


Sistemi di elaborazione dell'informazione (Sicurezza su Reti)
Anno Accademico 2003-2004




One Time Password

A cura di: Caccese Antonio
Cutolo Alfredo
De Simone Luigi

Prof : A. De Santis

Problema...



- Una forma di attacco ai sistemi connessi in rete consiste nel mettersi in ascolto sulle connessioni allo scopo di catturare informazioni di autenticazione come loginID e password di utenti legittimi
- Tali informazioni possono essere utilizzate in un secondo momento per accedere al sistema

I sistemi One Time Password sono progettati per neutralizzare questo tipo di attacco

Cos'è OTP ?

Il sistema One Time Password (OTP) è un meccanismo di autenticazione di tipo challenge-response che consente l'accesso ad un sistema (login) e ad altre applicazioni

Le One Time Password sono così denominate perché possono essere utilizzate validamente una sola volta

Il sistema OTP è stato sviluppato presso i laboratori della Bellcore (attualmente nota come Telcordia Technologies) dove è stato implementato con il nome di S/KEY

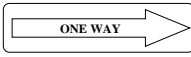
Cosa tratteremo...

- ✓ Funzionamento del sistema OTP
- ✓ Generazione delle One Time Password
- ✓ Considerazioni sulla sicurezza
- ✓ Esempi di verifica e codifica
- ✓ Risposte estese OTP
- ✓ Meccanismo OTP SASL

Funzioni hash in OTP

La sicurezza del sistema OTP è basata sulle funzioni hash

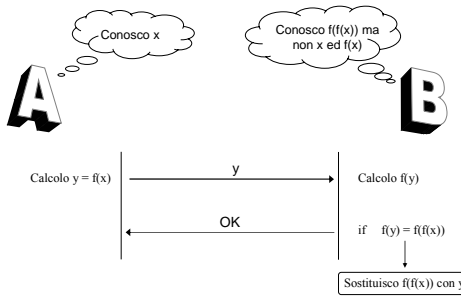
Funzioni che godono della proprietà "One Way": facili da calcolare ma computazionalmente difficili da invertire



Il protocollo OTP usa come funzioni hash gli algoritmi MD4, MD5 e SHA

Funzionamento del sistema OTP

Utilizzo delle funzioni hash nel sistema OTP:



Conosco x

Conosco $f(f(x))$ ma non x ed $f(x)$

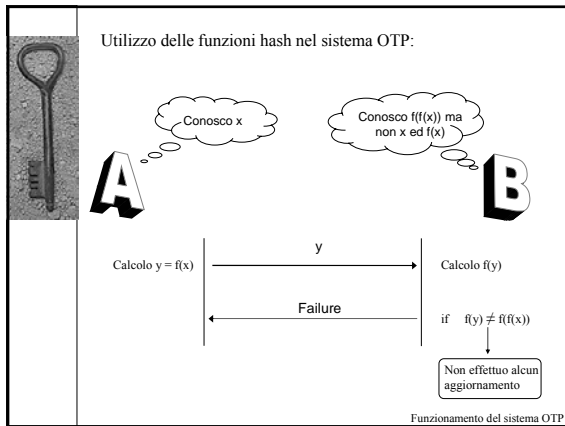
Calcolo $y = f(x)$

Calcolo $f(y)$

if $f(y) = f(f(x))$

Sostituisco $f(f(x))$ con y

Funzionamento del sistema OTP



- Richieste:
- Tutte le implementazioni sia dei server che dei client devono supportare MD5, dovrebbero supportare SHA e possono supportare anche MD4
 - Client e server devono usare lo stesso algoritmo allo scopo di interoperare
 - Altri algoritmi hash potrebbero essere utilizzati per il sistema OTP mediante la pubblicazione delle loro interfacce
- Funzionamento del sistema OTP

Funzionamento del sistema OTP

Scenario: un client desidera autenticarsi ad un server per ottenere un servizio

Informazioni possedute dal Client:

- user_ID: informazione testuale pubblica che identifica univocamente l'utente all'interno della rete
- passphrase: informazione testuale segreta

Funzionamento del sistema OTP

- Informazioni possedute dal Server:
- One Time Password relativa all'ultima autenticazione avvenuta con successo (memorizzata nel database delle password)
 - hash_id: identificatore della funzione hash utilizzata per la computazione delle One Time Password
 - seed: stringa alfanumerica generata alla prima autenticazione di un client
 - sequence number(SN): intero che indica il numero di applicazioni della funzione hash
- Seed, hash_id e sequence number costituiscono una unità di informazione che viene inviata al client, chiamata challenge
- Funzionamento del sistema OTP

Il protocollo OTP opera fondamentalmente attraverso due entità: il generatore delle password, che risiede nel client, ed il server

