



Authentifizierung im elektronischen Geschäftsverkehr

Methoden der Authentifizierung

Historisch gesehen erfolgte eine Authentifizierung von Personen durch Vorlage authentischer Dokumente. Dokumente hingegen werden durch eine eigenhändige Unterschrift oder durch eine spezielle Herstellungsweise mit Stempeln, Wasserzeichen oder ähnlichem authentifiziert.

In der modernen Welt erfolgt eine Authentifizierung meist über Wissen. Dieses Wissen ist vielfach an Besitz wie beispielsweise Chipkarten, WibuKey oder Code-Meter gekoppelt.

Für eine Authentifizierung über Wissen gibt es nun unterschiedliche Ausprägungen. Es kann sich einerseits um ein gemeinsames Geheimnis handeln, das beiden Seiten, dem Verifizierer (User B) und dem Authentifizierten (Beweiser, User A), gleichermaßen und symmetrisch zur Verfügung steht. Oder andererseits um den asymmetrischen Fall, bei dem der Verifizierer die Echtheit des Geheimnisses des Authentifizierten anhand mathematischer Kriterien überprüft.

Sowohl im symmetrischen als auch im asymmetrischen Fall besteht die Möglichkeit bei der Authentifizierung das Geheimnis offen zu legen oder nur das Wissen um das Geheimnis zu beweisen.

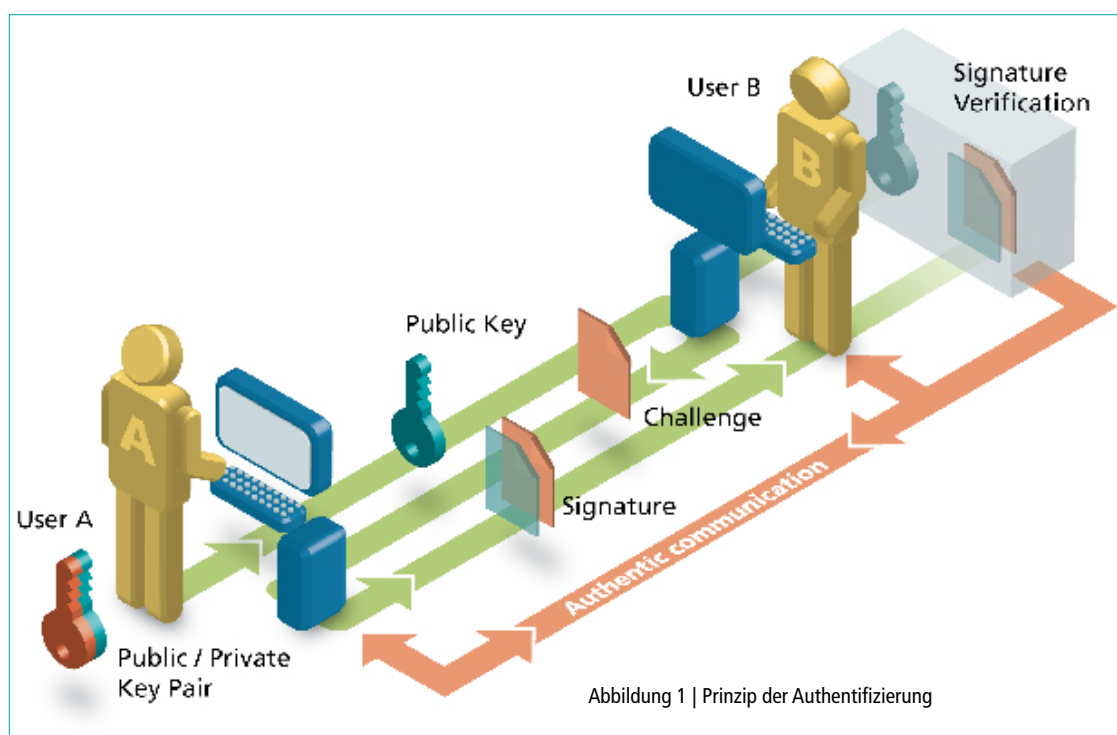
Obwohl heute vielfach Verfahren eingesetzt werden, bei denen das Geheimnis offen gelegt wird, sind solche Verfahren vom Sicherheitsstandpunkt aus sehr bedenklich, da am Übertragungskanal das gesamte Geheimnis mitgelesen werden kann und auch im asymmetrischen Fall (z.B. Kreditkarte) der Verifizierer sich später als der Authentifizierte ausgeben kann.

