Modulo di Sistemi Operativi per il corso di Master
RISS: Ricerca e Innovazione nelle Scienze della Salute
Unisa, 17-26 Luglio 2012



Sistemi Operativi

Sicurezza: attacchi, contromisure, tecniche crittografiche e tool di analisi

Contenuti

- Il problema della sicurezza
- Minacce dai programmi
- Minacce dal sistema e dalla rete
- Crittografia
- Autenticazione dell'utente
- Implementazione di misure di sicurezza
- Firewall
- Classificazione dei sistemi

Obiettivi

- Discutere le minacce alla sicurezza e i tipi di attachi
- Spiegare l'importanza di cifratura, autenticazione ed hashing
- Esaminare l'uso della crittografia nella computazione
- Descrivere diverse contromisure agli attacchi

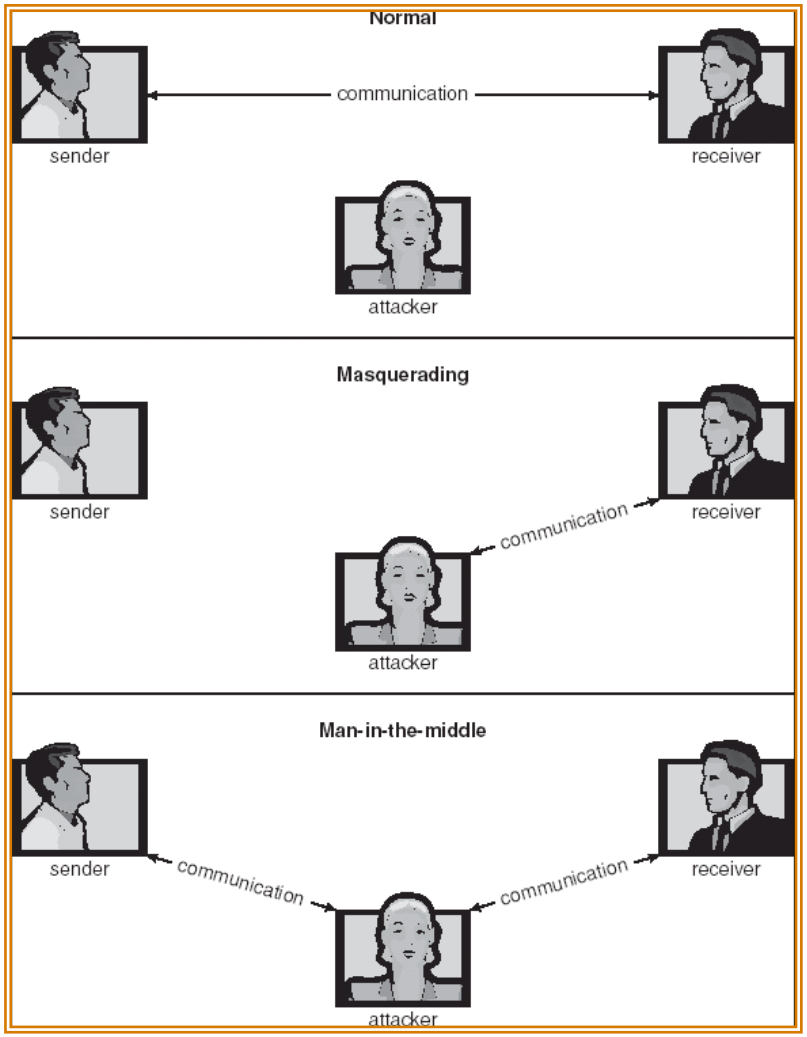
Il problema della sicurezza

- La sicurezza deve considerare l'ambiente esterno al sistema, e proteggere le risorse del sistema
- Intruder e cracker cercano di far breccia nel sistema
- **Minaccia (Threat):** potenziale violazione di sicurezza
- **Attacco (Attack):** tentativo di far breccia nel sistema
- Attacchi possono essere accidentali o voluti
- E' più facile proteggersi contro attacchi accidentali che contro usi maliziosi delle risorse

Violazioni della sicurezza

- Categorie
  - **Breach of confidentiality** (violazione della riservatezza)
  - **Breach of integrity** (violazione dell'integrità)
  - **Breach of availability** (violazione della disponibilità)
  - **Theft of service** (furto di servizio)
  - **Denial of service** (negazione del servizio)
- Metodi comuni d'attacco
  - **Masquerading (Impersonificazione) - violazione dell'autenticità**
  - **Replay attack**
    - **Message modification**
  - **Man-in-the-middle attack**
  - **Session hijacking** (dirottamento della sessione)

Attacchi standard



Misure di sicurezza

- La sicurezza deve essere implementata nei seguenti 4 livelli per essere efficace:
  - Fisico
  - Umano
    - Evitare **social engineering: phishing** (adescamento), **dumpster diving**
  - Sistema Operativo
  - Rete