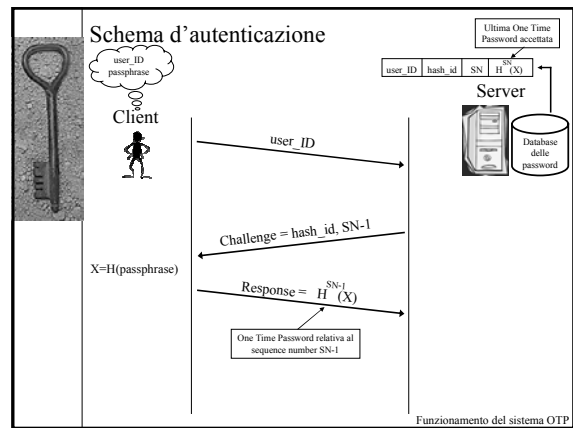
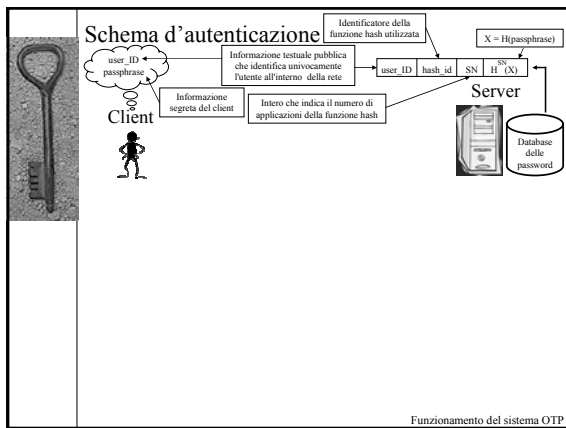
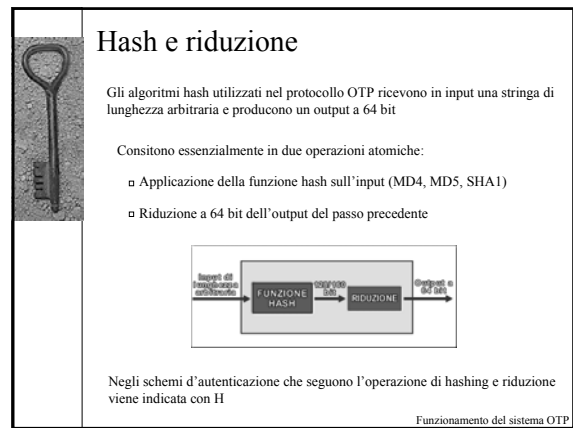
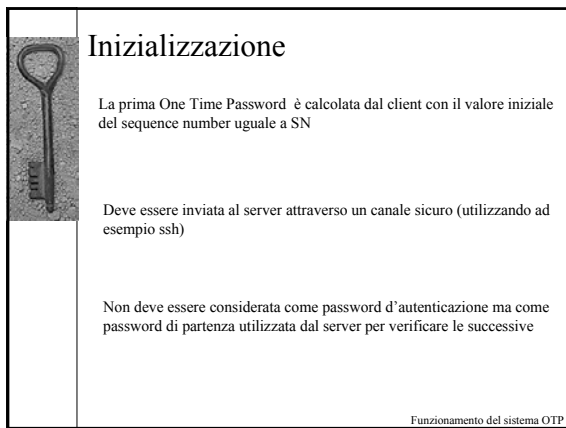
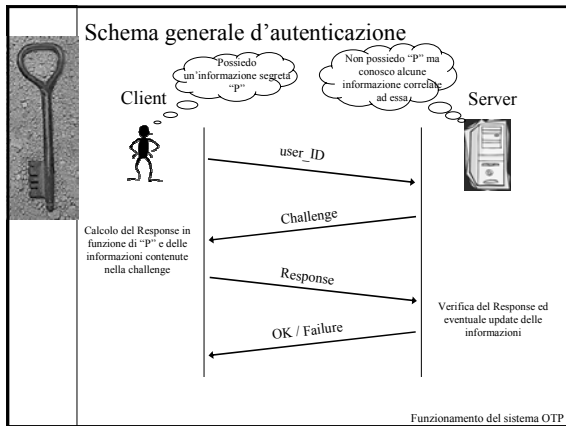
Generatore

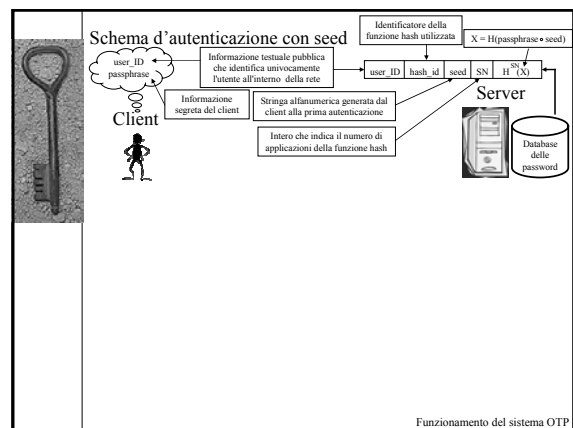
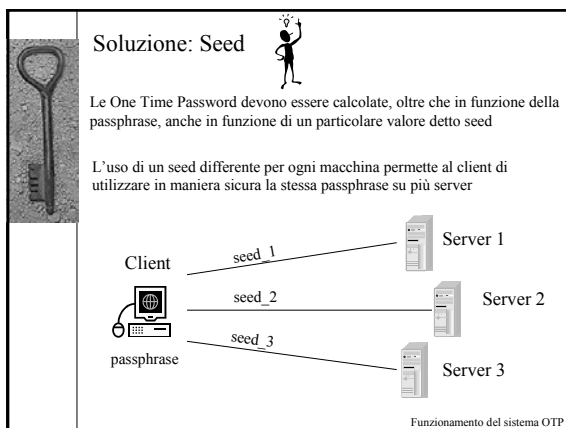
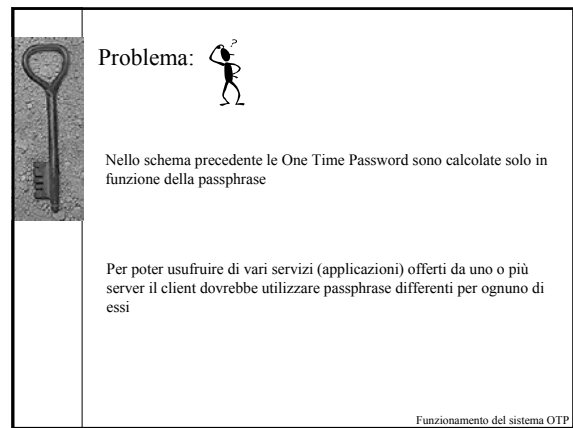
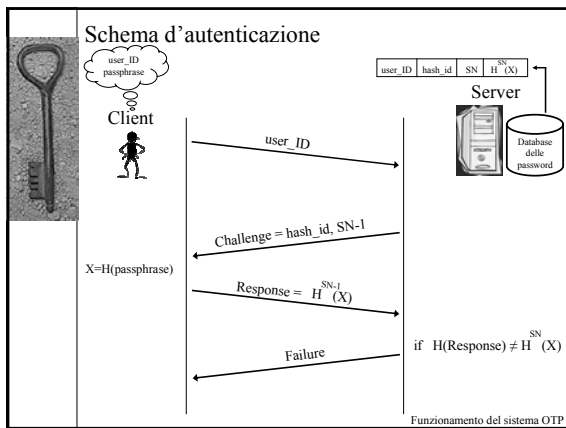
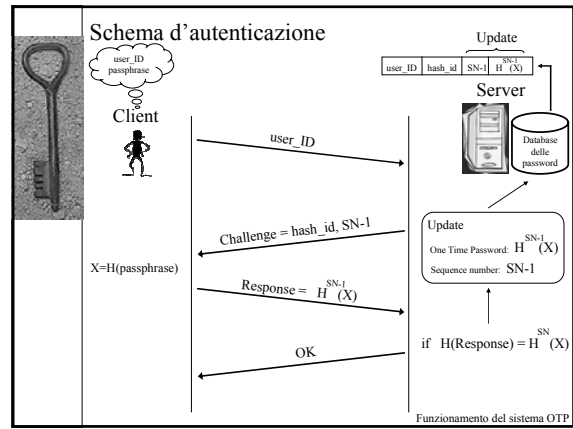
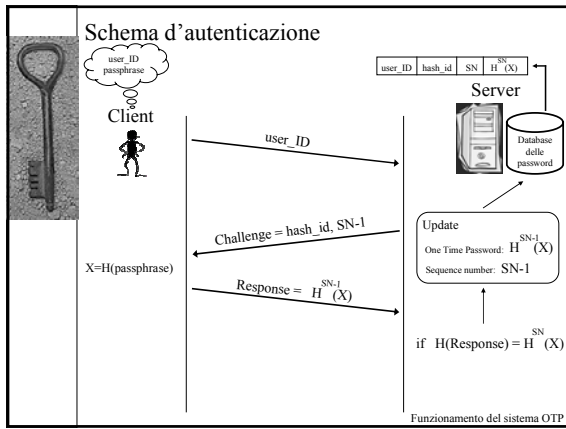
Produce l'appropriata One Time Password utilizzando come parametri la passphrase segreta dell'utente e le informazioni fornite nella challenge dal server