Autoren:
Rüdiger Kügler
Wolfgang Völker
Dr. Peer Wichmann

Prinzip der Authentifizierung

Authentizität stellt eines der ältesten Probleme des öffentlichen Lebens dar. Es unterteilt sich dabei in unterschiedliche Aspekte der Authentizität:

- Authentizität von Personen
- Authentizität von Gegenständen wie z.B. Handelsware oder Geld
- Authentizität von Informationen oder Dokumenten



Klassischerweise geht eine Authentifizierung mit einer Identifizierung Hand in Hand, da es üblicherweise wenig sinnvoll ist von Authentizität zu reden, ohne eine Identifikation zu haben.

Im elektronischen Geschäftsverkehr kann sich jedoch Authentizität auch auf ein Pseudonym beziehen. Dies ist vielfach aus Sicht des Datenschutzes wünschenswert und erfüllt in den meisten Fällen die gestellten Anforderungen, da durch einen weitergehenden Prozess das Pseudonym auf eine reale Identität abgebildet werden kann.

Sinnvoll und zeitgemäß ist eine Authentifizierung mittels elektronischen Signaturen. Dabei wird ein Dokument mit einer elektronischen Signatur unterzeichnet. Der Verifizierer überprüft über die Gültigkeit der Unterschrift die Authentizität des Beweisers.

Im Fall von CodeMeter werden hier mehrere positive Aspekte der Authentifizierung miteinander verbunden:

- Sicherheit durch die Verwendung einer standardisierten ECDSA-Signatur.
- Sichere Aufbewahrung des kryptographischen Schlüssels im CmStick.
- Der Benutzer ist nicht direkt identifizierbar, da über die Seriennummer des CmStick eine Pseudonymisierung erfolgt. Dies ist bei vielen heutigen Anwendungen wichtig, da hierüber z.B. auch eine weitgehend anonyme Authentifizierung im Rahmen einer Altersverifikation möglich ist.
- Einfach umzusetzende API.
- Authentifizierung ohne gemeinsames Geheimnis und ohne Offenlegung eines Geheimnisses.

Der Ablauf einer Authentifizierung durch eine Signatur im CmStick und die Verifikation sind in folgende Teilschritte gegliedert:

1. Übertragen des öffentlichen Verifikationsschlüssels an den Verifizierer (einmalig und nicht vertraulich)
2. Erstellen einer Anfragenachricht (Challenge) durch den Verifizierer und senden zum Beweiser
3. Erstellen des Hashwertes und der Signatur und übertragen zum Verifizierer
4. Überprüfen der Signatur

Die CodeMeter Technologie

■ Die Hardware

Das Herzstück von CodeMeter ist der CmStick, eine USB Hardware. Er bietet die Möglichkeit, Daten zu versenden und entschlüsseln. Die dazu nötigen Schlüssel werden im CmStick generiert und sind nicht von außen auslesbar. Im CmStick können unterschiedliche Schlüssel für viele verschiedene Lizenzen in einer Baumstruktur gespeichert werden. Auf der ersten Ebene findet man das Firm Item (FI). Jeder Licensor besitzt ein eigenes Firm Item. Nur er ist in der Lage, mit seiner FSB in seinem Firm Item Einträge anzulegen, zu modifizieren oder zu löschen.

In der zweiten Ebene befinden sich die Product Items (PI). Diese kennzeichnen ein Produkt bzw. eine Lizenz.

Unter den Product Items liegen die Product Item Options. Hier werden die Lizenzinformationen wie z.B. Pay-Per-Use-Counter gespeichert.

Der CmStick bietet 128 Bit AES (Advanced Encryption Standard) für symmetrische und 224 Bit ECC (Elliptic Curves Cryptography) für asymmetrische Verschlüsselung.

