

Einmalpasswörter (TANs) - Grundlagen, Anwendungen, Generierung

Jakob Lehner, Andreas Riedler

Communication Engineering, Institut für Wirtschaftsinformatik
Freistädterstrasse 315, 4040 Linz, Österreich
<http://www.ce.uni-linz.ac.at>

Zusammenfassung In dieser Seminararbeit werden die Grundlagen und Anwendungen von TAN-Verfahren erläutert und in einer Fallstudie gegenübergestellt und verglichen.

Key words: TAN, Einmalpasswörter, Two-Factor-Authentication

Inhaltsverzeichnis

1 Einleitung

Business Services; Was sind Business Services? Warum ist Business Services wichtig? Um die Sicherheit von Business Services speziell im Kontext des eCommerce zu erhöhen, setzen Banken, Regierungen und andere sicherheitssensitive Industrien Einmalpasswörter ein. Bei einer Authentifikation mit Einmalpasswörtern, besitzen die Benutzer mehrere Passwörter, welche jedoch nur einmal verwendet werden. Welche Vorteile bringt das?

Ziel dieser Arbeit ist es Grundlagen, Konzepte zur Generierung und Anwendungen von Einmalpasswörtern zu erläutern. Zu Beginn werden die wesentlichen Grundlagen und anschließend die gängigsten Verfahren zur Generierung und Verteilung von Einmalpasswörtern vor- und gegenübergestellt. Zum Abschluss der Arbeit wird anhand eines konkreten Anwendungsgebietes der Einsatz von Einmalpasswörtern aufgezeigt.

Bei den Verfahren noch rückbeziehen auf Grundlagen?

2 Grundlagen

In diesem Teil der Arbeit sollen die grundlegenden Konzepte zur Authentifizierung mit Einmalpasswörtern erläutert werden. Dazu werden zu Beginn die Arten der Authentifizierung vorgestellt und wie Einmalpasswörter einzuordnen sind. Im Anschluss werden die verschiedenen Konzepte zur Generierung von Einmalpasswörtern vorgestellt.

2.1 Arten der Authentifizierung

Die Unterscheidung von Authentifizierungsmechanismen erfolgt in der Regel in die drei Arten Wissen, Besitz und körperliche Merkmale. Bei einer Authentifizierung mittels Wissen hat der Benutzer sicherzustellen, dass er über ein bestimmtes Wissen verfügt. Dies ist in der Regel ein Passwort, eine Personal Identification Number (PIN) oder ein kryptographisches Geheimnis. Von einer Authentifizierung durch Besitz spricht man, wenn der Benutzer den Besitz einer bestimmten Sache belegen kann. Dabei muss es sich nicht nur um physische Sache wie Chipkarten oder Tokens sondern auch Software, welche ein gültiges Passwort für die Authentifizierung erzeugt, ist eine solche Sache. Erfolgt die Authentifizierung über körperliche Merkmale muss sich der Benutzer über biometrische Mechanismen wie Fingerabdruckscan oder Stimmanalyse zu erkennen geben.[1] Jede dieser Arten zur Authentifizierung bietet unterschiedliche Vor- und Nachteile hinsichtlich Komplexität, Sicherheit und Benutzerfreundlichkeit. Auf diese Aspekte wird in dieser Arbeit jedoch nur hinsichtlich Einmalpasswörtern und in einem der folgenden Kapitel eingegangen.

Wird nur eine der drei genannten Arten zur Authentifizierung verwendet, so spricht man von einer Single-Factor-Authentication. Dies ist der Fall wenn die Authentifizierung nur durch die Eingabe eines Passwortes erfolgt. Single-Factor-Authentication ist die schwächste Form der Authentifizierung da ein Passwort ausspioniert, ein Token gestohlen oder ein Fingerabdruck nachgemacht werden kann. Werden hingegen zwei Arten der Authentifizierung miteinander kombiniert, nennt man dies eine Two-Factor-Authentication. Durch Hinzufügen eines zweiten Identifiers wird die Sicherheit des Authentifizierungsprozesses erhöht. Zum Beispiel kann eine Authentifizierung mittels Passwort und Token erfolgen. Somit muss der Benutzer zusätzlich zu dem Wissen (Passwort) eine Sache (Token) besitzen. Von einer Three-Factor-Authentication spricht man wenn alle drei Arten zusammen eingesetzt werden. Typischerweise ist, zusätzlich zu Passwort und Token, die biometrische Identifikation der dritte Faktor bei einer Three-Factor-Authentication. Dies ist die stärkste Form der Authentifizierung, da der Benutzer zusätzlich zu Wissen (Passwort) und Sache (Token) auch die biometrischen Merkmale haben muss um eine Authentifizierung durchführen zu können.[2]