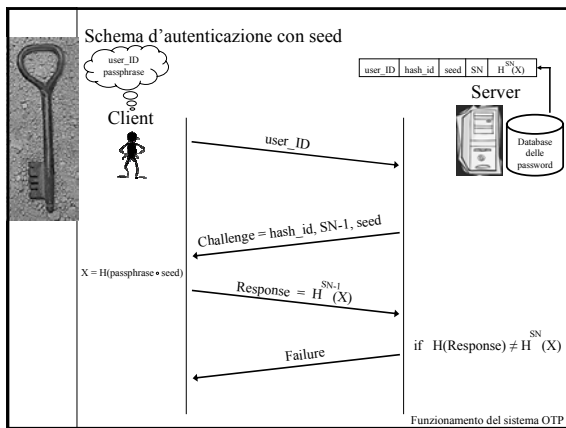
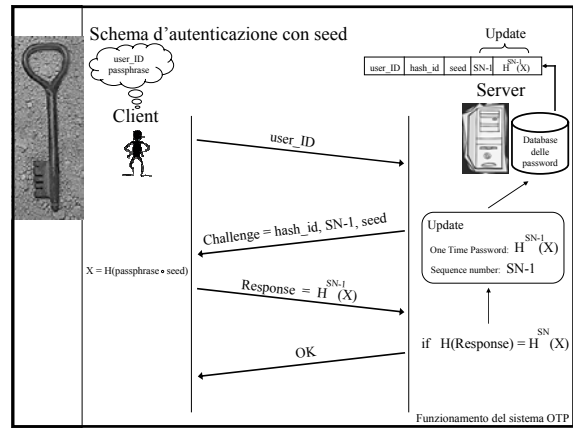
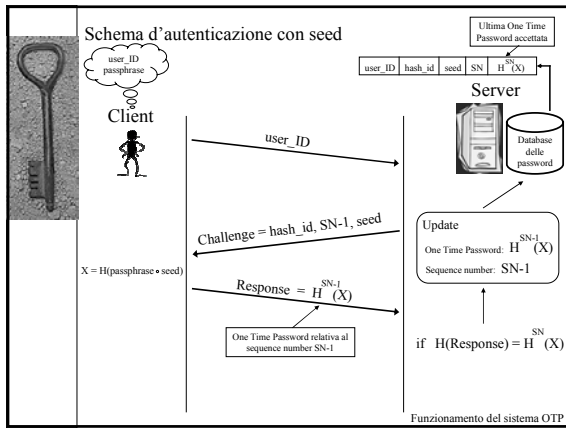
One Time Password

Funzionamento del sistema OTP

- ### Server
-
- trasmette una challenge al client che include degli appropriati parametri per il generatore
 - verifica la One Time Password ricevuta
 - memorizza l'ultima One Time Password valida ricevuta
 - memorizza il sequence number relativo alla One Time Password
 - facilita i cambi di passphrase segreta dell'utente in un modo sicuro
- Funzionamento del sistema OTP







Reinizializzazione

Il numero di applicazioni della funzione hash decresce di uno alla volta

L'utente ad un certo punto deve reiniziare il sistema altrimenti non può più autenticarsi

L'utente seleziona un nuovo seed ed un contatore hash (sequence number), potendo scegliere anche un valore di default

Fornisce tali valori, insieme alla corrispondente One Time Password generata, al server