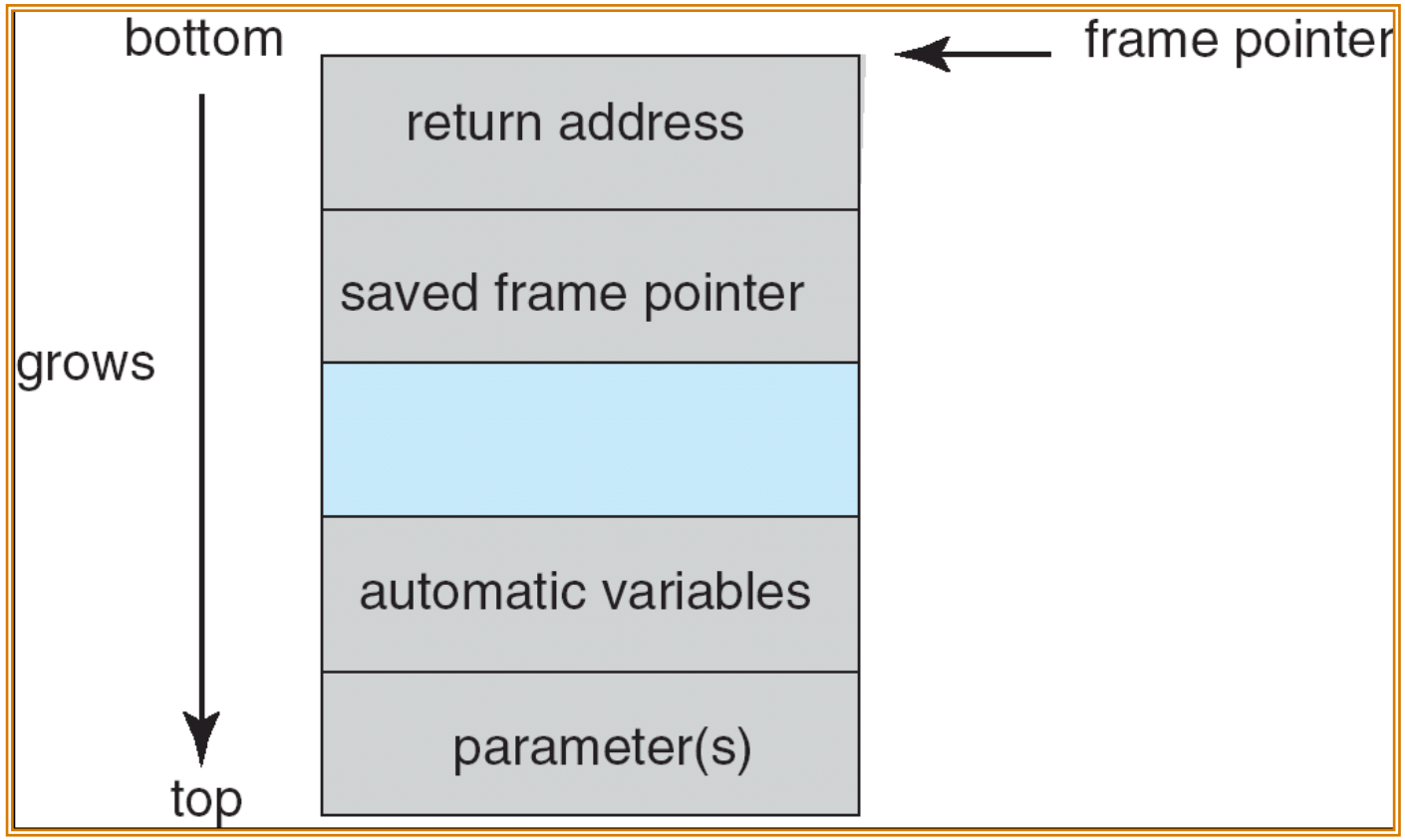
Minacce da programmi

- **Trojan Horse** (cavallo di troia)
  - Segmento di codice che abusa del proprio ambiente
  - Sfrutta meccanismi che permettono a programmi scritti da certi utenti di essere eseguiti da altri utenti
  - **Spyware: pop-up browser windows, covert channel**
- **Trap Door** (porta nascosta)
  - Identificativo utente specifico o password che permette di aggirare le normali procedure di sicurezza
  - Potrebbe essere inserita in un compilatore
- **Logic Bomb** (bomba logica)
  - Programma che provoca un problema alla sicurezza al verificarsi di un determinato evento
- **Stack e Buffer Overflow**
  - Sfrutta un bug in un programma (i.e., manda in overflow o lo stack o buffer di memoria)

Programma C con condizione per Buffer overflow

```
#include <stdio.h>
#define BUFFER SIZE 256
int main(int argc, char *argv[])
{
   char buffer[BUFFER SIZE];
   if (argc < 2)
       return -1;
   else {
       strcpy(buffer,argv[1]);
       return 0;
   }
}
```