Einmalpasswörter werden in der Regel zur Two-Factor-Authentication verwendet. Der initiale Login erfolgt mit Kennwort und Passwort, welche den ersten Faktor darstellen. Als zweiter Faktor dient das Einmalpasswort, das in einem zweiten Schritt eingegeben wird. Typische Anwendungsbereiche für Einmalpasswörter sind Remote-Zugänge zu Systemressourcen, Zugänge zu internen Netzwerken von Organisationen und Anwendungen im e-Business wie zum Beispiel Online-Banking.[3] Auf die Anwendungen von Einmalpasswörtern wird im Kapitel Anwendungen noch näher eingegangen.

2.2 Konzepte zur Generierung von Einmalpasswörtern

Nachdem wir nun die verschiedenen Arten der Authentifizierung erläutert und Authentifizierung mittels Einmalpasswörtern eingeordnet haben, werden wir in diesem Teil die wesentlichen Konzepte zur Generierung von Einmalpasswörtern vorstellen. Zuerst werden wir eine Einführung in Einweg-Hash-Funktionen geben, da diese für die Berechnung von Einmalpasswörtern herangezogen werden. Anschließend leiten wir zu den Methoden zur Generierung von Einmalpasswörtern über.

Einweg-Hashfunktionen Sind Hashfunktionen, die nur in eine Richtung funktionieren. Dies bedeutet, dass sich zu einer Eingabe ein Hashwert problemlos berechnen lässt. Jedoch ist es schwer ausgehend von einem bestimmten Hashwert, den Originalwert zu berechnen. Schwer bedeutet dabei, dass es vom Rechenaufwand her ist, aus einem Hashwert das Original zu berechnen.[4]

Hashfunktionen legen einen endlichen Bildbereich, der auch als Adressbereich bezeichnet wird, fest. Dieser Adressbereich ist meist erheblich kleiner als der Urbildbereich, der auch das Universum genannt wird. Hashfunktionen bilden jedes Objekt eines Universums auf eine Hashadresse ab. Da der Bildbereich meist erheblich kleiner ist als das abzubildende Universum, können Kollisionen auftreten.[5]

Die Sicherheit einer Einweg-Hashfunktion liegt in ihrer Einweg-Eigenschaft, da die Hashfunktion öffentlich und somit das Verfahren nicht geheim ist. Die Ausgabe steht in keinem nachvollziehbaren Zusammenhang mit der Eingabe und durch die Änderung eines Bits in der Eingabe ändert sich im Mittel die Hälfte aller Bits im Hashwert. Eine gute Einweg-Hashfunktion ist außerdem kollisionsresistent. Dies bedeutet, dass es erstens praktisch unmöglich ist, zwei Originalwerte mit demselben Hashwert zu erzeugen und zweitens soll es nicht möglich sein, zu einem gegebenen Hashwert eine Nachricht zu finden, die den selben Hashwert liefert.[4]

The S/Key One-Time Password System Das S/Key One-Time Password System basiert auf den Ideen von Leslie Lamport. Leslie Lamport adressiert mit seinem Verfahren zwei wesentliche Sicherheitsrisiken der Authentifizierung mit Passwörtern. Zum ersten nennt er das Risiko, dass ein Angreifer Zugang zur Passwortdatei erlangt. Das zweite Sicherheitsrisiko, auf das sich Lamport bezieht, ist die Gefahr, die beim Austausch und Überprüfen des Passworts entsteht. Ein Angreifer könnte dabei an das Passwort gelangen.[6]