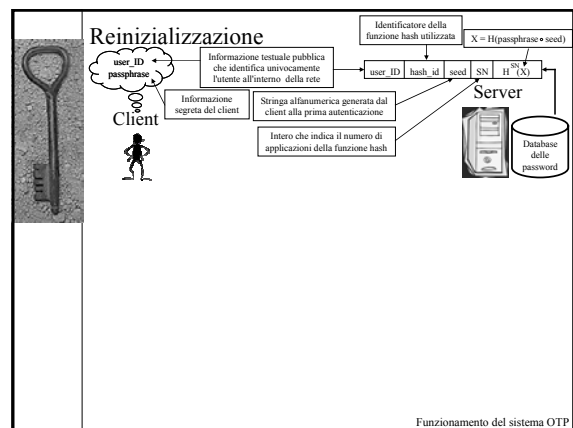
Funzionamento del sistema OTP

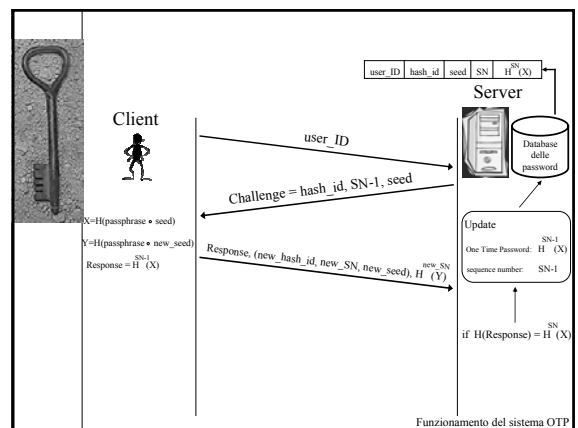
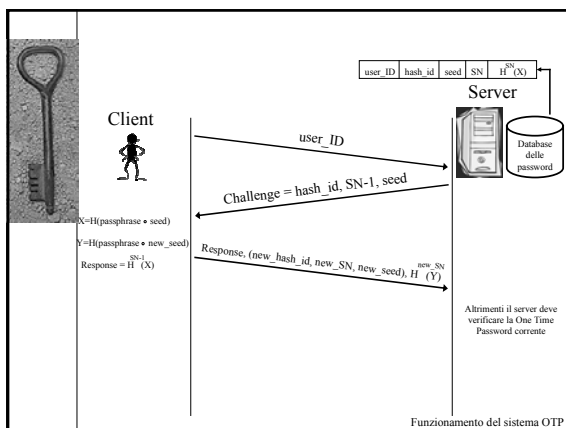
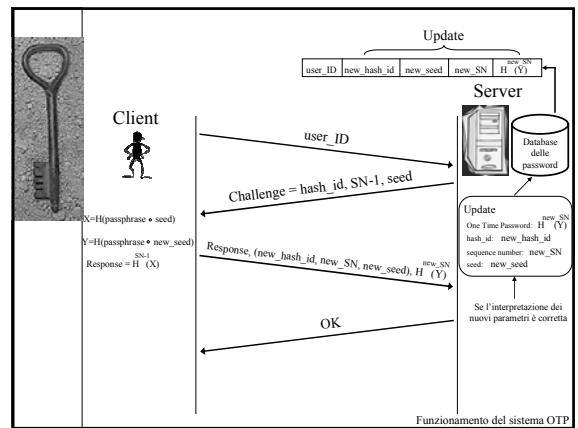
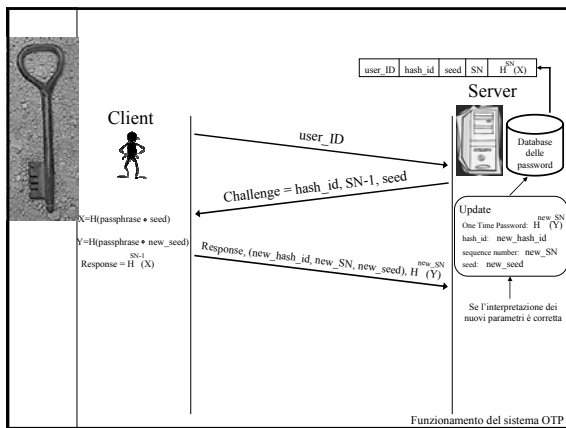
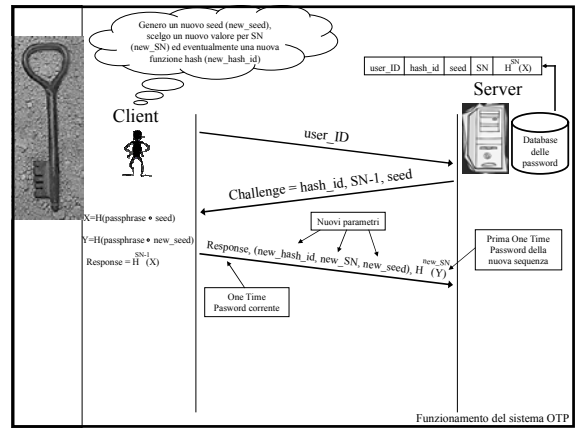
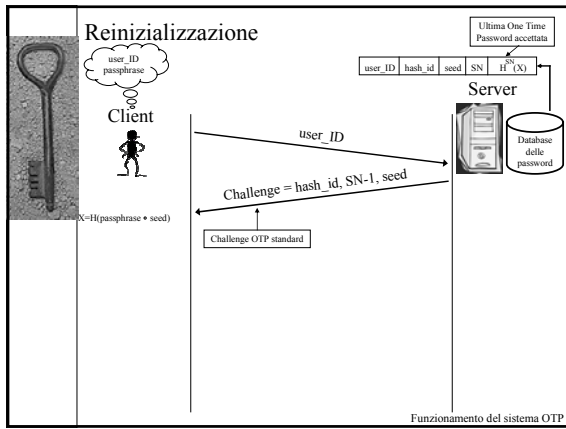
Considerazioni:

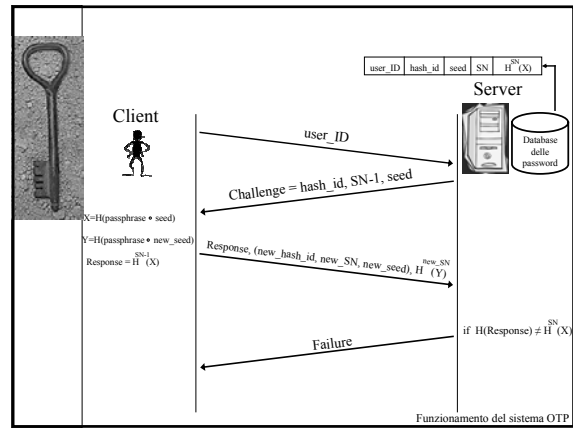
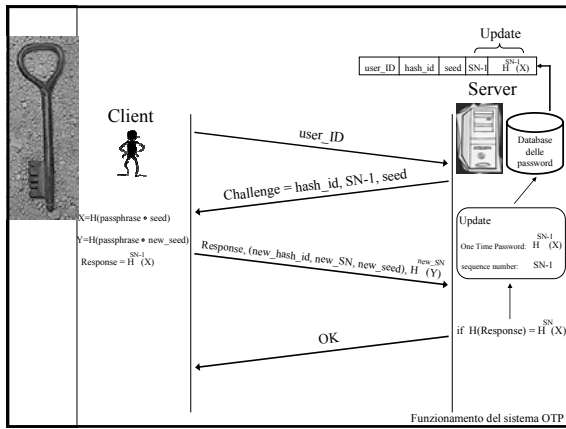
L'utilizzo di seed differenti garantisce l'unicità delle sequenze di One Time Password

L'utente può fornire la corrispondente One Time Password generata con il valore del sequence number uguale a -1, come un controllo di errore

L'utente dovrebbe fornire la One Time Password generata per il vecchio seed e sequence number al fine di proteggere terminali o workstation inattivi







Problema:

Le informazioni contenute nel messaggio di reinizializzazione sono inviate in chiaro dal client sulla rete

Un nemico potrebbe catturare il messaggio di reinizializzazione e sostituire la parte relativa ai nuovi parametri e alla nuova password con informazioni proprie

In tal modo il nemico è capace di ingannare il server ed ottenere l'autenticazione al posto dell'utente legittimo

Soluzione:

L'uso di IPsec o di altre tecniche con la stessa robustezza dovrebbero essere usate contro questo tipo di attacchi

Bisogna impedire ad un utente l'inizio di diverse sessioni di autenticazione contemporanee

Al più una sessione di autenticazione può essere attiva

Cosa tratteremo...

- ✓ Funzionamento del sistema OTP
- ✓ Generazione delle One Time Password
- ✓ Considerazioni sulla sicurezza
- ✓ Esempi di verifica e codifica
- ✓ Risposte estese OTP
- ✓ Meccanismo OTP SASL

Generazione delle One Time Password

I parametri richiesti dal generatore per calcolare la One Time Password a partire dalla passphrase segreta sono forniti dalla challenge inviata dal server:

passphrase
hash_id
sequence number
seed


Generatore

One Time Password

Generazione delle One Time Password

passphrase

- informazione testuale fornita dall'utente
- visibile solo al generatore OTP
- può essere di qualsiasi lunghezza



Osservazioni

Passphrase di almeno 10 caratteri riducono il rischio di tecniche come ricerca esaustiva o attacchi dizionario

Passphrase di lunghezza compresa tra 10 e 63 caratteri consentono l'intercambiabilità dei generatori

Generazione delle One Time Password

hash_id (identificatore funzione hash)

- "md4" indica MD4 Message Digest
- "md5" indica MD5 Message Digest
- "sha1" indica NIST Secure Hash Algorithm Revision 1

sequence number (SN)

Indica il numero di applicazioni della funzione hash


Ad ogni generazione di una One Time Password viene decrementato di uno

Deve essere reinizializzato quando raggiunge il valore 0

Può essere reinizializzato esplicitamente dall'utente

Generazione delle One Time Password

seed



Stringa contenente esclusivamente caratteri alfanumerici la cui lunghezza è compresa tra 1 e 16

Osservazioni

Non deve contenere nessuno spazio vuoto (blank)

Dovrebbe consistere di caratteri alfanumerici del Set ISO-646

Non vi è alcuna differenza tra caratteri maiuscoli e minuscoli ma, prima che esso venga elaborato, il sistema OTP provvede a convertirlo in caratteri minuscoli

Generazione delle One Time Password

Sintassi della challenge

I parametri devono essere separati mediante uno spazio vuoto (definito come un numero di spazi e/o tabulazioni)

`otp-[hash_id] [sequence number] [seed]`

Il comando "otp-" deve essere interamente in minuscolo, come anche l'identificatore dell'algoritmo della funzione hash utilizzata

L'intera stringa deve terminare o con uno spazio o con un new line

Un esempio di challenge OTP è il seguente

```
otp-md5 487 dog2
```

Generazione delle One Time Password


Il processo di generazione delle One Time Password è costituito principalmente da 2 passi: Passo iniziale e Passo di computazione

Passo iniziale

La passphrase viene concatenata con il seed trasmesso in chiaro dal server

Viene applicata la funzione hash a tale concatenazione

L'output della funzione hash è ridotto a 64 bit

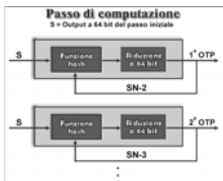


Generazione delle One Time Password

Passo di computazione

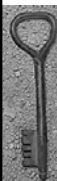
La prima One Time Password utilizzata è prodotta applicando la funzione hash SN-2 volte ad S

La One Time Password successiva viene generata applicando SN-3 volte la funzione hash ad S



Il processo termina quando il sequence number è 1 e la password corrispondente coincide con S


Generazione delle One Time Password



Decodifica del server con dizionario alternativo

- il server converte ogni parola in un numero ad 11 bit
- i numeri a 11 bit sono usati per formare la One Time Password a 66 bit
- il server non ha bisogno di accedere al dizionario alternativo in quanto quest'ultimo non viene usato nella decodifica dalle parole ai numeri ad 11 bit
- un server accettando un dizionario alternativo accetta tutti i dizionari alternativi