Er ist gegen hardware-spezifische Angriffe, wie z.B. durch das Auslesen mit Elektronenmikroskopen, geschützt.

Über 1.000 Lizenzen von mehr als 100 verschiedenen Herstellern finden im CmStick Platz. Dabei kann jede einzelne Lizenz mit verschiedenen Optionen, wie z.B. Begrenzungszähler oder Ablaufdatum, versehen werden.

■ Infrastruktur

CodeMeter bietet mit der Infrastruktur CmTalk die Möglichkeit, Lizenzen in den CmStick beim Anwender zu übertragen.

Dazu geht der Anwender (U) in den Online-Shop des Händlers (T) und bestellt eine Lizenz. Diese Anfrage wird vom Händler automatisch an den Softwarehersteller (L) weitergeleitet.

Als nächstes bezahlt der Anwender die Lizenz bei der Zahlstelle (C).

Sobald die Zahlstelle die Zahlung bestätigt hat, erfolgt die Erzeugung der Lizenz durch den Softwarehersteller und die Übertragung zum Anwender.

Dabei identifizieren sich alle Beteiligten durch eigene CmSticks. Der Licensor benötigt eine Firm Security Box (FSB), um die Lizenzen zu erzeugen.



Abbildung 2 | CmTalk

Wie in Abbildung 2 dargestellt, bietet die Flexibilität von CodeMeter einem Softwarehersteller auch die Möglichkeit, alle drei Rollen selbst auszuführen.

Asymmetrische Implementierung mit CodeMeter

■ Initialer Schlüsselaustausch

Für die Signatur wird ein Product Item in einem CmStick benötigt. Über dieses Product Item und den darin enthaltenen Secret Data (Private Key) wird dann eine ECDSA-Signatur berechnet.

Der zu diesem Private Key passende Public Key kann mit diesem CmStick berechnet werden.

Der Public Key des Beweisers muss nun einmalig zum Verifizierer geschickt werden. Dies kann über verschiedene Wege erfolgen:

- Programmierung des Product Items vor der Auslieferung und Ablage des Public Keys in der Datenbank des Verifizierers.
- Programmierung des Product Items über CmTalk.

■ Clientseite

Auf einer Authentifizierungsseite wird je nach Rechner-Plattform und Browser über ein Java-Applet oder ActiveX-Control eine Authentifizierung über den CmStick durchgeführt. Hierbei fordert der Beweiser beim Verifizierer (Server) eine Challenge an, die vom Beweiser über das Product Item im CmStick signiert wird.

Nach erfolgreicher Authentifizierung wird die Sitzung am Server als authentifiziert gekennzeichnet und der Client erhält Zugriff auf die geschützten Seiten. Die Authentifizierung kann jederzeit, z.B. nach Ablauf einer bestimmten Zeitspanne, auf Anforderung des Servers erneut durchgeführt werden.

Auf der Clientseite muss das CodeMeter Runtime Kit installiert sein.

■ Serverseite

Serverseitig müssen neben der vorhandenen Sitzungsverwaltung zum einen die oben erwähnte Challenge berechnet und zum anderen die vom Beweiser berechnete Signatur überprüft werden.

Die Challenge muss bis zur Prüfung der Signatur vom Server auf geeignete Weise aufbewahrt werden (Sitzungsvariable) oder aber mit einem kryptographischen Prüfkriterium versehen sein, so dass später eine Überprüfung auf Echtheit durchgeführt werden kann.

Nach Erhalt der Signatur vom Beweiser wird serverseitig geprüft, ob die verwendete Challenge und die Signatur gültig sind. Dabei wird zur Prüfung der Signatur der auf dem Server hinterlegte Public Key verwendet, der zur Seriennummer passen muss.

Im Fall, dass die Signatur gültig ist, gilt die Session als authentisch. Ab hier wird das bisherige Sessionhandling eingesetzt. Zur Überprüfung der Signatur wird kein CmStick am Server benötigt. Hier muss lediglich das CodeMeter SDK auf dem Server installiert sein und die Funktion CmValidateSignature aufgerufen werden.