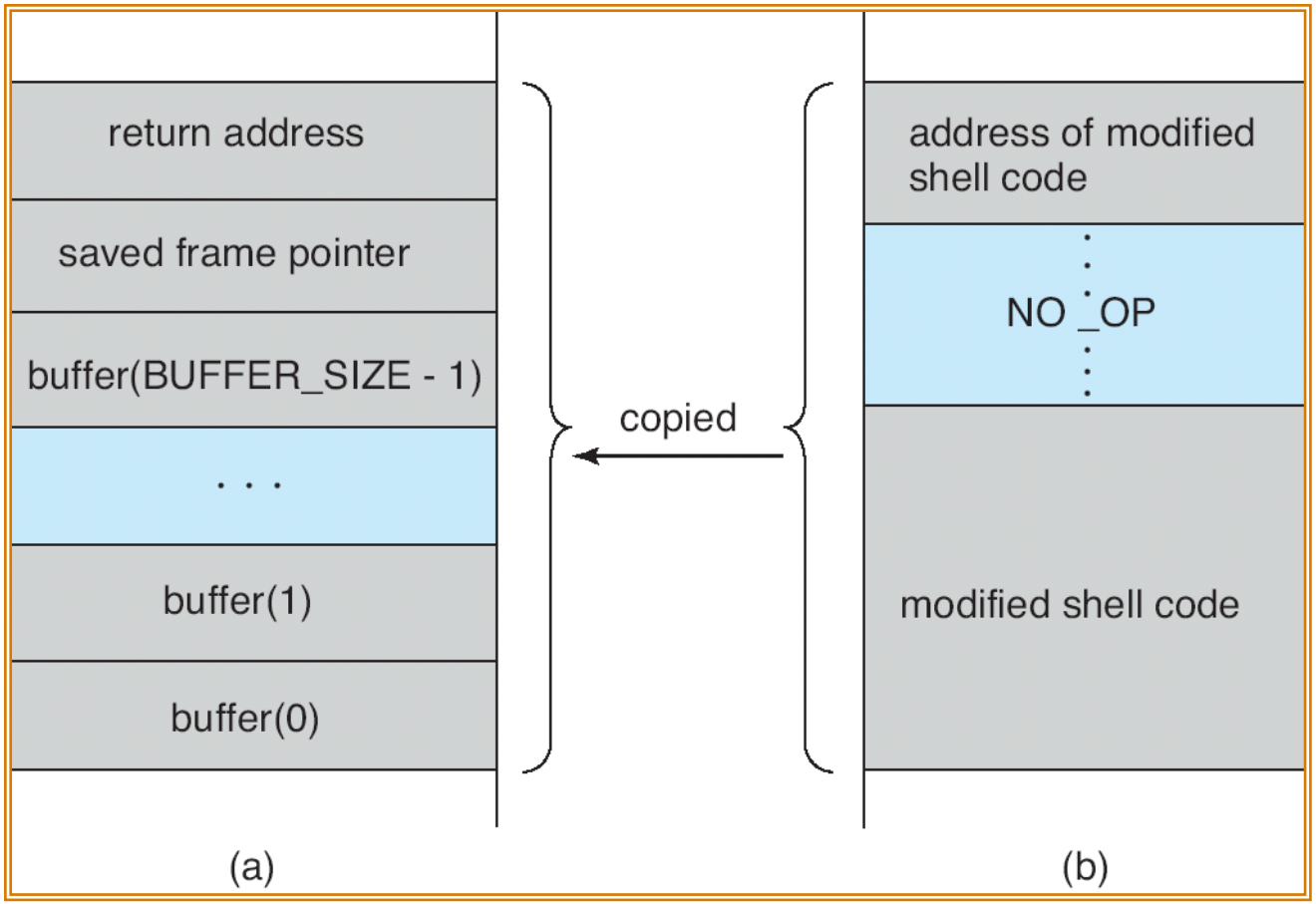
Layout di un tipico frame dello stack



Codice per lanciare una shell

```
#include <stdio.h>
int main(int argc, char *argv[])
{
   execvp(``\bin\sh'',``\bin \sh'', NULL);
   return 0;
}
```

Modifica del frame dello stack



Before attack

After attack

Virus

- Virus
  - Frammento di codice immerso in un programma legittimo
  - Progettato per una specifica architettura di CPU, sistema operativo o applicazioni
  - Solitamente trasmessi tramite email o come macro (pacchetto Office)
    - Visual Basic Macro per riformattare un hard disk