Generazione delle One Time Password

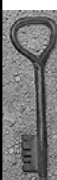


Dizionario per la conversione dal formato a sei parole al formato binario

Tratto dal modulo "put.c" nella distribuzione di riferimento originale della Bellcore

```
{ "A", "ABE", "ACE", "ACT", "AD", "ADA", "ADD",
"AGO", "AID", "AIM", "AIR", "ALL", "ALP", "AM", "AMY",
"AN", "ANA", "AND", "ANN", "ANT", "ANY", "APE", "APS",
"APT", "ARC", "ARE", "ARK", "ARM", "ART", "AS", "ASH",
"ASK", "AT", "ATE", "AUG", "AUK", "AVE", "AWE", "AWK",
"AWL", "AWN", "AX", "AYE", "BAD", "BAG", "BAH", "BAM",
-----
"WITH", "WOLF", "WONT", "WOOD", "WOOL", "WORD",
"WORE", "WORK", "WORM", "WORN", "WOVE", "WRIT",
"WYNN", "YALE", "YANG", "YANK", "YARD", "YARN",
"YAWL", "YAWN", "YEAH", "YEAR", "YELL", "YOGA",
"YOKE" };
```

Generazione delle One Time Password



Algoritmi di riduzione

Un calcolatore, in funzione della sua architettura hardware, può memorizzare i dati in due formati:

"big endian" i dati sono memorizzati a partire dal byte più significativo

"little endian" i dati sono memorizzati a partire dal byte meno significativo

Esempio


Valore esadecimale a 4 byte: 0x11AABBCC

big endian \Rightarrow 0x11 0xAA 0xBB 0xCC

little endian \Rightarrow 0xCC 0xBB 0xAA 0x11

Gli algoritmi hash nel protocollo OTP devono memorizzare l'output nel formato LITTLE ENDIAN

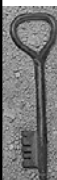
Generazione delle One Time Password



Consistono di tre passi fondamentali

- concatenazione tra passphrase e seed
- applicazione della funzione hash a tale concatenazione
- riduzione a 64 bit dell'output della funzione hash

Generazione delle One Time Password



MD4

```
MD4_CTX md;
unsigned char result[16];
strcpy(buf, seed);
strcat(buf, passwd);
MD4Init(&md);
MD4Update(&md, (unsigned char *)buf, strlen(buf));
MD4Final(result, &md);
for (i = 0; i < 8; i++)
result[i] ^= result[i+8];
```

result conterrà il valore finale a 64 bit

buf contiene la concatenazione tra passphrase e seed

inizializzazione della variabile md

pone nella variabile md il risultato dell'applicazione della funzione hash MD4 alla stringa buf


estrae dalla variabile md l'output a 128 bit della funzione hash ponendolo in result

Il risultato a 128 bit viene ridotto a 64 bit

MD5

Come MD4 ma con l'uso delle funzioni MD5Init, MD5Update ed MD5Final. Più sicuro dell'algoritmo che usa MD4

Generazione delle One Time Password



SHA

```
SHA_INFO sha;
unsigned char result[16];
strcpy(buf, seed);
strcat(buf, passwd);
sha_init(&sha);
sha_update(&sha, (unsigned char *)buf, strlen(buf));
sha_final(&sha);
sha.digest[0] ^= sha.digest[2];
sha.digest[1] ^= sha.digest[3];
sha.digest[0] ^= sha.digest[4];
```

result conterrà il valore finale a 64 bit

buf contiene la concatenazione tra passphrase e seed

inizializzazione della variabile md

pone nella variabile sha il risultato dell'applicazione della funzione hash SHA alla stringa buf

La funzione sha_final produce un message digest di 160 bit che viene memorizzato nell'array di stringhe sha.digest di 32 bit ciascuna

Il risultato di tali computazioni viene memorizzato unicamente nelle stringhe sha.digest[0] e sha.digest[1] continuando così il valore a 64 bit desiderato

Il valore della computazione della funzione hash sha viene memorizzato nel buffer di output result in accordo al formato little endian

```
for (i = 0, j = 0; j < 8; j++, i += 4) {
result[j] = (unsigned char)(sha.digest[i] & 0xff);
result[j+1] = (unsigned char)((sha.digest[i] >> 8) & 0xff);
result[j+2] = (unsigned char)((sha.digest[i] >> 16) & 0xff);
result[j+3] = (unsigned char)((sha.digest[i] >> 24) & 0xff); }
```

Generazione delle One Time Password

Valore del seed

Input:
 Passphrase: A_Valid_Passphrase
 Seed: Length_Okay
 Count: 99
 Hash: ANY

Output:
 ERROR: Seed must be purely alphanumeric

Il seed deve essere costituito esclusivamente da caratteri alfanumerici

Input:
 Pass Phrase: A_Valid_Passphrase
 Seed: LengthOfSeventeen
 Count: 99
 Hash: ANY

Output:
 ERROR: Seed must be between 1 and 16 characters in length

La lunghezza del seed è di 17 caratteri, uno in più del massimo consentito (16)

Esempi di verifica e codifica

Input:
 Passphrase: A_Valid_Passphrase
 Seed: A Seed
 Count: 99
 Hash: ANY

Output:
 ERROR: Seed must not contain any spaces

Il seed non può contenere nessun tipo di spazio vuoto (spazi e/o tabulazioni)

Calcolo della parità

Input:
 Passphrase: A_Valid_Passphrase
 Seed: AValidSeed
 Count: 99
 Hash: MD5

Output:
 Hex: 85c43ee03857765b
 Six Word(CORRECT): FOWL KID MASH DEAD DUAL OAF
 Six Word(INCORRECT PARITY): FOWL KID MASH DEAD DUAL NUT
 Six Word(INCORRECT PARITY): FOWL KID MASH DEAD DUAL O
 Six Word(INCORRECT PARITY): FOWL KID MASH DEAD DUAL OAK

L'ultima delle sei parole dipende dal valore del checksum. Solo una delle quattro parole possibili è corretta