Das erste Problem löst Lamport durch die verschlüsselte Ablage des Passwortes auf dem Zielrechner. Die Verschlüsselung erfolgt mittels Einwegfunktion. Durch den Einsatz einer Sequenz von Passwörtern löst Lamport das zweite Problem. Diese Sequenz von Passwörtern kann zum einen Initial gewählt werden oder der Benutzer sendet beim aktuellen Login das verschlüsselte Passwort für den nächsten Login. Beide Varianten weisen jedoch Mängel auf. Bei der ersten müssten sowohl Benutzer als auch das System eine umfangreiche Passwortliste speichern und verwalten. Die Schwäche der zweiten Variante liegt in der Kommunikation zwischen Benutzer und System. Durch Kommunikationsfehler oder durch Angreifer ist es möglich, dass das System den verschlüsselten Wert des Passwortes für die nächste Anmeldung nicht oder falsch erhält.[6]

Lamport löst die angeführten Probleme durch eine wiederholte Ausführung einer Einwegfunktion F auf ein Passwort x . Dies bedeutet, dass das Ergebnis von $F(x)$ als Input für das nächste Passwort verwendet wird. Eine Sequenz von Passwörtern für den Benutzer ist somit

$$F^{n-1}(x), \dots, F(F(F(F(x))), F(F(x))), F(x), x$$

und die Passwortsequenz y auf dem System lautet

$$F^n(x), \dots, F(F(F(F(x))), F(F(x))), F(x).$$

Dies bedeutet, dass jedes Benutzerpasswort, jenes Passwort ist, mit dem das System das nächste Passwort bestimmen kann. Das System muss mit dem Wert $y_1 = F^n(x)$ initialisiert werden. Nun muss es sich nur mehr das letzte Passwort merken, welches vom Benutzer gesendet wurde.[6]

Das S/Key One-Time Password System (SKOTP) übernimmt die Ideen von Lamport und wurde erstmals auf einem Unix System implementiert. Die Ziele dieses Systems sind der Schutz gegen **Eavesdropping**, einfache Benutzbarkeit,

automatische Ausführung, keine Verwendung von geheimen Algorithmen und keine Speicherung von Geheimnissen. Wie bei Lamport soll das SKOTP Schutz gegen passives **Eavesdropping** bieten, in dem keine Information zwischen System und Benutzer wechselt, die zur Authentifikation verwendet werden kann. Die einfache Ausführbarkeit soll die Benutzerakzeptanz erhöhen und die automatische Ausführung soll eine leichte Konvertierung auf ein tokenbasiertes System ermöglichen. Also ein Wechsel von Authentifizierung durch Wissen auf Authentifizierung durch Besitz einer Sache. Dadurch dass keine geheimen Algorithmen verwendet werden, kann das SKOTP von der Industrie evaluiert werden. Sicherheitslücken können so mit größerer Wahrscheinlichkeit entdeckt. In Bezugnahme auf Lamport erfolgt beim SKOTP ebenfalls keine Speicherung von geheimen Schlüsseln oder Passwörtern auf dem Host. Dies vermindert die Attraktivität des Hosts als Ziel.[7] Als Basis für das Verfahren dient ein geheimes Passwort s , welches zwischen Benutzer und seinem Arbeitsplatzrechner vereinbart wurde. Hervorzuheben ist hierbei noch einmal, dass dem Server das Geheimnis s nicht bekannt sein muss. Also muss weder ein Austausch des Geheimnisses erfolgen noch muss der Server das Geheimnis verwalten. Die Ausführung erfolgt in den zwei Schritten Vorbereitung Authentifizierung.[5] Im Vorbereitungsschritt wird