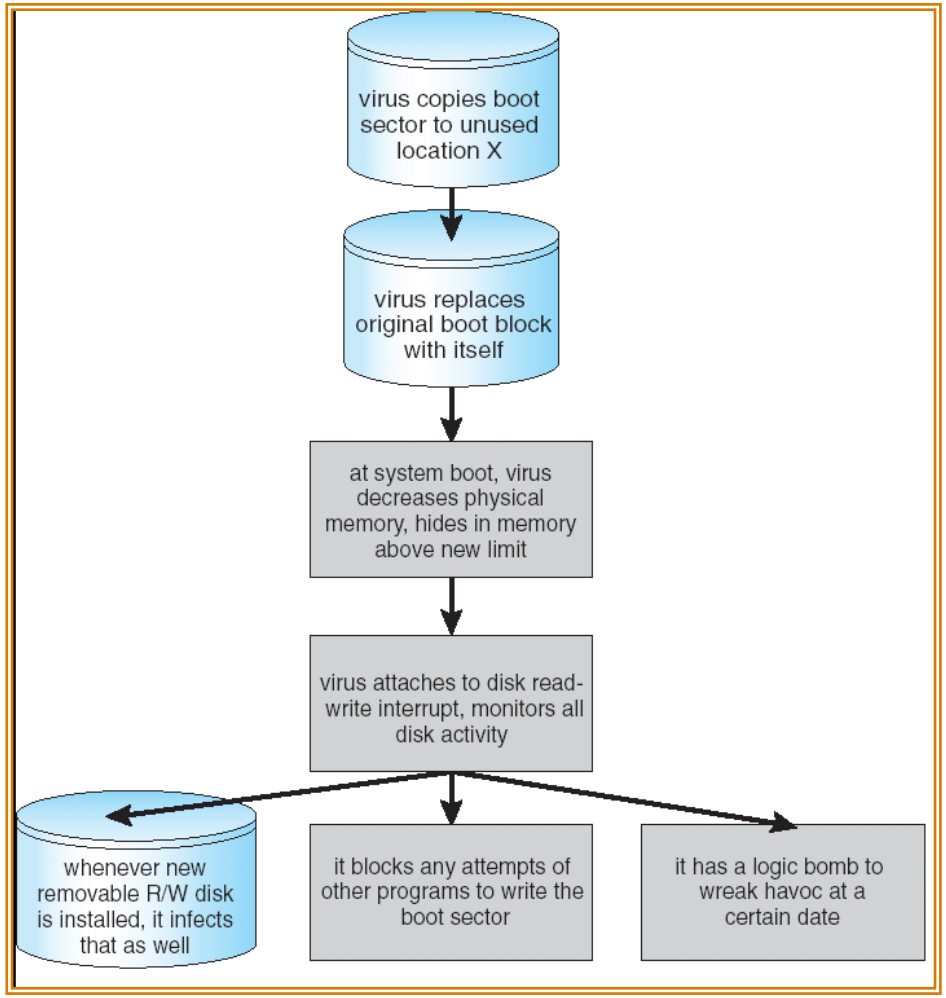
```
Sub AutoOpen()
Dim oFS
  Set oFS =
  CreateObject(''Scripting.FileSystemObject'
  '')
  vs = Shell(''c:command.com /k format
    c:'',vbHide)
End Sub
```

Tipologie di virus

- Un **Virus dropper** inserisce un virus nel sistema

- Esistono molte categorie di virus, diverse migliaia
  - File
  - Boot
  - Macro
  - Source code
  - Polymorphic
  - Encrypted
  - Stealth
  - Tunneling
  - Multipartite
  - Armored

