suites, wie sie in [2] definiert sind und deren Unterstützung in den Browsern Firefox und Internet Explorer.

Cipher Suite	IE [1] [2]	FF [3]
TLS_ECDH_ECDSA_WITH_NULL_SHA	-	✓[4]
TLS_ECDH_ECDSA_WITH_RC4_128_SHA	-	✓
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA	-	✓
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA	-	✓
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA	-	✓
TLS_ECDHE_ECDSA_WITH_NULL_SHA	-	✓[4]
TLS_ECDHE_ECDSA_WITH_RC4_128_SHA	-	✓
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	-	✓
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	✓	✓
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	✓	✓
TLS_ECDH_RSA_WITH_NULL_SHA	-	✓[4]
TLS_ECDH_RSA_WITH_RC4_128_SHA	-	✓
TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA	-	✓
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA	-	✓
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA	-	✓
TLS_ECDHE_RSA_WITH_NULL_SHA	-	✓[4]
TLS_ECDHE_RSA_WITH_RC4_128_SHA	-	✓
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	-	✓
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	✓	✓
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	✓	✓
TLS_ECDH_anon_NULL_WITH_SHA	-	-
TLS_ECDH_anon_WITH_RC4_128_SHA	-	-
TLS_ECDH_anon_WITH_3DES_EDE_CBC_SHA	-	-
TLS_ECDH_anon_WITH_AES_128_CBC_SHA	-	-
TLS_ECDH_anon_WITH_AES_256_CBC_SHA	-	-

Tabelle 2: Unterstützte Cipher Suites

[1] Microsoft Internet Explorer 7 mit Windows Vista bzw. Windows 7
[2] Gilt auch für Safari und Google Chrome
[3] Mozilla Firefox 3.5
[4] Nicht aktiv in der Standardkonfiguration

B ECC TLS Tests

Um die Unterstützung unterschiedlicher elliptischer Kurven bei den untersuchten Browsern (Internet Explorer und Firefox) zu untersuchen, wurden einige Konfigurationen getestet. Dabei kam als Webserver openssl zum Einsatz. Die folgende Tabelle fasst die Ergebnisse zusammen.

Kurvenname	Firefox	Internet Explorer	Suite B	Kommentar
P-192	-	-	-	
P-256	✓	✓	✓	
P-384	✓	✓	✓	
P-521	✓	✓	-	
B-283	-	-	-	Kurve über $GF(2^n)$

Tabelle 3: Unterstützte Kurven für TLS