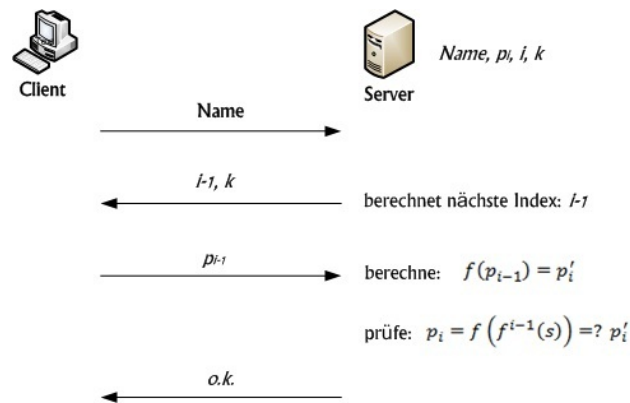


Abbildung 1. Schematische Darstellung des S/Key Schemas[5]

vom Server zum Client ein Seed k in Klartext gesendet. Dieser Seed ist eindeutig für den Benutzer auf dem entsprechenden Server und dient dazu, dass ein Benutzer das gleiche Passwort auf mehreren Rechnern verwenden kann. Aus dem Geheimnis s und dem Seed-Wert k berechnet der Client die Folge der benutzbaren Passwörter p_1, \dots, p_n wie folgt[5]:

$$p_i = f^i(s|k) \text{ für } i = 1, \dots, n$$

Für die Authentifizierung des Clients zum Server werden ausschließlich diese Einmalpasswörter p_i verwendet. So erfolgt nie ein Austausch des Geheimnisses s .

Für den Benutzer ändert sich nichts gegenüber eine herkömmlichen Anmeldung. Er weist sich gegenüber dem Client mit seinem Geheimnis s als authentisches Subjekt aus.[5] Nach der i -ten erfolgreichen Authentifizierung gilt, dass der Server das Passwort p_i überprüft und zusammen mit dem Index i protokolliert hat. Erfolgt eine neuerlicher Login-Anfrage durch den Client-Rechner, wird er vom Server aufgefordert, das $(i - 1)$ -te Passwort vorzuweisen. Für diesen Zweck sendet der Server die laufende Nummer $i - 1$ sowie den Seed-Wert k an den Client-Rechner. Daraus berechnet(oder wählt es aus einer Liste aus) und überträgt der das entsprechende Passwort p_{i-1} zum Server. Das Passwort p_{i-1} erhält man durch wiederholtes($i - 1$ mal) Anwenden der Funktion f auf das Geheimnis s welches mit dem Seed k angereichert wurde. Ob das Passwort p_{i-1} korrekt ist, kann der Server durch die einmalige Anwendung der Funktion f auf p_{i-1} überprüfen. Für das Passwort gilt[5]:

$$p_i = f^i(s|k) = f(f^{(i-1)}(s|k)) = f(p_{i-1})$$

Somit ist zu überprüfen, ob gilt

$$f(p_{i-1}) = p_i$$

Bei einer Übereinstimmung ist die Authentifizierung erfolgreich. Der Server ersetzt nun den Wert p_i durch das verbrauchte Passwort p_{i-1} und erniedrigt seinen Sequenzzähler um eins.[5]

Die Sicherheit dieses Verfahrens ist abhängig von der Einwegeigenschaft der Funktion f und von der Qualität des zugrundeliegenden Geheimnisses s . Die Übertragung des Passwortes p_i zwischen Client und Server erfolgt unverschlüsselt. Dadurch ist es einem Angreifer möglich dieses zu beobachten. Jedoch ist es ihm nicht möglich durch Kenntnis des Passwortes p_i , das nächste Passwort zu bestimmen. Da er das Geheimnis s nicht kennt, müsste er das Urbild des Passwortes $p - i$ wie folgt berechnen[5]:

$$p_{i-1} = f^{-1}(p_i)$$

Somit wird eine erfolgreiche Maskierung erschwert, jedoch nicht vollständig ausgeschlossen. Ist es dem Angreifer möglich das Passwort p_i welches vom Client zum Server gesendet wird, so abzufangen dass es den Server nicht erreicht, bricht dieser die Authentifizierung ab, ohne seinen Passwortzähler zu vermindern. Der Angreifer kann sich nun beim Server anmelden in dem er das Abgefangene Passwort p_i verwendet. Diese Schwachstelle kann jedoch beseitigt werden im dem der Server bei einem Fehlgeschlagenen Login den Passwortzähler trotzdem vermindert.[5] Neben einer clientseitigen Maskierung ist auch eine Serverseitige möglich. Maskiert sich ein Angreifer als Server und übermittelt dem Client eine Sequenznummer j , die wesentlich kleiner ist als die zuletzt Verwendete, so wird der Client dem Server das korrekte Passwort p_i übermitteln. Somit ist es dem Angreifer möglich alle Sequenznummern $r \in j, \dots, i - 1$ des Original-Servers mit einem korrekten Passwort zu beantworten, indem er ausgehend vom erschlichenen Passwort p_j den Wert

$$f^{r-j}(p_j) = p_r$$

berechnet.[5]

Durch das Abfangen von Einmalpasswörtern, der Sequenznummer i und dem Seed k entsteht ein weiteres Sicherheitsproblem. Ein Angreifer kann mit diesen Werten einen Wörterbuchangriff auf das geheime Passwort s durchzuführen. Bei einem Wörterbuchangriff muss der Angreifer mit allen möglichen Wörtern s' eine S/Key-Berechnung durchführen. Das heißt er muss die bekannte Hashfunktion i -mal auf s' , angereichert mit dem Seed k anwenden. Dieses Problem kann dadurch gelöst werden, dass dem Benutzer durch das System ein geheimes Passwort s zugewiesen wird, welches eine gute Zufallszahl darstellt.[5]

Challenge Response Mitchell: Comments on the SKEY user authentication scheme

Zeitbasierend Claudia Eckert: IT Sicherheit OTP-Tokens/RSA SecurID

3 Generierung und Verteilung

Papers in meinem new Ordner!

3.1 TAN - Liste

TAN Liste, iTAN

3.2 SMS

3.3 Mobiltelefon

3.4 Tokens

3.5 Einordnung und Vergleich der Technologie

Welche Art der Generierung wird verwendet? Sicherheit? Vorteile und Nachteile?

4 Anwendung

Spezielle Case Studies, zwei oder mehr Beispiele, die aufgrund einer Skala z.B. Sicherheit, Aufwendigkeit usw. bewertet werden.

alte Quellen:

Gianluigi Me et al: A mobile based approach to strong authentication on Web
Kuan-Chieh Liao: A One-Time Password Scheme with QR-Code Based on Mobile Phone

Fadi Aloul: Two Factor Authentication Using Mobile Phones
Telvis E. Calhoun Jr.: Authentication in 802.11 LANs using a Covert Side Channel
Binod Vaidya: Authentication Mechanism Using One-Time Password for 802.11 Wireless LAN
Chun-Ming Leung: Depress Phishing by CAPTCHA with OTP
Daiki NObayashi: Development of Single Sign-On System with Hardware Token and Key Management Server
Paras Babu Tiwari: Single Sign-on with One Time Password
Wen-Chung Kaol: Integrating Flexible Electrophoretic Display And One-Time Password Generator in Smart Cards
Yongjin Lee: One-Time Templates for Face Authentication
Christian Gilmore: Secure Remote Access to an Internal Web Server
Liang Fang: Secure Password-Based Authenticated Key Exchange for Web Services

neue Quellen:

MohammedAlZomai: An Experimental Investigation of the Usability of Transaction Authorization
Roberto Di Pietro: A Two-Factor Mobile Authentication Scheme
Christos K. Dimitriadis: Analyzing the Security of Internet Banking
Anders Moen Hagalisletto: Analyzing two-factor authentication devices
Cornel de Jong: Online authentication methods

5 Fazit

6 Quellen

[1]Two-factor authentication Siemens [2]Mike Meyers' CompTIA Security+ Certification Passport, Second Edition Von T. J. Samuelle [3]Information security architecture: an integrated approach to security in the ... Von Jan Killmeyer Tudor [4]Angewandte Kryptographie Bruce Schneider [5]IT-Sicherheit Claudia Eckert [6]Password Authentication with Insecure Communication; Leslie Lamport [7]The S/Key One-Time Password System; Neil M. Haller