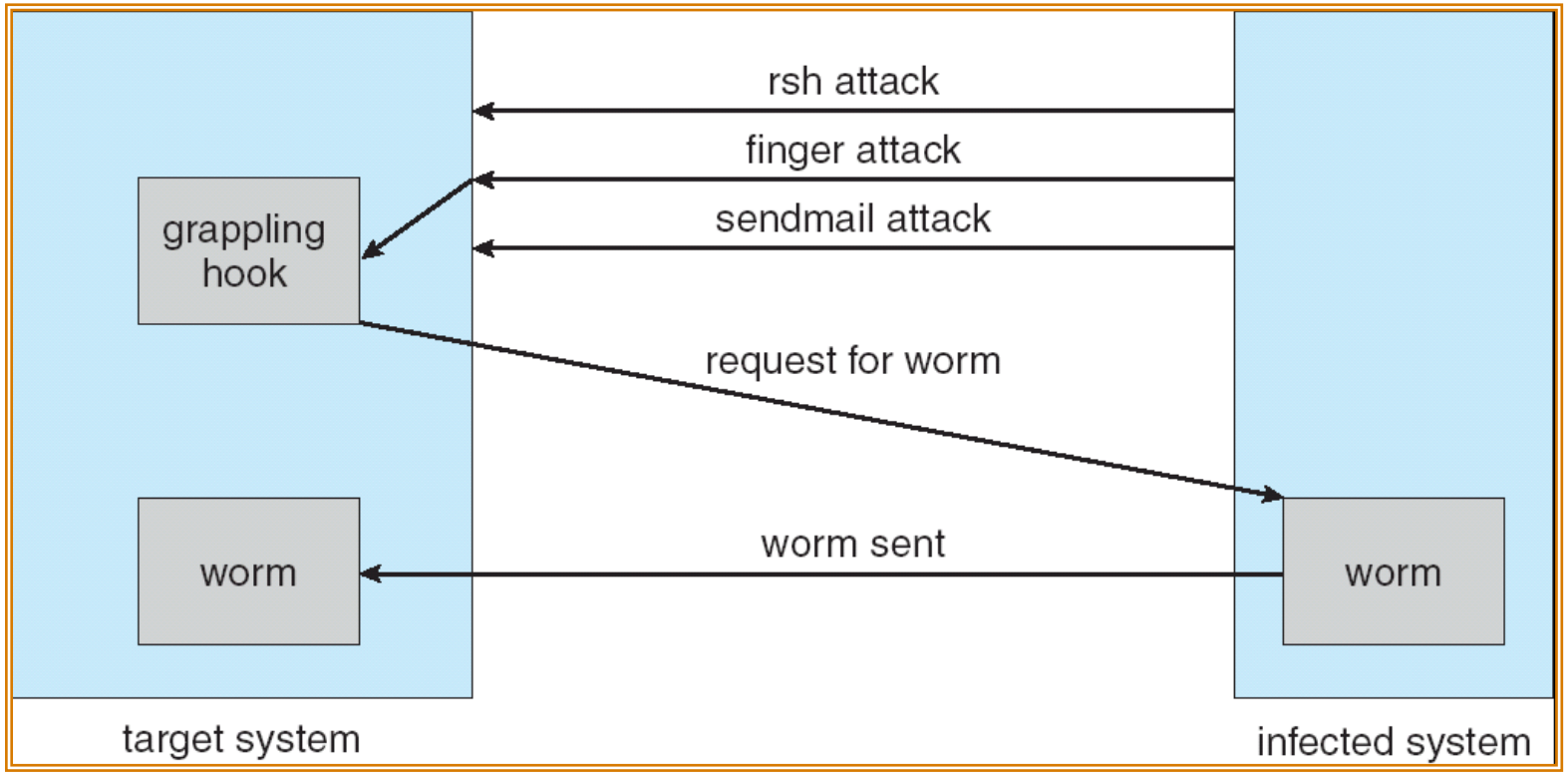
Un virus che infetta il settore di boot



Minacce dal sistema e dalla rete

- **Worm** – usa il meccanismo di creazione di processi;
- Internet worm
  - Sfruttò le caratteristiche di networking di UNIX (e.g., accesso remoto) e bug nei programmi **finger** and **sendmail**
  - Diviso in due parti: una prima parte (grappling hook) caricava il programma principale del worm
- **Port scanning**
  - Tentativo automatizzato di connessione ad un range di porte o di indirizzi IP
- **Denial of Service - DOS**
  - Sovraccaricare il computer target in modo da impedirgli di svolgere lavoro utile
  - Distributed denial-of-service (**DDOS**) è un attacco sferrato da più parti contemporaneamente