Das SDK ist im Augenblick für Windows, Macintosh und Linux verfügbar. Spezielle andere Plattformen sind auf Anfrage verfügbar.

■ Beispiele

Für die asymmetrische Authentifizierung mit CodeMeter sind Beispielimplementierungen für verschiedene Technologien vorhanden, wie z.B. PHP oder JSP mit Serverkomponenten in Java oder C. Diese können beim technischen Support unter support@wibu.de angefordert werden.

Symmetrische Implementierung mit CodeMeter

Alternativ zur oben beschriebenen Implementierung mit asymmetrischen Algorithmen kann eine Authentifizierung ebenfalls mit symmetrischen Algorithmen erfolgen. Der Hauptunterschied ist, dass dann beide Seiten über das gleiche Geheimnis verfügen. Daher erfüllt die symmetrische Variante nicht die gleichen hohen Sicherheitsanforderungen wie die asymmetrische Variante.

Die übliche Variante zur symmetrischen Authentifizierung ist die Authentifizierung über Passworte. Diese hat jedoch den Nachteil, dass das Geheimnis bei der Authentifizierung im Rahmen des Übertragungskanal öffentlich bekannt wird und somit nur noch begrenzt für weitere Authentifizierungen eingesetzt werden kann.

Wesentlich sinnvoller ist die Verwendung eines Verfahrens, welches das Geheimnis nicht offen legt, sondern nur dessen Besitz zweifelsfrei nachweist. Der Ablauf ähnelt dem oben geschilderten Ablauf. Im Gegensatz zu einer elektronischen Signatur wird hier jedoch die symmetrische Verschlüsselung mit AES eingesetzt.

■ Initialer Schlüsselaustausch

Der auf beiden Seiten benötigte Schlüssel kann entweder vor Auslieferung der Hardware an den Kunden programmiert werden oder über CmTalk programmiert werden. Der Schlüssel kann als Secret Data im Product Item abgelegt werden und ist damit nicht von außen auslesbar. Die Schlüssel müssen serverseitig vertraulich archiviert werden, um eine Kompromittierung und damit den Wegfall der Authentifizierungsbasis zu verhindern.

■ Clientseite

In die Login-HTML-Seite wird wie bei der asymmetrischen Implementierung ein Java-Applet oder ActiveX-Control integriert (CmAuthMod). Dabei übernimmt das CmAuthMod die Verbindung zum CmStick. Der Ablauf ist für den Benutzer identisch mit dem asymmetrischen Protokoll.

■ Serverseite

Auch die Serverseite ist vom Ablauf her identisch mit der asymmetrischen Variante. Die verwendeten Schlüssel müssen allerdings vertraulich aufbewahrt werden. Im asymmetrischen Fall müssen die Schlüssel nur authentisch sein.

Symmetrische Implementierung mit WibuKey

Eine Implementierung auf Basis von WibuKey ist prinzipiell vom Ablauf und den Modulen her identisch zu einer symmetrischen CodeMeter-Implementierung.

Es treten jedoch einige Einschränkungen auf, da hier die kryptographische Schlüssel nicht so frei vergeben werden können wie mit CodeMeter. Authentifiziert werden kann hier nur die Existenz einzelner User Codes. Damit ergibt sich eine Art Gruppenauthentifizierung, durch die die Zugehörigkeit zu einer autorisierten Gruppe bestätigt wird, ohne dabei den einzelnen Benutzer zu authentifizieren. Die durch diesen Mechanismus erreichte Sicherheit ist nochmal geringer, als mit der symmetrischen CodeMeter-Variante.

Sicherheitsoptionen

Der CmStick verfügt über ein Passwort und drei Zustände einer Enabling-Funktion:

- Deaktiviert
- Aktiviert bis zum Abziehen
- Voll aktiviert

Wir empfehlen die Einstellung „Aktiviert bis zum Abziehen“, da nur dann das Passwort bei jedem Anstecken an den PC erneut eingegeben werden muss. Ohne diese Passwortheingabe ist eine Erzeugung der Signatur nicht möglich. Die zur Verschlüsselung verwendeten Secret-Data-Einträge können direkt im CmStick erzeugt werden und verlassen den CmStick nie.