Esempi di verifica e codifica

Esempi di codifica

MD4 ⇒

Passphrase	Seed	SN	Formato esadecimale	Formato a 64 parole
This is test	1234	0	3141-4518 8928-1914	FOAM MFO FPD: 80431193 LACR
This is test	1234	1	8347 3810 1130 8444	SAKD EAD MDC8 BYE COLE EBN
This is test	1234	99	7268 1877 88AC 577A	NOUO OOT BUC 828E HAVI MCEB

MD5 ⇒

Passphrase	Seed	SN	Formato esadecimale	Formato a 64 parole
This is test	1234	0	3057 6473 831A 5746	ANAY EBN FOCQ DAL1193 MAD
This is test	1234	1	8502 4819 4983 27A8	CRBW GRM WU RANR BUC8 EAD
This is test	1234	99	2188 082C 058F 43D3	BUC8 FEEL COO BBNB WARD COCA

SHA1 ⇒

Passphrase	Seed	SN	Formato esadecimale	Formato a 64 parole
This is test	1234	0	8687 6134 5868 960C	INCR SEA ANNE LON8 ABEM TUD8
This is test	1234	1	7965 8954 8F51 029F	RAIS OI FIM OTR8 AWY1 AVY8
This is test	1234	99	3078 1862 058E 5888	RAIS TPEE BPE OADR OISE TRY

Esempi di verifica e codifica

Cosa tratteremo...

- ✓ Funzionamento del sistema OTP
- ✓ Generazione delle One Time Password
- ✓ Considerazioni sulla sicurezza
- ✓ Esempi di verifica e codifica
- ✓ Risposte estese OTP
- ✓ Meccanismo OTP SASL

Risposte estese OTP

Consentono ad un client di richiedere al server la reinizializzazione di una sequenza di One Time Password e/o di cambiare alcuni parametri di autenticazione

Sintassi

Singola linea di testo stampabile terminata da una sequenza di caratteri di newline contenente due o più token separati dal carattere ':'

⇒ <type-specifier>:<arg1>[:<arg2>[:...]]

Tipico specificatore che indica il formato del resto della risposta

Parametri per la risposta estesa OTP. Deve esserne presente almeno uno

Risposte estese OTP

Challenge estesa

Challenge OTP che include la lista delle estensioni supportate dal server

Sintassi

Singola linea di testo stampabile terminata da una sequenza di caratteri di newline o da uno spazio vuoto


⇒ <challenge OTP standard> ext[:<extension set id>[: ...]]

Challenge OTP standard che include gli appropriati parametri per il generatore

Comando UNIX per l'utilizzo delle challenge estese

Insieme delle estensioni supportate dal server

Risposte estese OTP



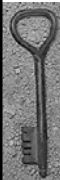
Il server:

- deve essere in grado di ricevere ed analizzare la forma generale di una risposta estesa
- deve essere capace di ricevere, analizzare ed elaborare correttamente tutte le risposte estese specificate
- deve elaborare i campi in maniera case-insensitive
- deve rifiutare qualsiasi autenticazione che tenti di utilizzare una risposta estesa se esso non supporta tale tipo di risposta
- dovrebbe fornire un'adeguata descrizione al generatore se la risposta è stata rifiutata
- deve limitare la lunghezza dell'input ragionevolmente
- deve accettare quantità arbitrarie di spazi bianchi laddove una risposta glielo consenta
- deve essere in grado di ricevere ed elaborare correttamente le risposte OTP standard


Il generatore:

- deve essere capace di generare risposte OTP standard
- deve utilizzare le risposte standard a meno che non sia stata ricevuta una challenge estesa per un particolare server e seed
- deve generare i campi in lettera minuscola
- non deve spedire un tipo di risposta che il server, attraverso una challenge estesa, ha indicato di non supportare

Risposte estese OTP




Problema:




Una risposta OTP standard potrebbe rappresentare una codifica valida sia in esadecimale che nel formato a sei parole. Un esempio di tale situazione è rappresentato dalla risposta "ABE ACE ADA ADD BAD A"

Soluzione:



Il problema può essere risolto facilmente utilizzando le risposte estese "hex" e "word"

Risposte estese OTP



Risposte estese "hex" e "word"

Consentono di specificare esplicitamente la codifica utilizzata per la One Time Password

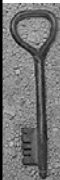
Hanno un campo che contiene una risposta OTP standard codificata nel formato indicato

Sintassi

hex:<numero esadecimale>

word:<sei parole del dizionario>

Risposte estese OTP



Esempio

hex:8720 33d4 6202 9172


word:VAST SAUL TAKE SODA SUCH BOLT

Il generatore:

Dovrebbe generare i token otp-word in lettera maiuscola separati da singoli spazi

Dovrebbe generare numeri esadecimali utilizzando per le lettere caratteri minuscoli

Risposte estese OTP




Risposte estese "init-hex" e "init-word"

Forniscono al client un modo standard per reiniziare le proprie informazioni OTP con un server

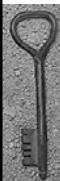
Sintassi

init-hex:<current-OTP>:<new-params>:<new-OTP>

init-word:<current-OTP>:<new-params>:<new-OTP>



Risposte estese OTP



Esempio


init-hex:f6bd 6b33 89b8 7203.md5 499 ke61 18:23d1 b253 5ae0 2b7e

init-word:MOOD SOFT POP COMB BOLO LIFE.md5 499 ke1235:ARTY WEAR TAD RUG HALO GIVE

Se la reinizializzazione va a buon fine, il server deve memorizzare la nuova One Time Password nel suo database come l'ultima One Time Password ricevuta con successo

Il sequence number nella prossima challenge presentata dal server deve essere più piccolo di una unità rispetto al sequence number specificato nel campo "new-params"

Risposte estese OTP



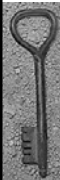

Il server:

- non dovrebbe consentire ad un utente di utilizzare uno stesso valore sia per il seed che per la passphrase
- deve impedire la reinizializzazione della sequenza delle One Time Password qualora venisse specificato un sequence number inferiore ad uno
- deve decrementare il sequence number corrente nel caso in cui non è in grado di elaborare correttamente il valore "new-params" o "new-OTP" e la One Time Password current-OTP contenuta nella risposta di reinizializzazione è valida