Il Worm di Internet - 1988

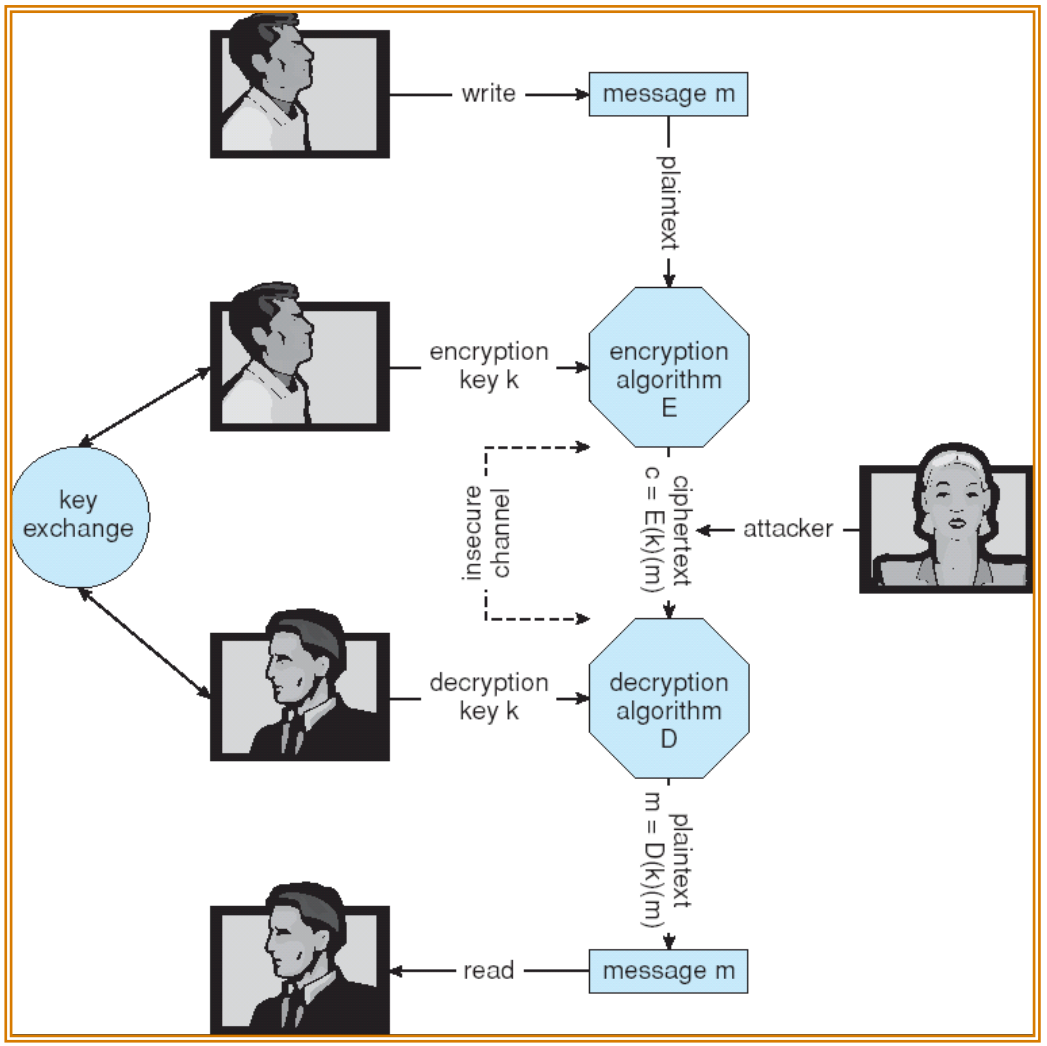


Crittografia

- Maggior strumento disponibile per la sicurezza
  - Sorgente e destinazione dei messaggi non possono fidarsi l'una dell'altra senza la crittografia
  - Fornisce mezzi per restringere i potenziali mittenti (**source**) e/o riceventi (**destination**) di messaggi

- Sistemi basati su segreti: **chiavi (key)**

Comunicazione sicura su un canale insicuro



Cifratura

- Un algoritmo di cifratura consiste di
  - Insieme K di chiavi
  - Insieme M di messaggi in chiaro
  - Insieme C di messaggi cifrati
  - Un insieme di funzioni $\mathbf{E_k} : \mathbf{M} \rightarrow \mathbf{C}$. Cioè, per ogni $\mathbf{k} \in \mathbf{K}$, $\mathbf{E_k}$ è una funzione per generare cifrati da messaggi in chiaro.
    - $\mathbf{E_k}$ per ogni **k** deve essere computabile efficientemente.
  - Un insieme di funzioni $\mathbf{D_k} : \mathbf{C} \rightarrow \mathbf{M}$. Cioè, per ogni $\mathbf{k} \in \mathbf{K}$, $\mathbf{D_k}$ è una funzione per generare messaggi in chiaro da messaggi cifrati.
    - $\mathbf{D_k}$ per ogni **k** deve essere computabile efficientemente

Cifratura

- Un algoritmo di cifratura deve soddisfare la seguente proprietà: dato un cifrato **c** $\in$ **C**, è possibile calcolare **m** tale che $\mathbf{E_k(m) = c}$ solo che si possiede la chiave **k**.
  - Pertanto, per esempio, un computer che possiede **k** può decifrare messaggi cifrati nei corrispondenti messaggi in chiaro usati per produrli, mentre un computer che non possiede **k** non può farlo.
  - Poichè i messaggi cifrati sono inviati su rete, è importante che non sia possibile ricavare **k** dai cifrati.

Cifrari simmetrici

- Stessa chiave usata per cifrare e decifrare
  - $\mathbf{E}_k$ può essere derivata da $\mathbf{D}_k$, e viceversa
- DES è stato l'algoritmo di cifratura simmetrico più usato (creato dal governo USA)
  - Cifra un **blocco** di dati per volta
- Triple-DES $c = E_{k3}(D_{k2}(E_{k1}(m)))$
- Advanced Encryption Standard (AES) ha sostituito DES
- Stream cipher
  - Cifra/decifra un flusso (stream) di byte (i.e trasmissioni wireless)
  - La chiave viene usata come input per un generatore di bit pseudo-random
    - Genera una **keystream**

Cifrari asimmetrici

- Cifratura a chiave pubblica. Ogni utente ha **due chiavi:**
  - chiave pubblica – pubblica, usata per cifrare dati
  - chiave privata – chiave nota solo all'utente, usata per decifrare dati

- Deve essere computazionalmente inammissibile calcolare la chiave privata a partire dalla chiave pubblica
  - Il più comune è l'algoritmo RSA
  - Esistono algoritmi efficienti per verificare se un numero di grossa taglia è primo
  - Non si conoscono algoritmi efficienti per calcolare i fattori primi di un numero composto di grossa taglia