Beim Einsatz symmetrischer Verfahren wird bei CodeMeter AES und bei WibuKey Feal-32 verwendet. Bei CodeMeter existieren die gleichen Möglichkeiten im Rahmen des Enabling wie bei asymmetrischen Protokollen.

Bei WibuKey bezieht sich die Authentifizierung lediglich auf den Besitz des WibuKey. Ein zusätzliches Wissen im Rahmen eines Enabling ist nicht möglich.

Vom Standpunkt der Sicherheit aus gesehen überwiegen die Vorteile einer asymmetrischen Implementierung mit CodeMeter.

Vorteile

Eine Authentifizierung mit Hardwaremechanismen, wie CodeMeter oder WibuKey, bringt den Vorteil, dass eine weitere Komponente hinzukommt. Zur klassischen Authentifizierung über Wissen (Passwort) kommt hier die Authentifizierung über Besitz hinzu. Beide können zur Erhöhung der Sicherheit kombiniert werden.

Im Fall von CodeMeter ergibt sich über das Enabling noch mal eine engere Verknüpfung zwischen beiden Varianten. CodeMeter zusammen mit der asymmetrischen Authentifizierung zeigt sich also auch hier als eine ideale Lösung:

- Keine geheimen Schlüssel beim Verifizierer
- Höhere Performance, da keine Hardware am Server notwendig ist
- Signaturen sind auch für andere Zwecke einsetzbar z.B. für Transaktionen
- Einfache Implementierung auf verschiedenen Plattformen am Server
- Einfache Integration auf der Clientseite als Applet oder ActiveX
- Höchste Sicherheit
- Sichere Aufbewahrung des Private Keys im CmStick
- Optimale Pseudonymisierung z.B. für Altersverifikation



WIBU-SYSTEMS ist nach
DIN EN ISO 9001:2000
zertifiziert
und Mitglied u.a. bei
BITKOM e.V., PCMCIA,
USB Implementers Forum
sowie SIIA.

**„Wir wollen das
Vertrauen unserer
Kunden jeden Tag aufs
Neue verdienen“**

WIBU-SYSTEMS wurde 1989 von Oliver Winzenried und Marcellus Buchheit gegründet und hat sich auf Digital-Rights-Management, d.h. Schutz von Software, Dokumenten, Zugang und Medien, spezialisiert. Sowohl im Hauptsitz in Karlsruhe als auch in den Niederlassungen in Seattle, USA, und in Shanghai, China, wird die Philosophie von WIBU-SYSTEMS gelebt.

Kontinuierlich erweitert WIBU-SYSTEMS das Portfolio mit wichtigen Sicherheitslösungen. Mit CodeMeter steht eine patentierte Digital-Rights-Management-Lösung zur Verfügung, um Software und andere digitale Inhalte, auch gleichzeitig für viele Produkte verschiedener Anbieter, sicher zu schützen - mit Vorteilen für Anbieter und Anwender. WibuKey ist seit vielen Jahren auf dem internationalen Markt etabliert. Neben dem Schutz von Software bietet WibuKey Lizenzmanagement, modularen Software-Vertrieb und Electronic Software Distribution (ESD) auf einem hohen Sicherheitslevel.

Weitere Produkte sind SmartShelter zum Schutz von HTML-, PDF- und Media-Dokumenten sowie SecuriKey als Zugangsschutz.

Weltweit wird WIBU-SYSTEMS durch kompetente Distributoren unterstützt.

Beim SIIA Codie Awards 2005 und 2006 wurde WIBU-SYSTEMS mit CodeMeter als Finalist in den Kategorien Best Digital Rights Management Solution: Software und Best Security Software ausgezeichnet, der CmStick wurde beim internationalen Designwettbewerb iF Product Design Award auf der CeBIT ausgezeichnet und zum Designpreis der Bundesrepublik Deutschland 2006 nominiert.



**WIBU
SYSTEMS**

WIBU-SYSTEMS AG · Rüppurrer Straße 52-54 · D-76137 Karlsruhe
Tel: +49-721-93172-0 · Fax: +49-721-93172-22
info@wibu.de · www.wibu.de