Il generatore:

- non dovrebbe consentire ad un utente di utilizzare uno stesso valore sia per il seed che per la passphrase
- deve prevedere dei passi specifici per prevenire dei cicli infiniti dovuti a tentativi di reinizializzazione in caso di fallimento
- dovrebbe fornire all'utente una qualche informazione per indicare la avvenuta reinizializzazione
- non dovrebbe effettuare la reinizializzazione senza il permesso dell'utente, sia per un'istanza specifica che per un'opzione di configurazione
- non dovrebbe ritentare una reinizializzazione fallita senza il permesso dell'utente
- deve avvertire l'utente se il numero di sequenza scende al di sotto di dieci
- deve rifiutare di generare delle One Time Password con un numero di sequenza minore di uno

Risposte estese OTP





Sicurezza

Tutte le considerazioni relative alla sicurezza del sistema OTP valgono anche per il sistema OTP con risposte estese

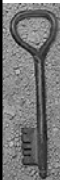
Se il server viene meno alla terza richiesta descritta precedentemente, l'implementazione risulta vulnerabile ad un attacco basato sul replay della parte della risposta contenente la One Time Password corrente (current-OTP)

Risposte estese OTP



Cosa tratteremo...

- ✓ Funzionamento del sistema OTP
- ✓ Generazione delle One Time Password
- ✓ Considerazioni sulla sicurezza
- ✓ Esempi di verifica e codifica
- ✓ Risposte estese OTP
- ✓ Meccanismo OTP SASL



Meccanismo OTP SASL


Simple Authentication and Security Layer

Metodo per includere l'autenticazione a supporto di protocolli basati sulla connessione

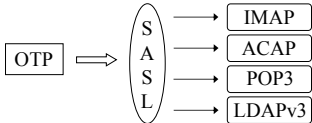
I meccanismi SASL sono identificati da stringhe, lunghe da 1 a 20 caratteri

I nomi dei meccanismi SASL devono essere registrati dalla IANA

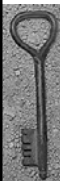
Meccanismo OTP SASL



I meccanismi SASL offrono un modo formale per integrare OTP all'interno dei protocolli dove è supportato SASL includendo IMAP, ACAP, POP3 e LDAPv3



Meccanismo OTP SASL



Il meccanismo OTP SASL definisce le seguenti regole:

- o deve essere utilizzata la sintassi di risposta estesa
- o i server devono supportare le seguenti quattro risposte estese OTP: "hex", "word", "init-hex" e "init-word"
- o i client devono indicare quando l'autenticazione fallisce a causa di un sequence number troppo basso e offrire all'utente un'opzione per resettarlo usando "init-hex" o "init-word"

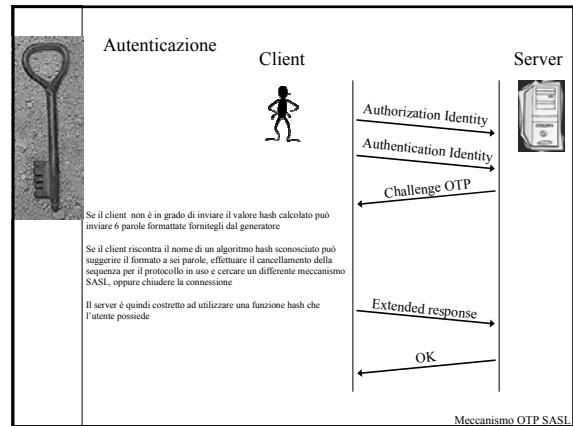
Meccanismo OTP SASL

Autenticazione

Due entità fondamentali:

- Authorization Identity
Spesso conosciuta come user_id, è inviata dal client per loggarsi al server. La stringa vuota è il valore di default utilizzato dall'amministratore di sistema o dai server proxy per effettuare il login con una diversa identità. Deve contenere almeno 255 ottetti e termina con un ottetto nullo (0)
- Authentication Identity
L'identità della passphrase che sarà utilizzata. Può superare i 255 ottetti

Meccanismo OTP SASL



Esempi d'autenticazione

Si riferiscono ad un meccanismo OTP che usa il profilo ACAP di SASL

C: a001 AUTHENTICATE "OTP" {4}

C: tim

S: + "otp-md5 499 ke1234 ext"

C: "hex:5bf075d9959d036f"

S: a001 OK "AUTHENTICATE completed"

Il client invia un messaggio iniziale al server con la Authentication Identity

Il client spedisce la sua user_id

Il server invia la challenge appropriata all'authorization identity ricevuta. Essa termina con il comando "ext" indicante il supporto per le risposte estese

Il client risponde con la One Time Password espressa nel formato esadecimale utilizzando la risposta estesa "hex"

Il server comunica al client l'esito positivo del processo di autenticazione che può quindi definirsi completo

Meccanismo OTP SASL

Con risposta nel formato a sei parole

C: a001 AUTHENTICATE "OTP" {4}
C: tim
S: + "otp-md5 499 ke1234 ext"
C: "word:BOND FOGY DRAB NE RISE MART"
S: a001 OK "AUTHENTICATE completed"

Il server specifica nella challenge, l'utilizzo dell'algoritmo hash "sha1"

C: a001 AUTHENTICATE "OTP" {4}
C: tim
S: + "otp-sha1 499 ke1234 ext"
C: "hex:c90fc02cc488df5e"
S: a001 OK "AUTHENTICATE completed"

Meccanismo OTP SASL

Sicurezza

- non prevede sessioni private, autenticazione server o protezione da attacchi attivi
- è soggetto ad attacchi passivi con dizionario ma il rischio di tali attacchi può essere ridotto scegliendo passphrase opportune
- il database di autenticazione nel server, necessario per l'uso con OTP, non deve essere plaintext-equivalent
- le implementazioni dei server devono essere progettate per resistere ad attacchi race

Meccanismo OTP SASL