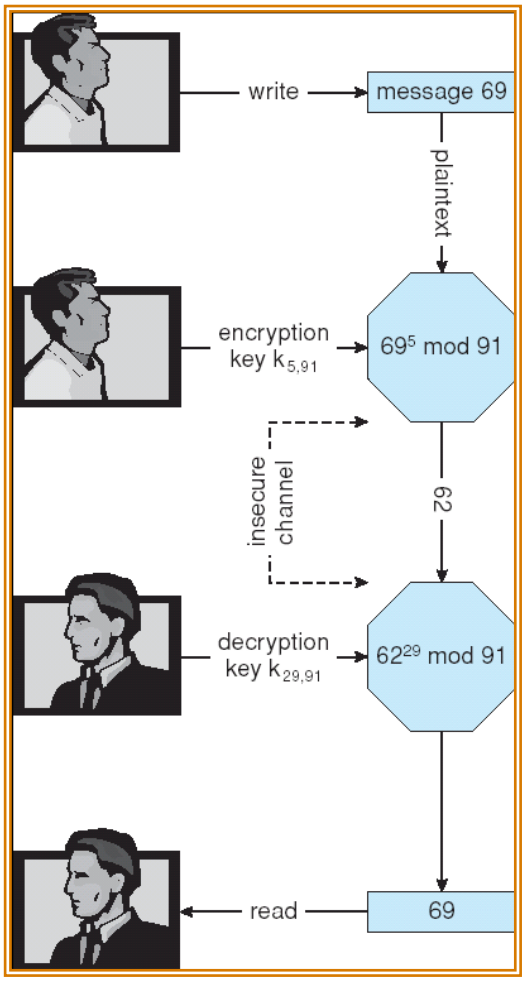
Cifrari asimmetrici

- Formalmente, è computazionalmente inammissibile calcolare **d** mod **N** da **e** mod **N**, e quindi **e** mod **N** può essere resa pubblica
  - **e** chiave pubblica
  - **d** chiave privata
  - **N** è il prodotto di due grandi primi scelti a caso, p e q (per esempio, **p** e **q** sono di 512 bit ciascuno)
  - Algoritmo di cifratura: $E_e(m) = m^e \bmod N$, dove **e** soddisfa $ed \bmod (p-1)(q-1) = 1$
  - Algoritmo di decifratura: $D_d(c) = c^d \bmod N$

Esempio giocattolo

- Siano **p** = 7 e **q** = 13
- Calcoliamo **N** = 7*13 = 91 e (**p**–1)(**q**–1) = 72
- Scegliamo **e** relativamente primo a 72, per esempio 5
- Infine, calcoliamo **d** tale che **ed** mod 72 = 1, ottenendo 29
- Disponiamo ora delle nostre chiavi
  - Chiave pubblica (**e**, **N**) = (5, 91)
  - Chiave privata (**d**, **N**) = (29, 91)
- Cifrare il messaggio 69 con la chiave pubblica dà il cifrato 62
- Il cifrato può essere decifrato con la chiave privata

RSA: cifratura e decifratura



Crittografia

- I cifrari simmetrici sono essenzialmente basati su trasformazioni (e.g., sostituzioni, permutazioni), quelli asimmetrici su funzioni matematiche
  - Cifratura asimmetrica richiede uno sforzo computazionale di gran lunga maggiore
  - Tipicamente non viene usata per cifrare grosse quantità di dati
  - Approccio ibrido

Autenticazione

- Restringe l'insieme dei potenziali mittenti di messaggi
  - Complementare alla cifratura
  - Prova anche che il messaggio non è stato modificato
- Componenti di un algoritmo
  - Un insieme **K** di chiavi
  - Un insieme **M di messaggi**
  - Un insieme ***A* di autenticatori**
  - Un insieme di funzioni $S_k : M \rightarrow A$
    - Cioè, per ogni $k \in K$, S_k è una funzione per generare autenticatori per messaggi
    - S_k per ogni **k** devono essere funzioni efficientemente computabili
  - Un insieme di funzioni $V_k: M \times A \rightarrow \{true, false\}$. Cioè, per ogni $k \in K$, V_k è una funzione per verificare gli autenticatori dei messaggi
    - V_k per ogni **k** devono essere funzioni efficientemente computabili

Autenticazione

- Per un messaggio **m**, un computer può generare un autenticatore **a** $\in$ ***A*** tale che $V_k(m, a)$ = `true` solo se possiede **k**
- Pertanto, il computer che possiede **k** può generare autenticatori su messaggi in modo tale che ogni altro computer che possiede **k** li può verificare
- Computer che non posseggono **k** non possono generare autenticatori di messaggi che possono essere verificati usando **k**
- Poichè gli autenticatori vengono inviati con i messaggi su rete, non deve essere possibile derivare **k** dagli autenticatori

Funzioni hash

- Base per autenticazione
- Crea un piccolo blocco di dati **h**, di taglia fissa (**message digest, valore hash**) a partire dal messaggio **m**
- Una funzione hash **H** deve essere resistente a collisioni su **m**
  - Deve essere inammissibile trova due messaggi **m'** ≠ **m** tali che **H(m)** = **H(m')**
- Uso: se **H(m)** = **h**, allora **m** non è stato modificato
- Funzioni hash comuni sono **MD5**, che produce un message digest di 128-bit, e **SHA** che dà in output un valore hash di 160 bit

MAC

- Cifratura simmetrica usata in algoritmi di autenticazione: **message-authentication code (MAC)**

- Semplice esempio:
  - MAC definito da $S_k(m) = f(k, H(m))$
    - dove **f** è una funzione one-way nel primo argomento
      - **k** non può essere calcolato da $a=f(k, H(m))$
    - La resistenza a collisioni della funzione hash, assicura che nessun altro messaggio genera lo stesso MAC
    - Algoritmo di verifica: $V_k(m, a) \equiv (f(k,H(m)) = a)$
    - Nota che **k** è necessario per calcolare sia S_k che V_k

Firme digitali (digital signature)

- Basato su chiavi asimmetriche
- Gli autenticatori prodotti sono detti **firme digitali (digital signature)**
- In un algoritmo per firme digitali, è computazionalmente inammissibile calcolare S_{ks} da V_{kv}
  - **V è una funzione** one-way
  - Pertanto, **kv** è la chiave di verifica (pubblica) e **ks** è la chiave di firma (privata)
- Algoritmo di firma RSA
  - Simile all'algoritmo di cifratura RSA, ma l'uso delle chiavi è **invertito**
  - Firma di un messaggio $S_{ks}(m) = H(m)^{ks} \bmod N$
  - La chiave **ks è il valore d ed N è il prodotto di p e q (primi grandi scelti a caso)**
  - Algoritmo di verifica: $V_{kv}(m, a) \equiv (a^{kv} \bmod N = H(m))$
    - dove **kv** soddisfa $kv \cdot ks \bmod (p-1)(q-1) = 1$

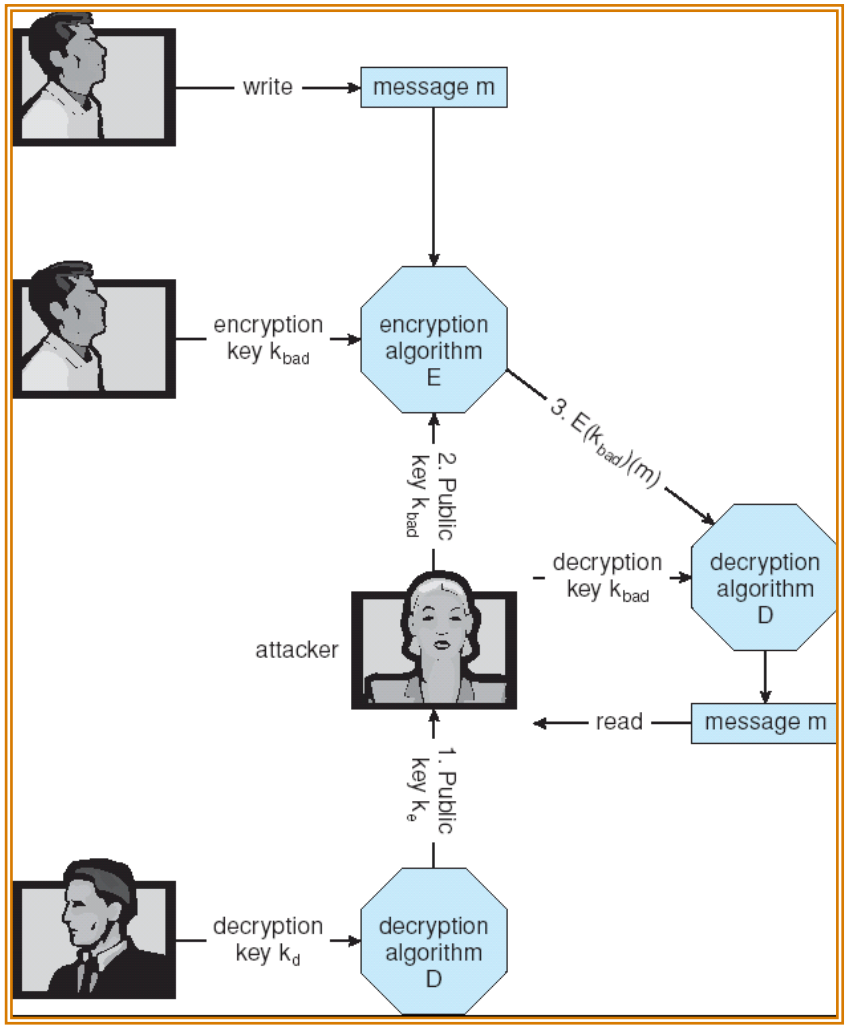
Autenticazione

- Autenticazione vs cifratura
  - Meno computazioni (eccetto per firme RSA)
  - Autenticatore di solito più corto del messaggio
  - Spesso si desidera autenticazione ma non riservatezza
    - Moduli firmati et al
  - Base per il non ripudio (**non-repudiation**)

Distribuzione di chiavi

- Distribuzione di chiavi simmetriche grossa sfida
  - Spesso effettuata in fase di set up
- Chiavi simmetriche possono proliferare
  - Anche le chiavi pubbliche negli algoritmi asimmetrici debbono essere distribuite con cura
    - man-in-the-middle attack

Man-in-the-middle Attack



Certificati digitali

- Stabilisce l'associazione utente - chiave pubblica
- Chiave pubbliche firmate digitalmente da una parte fidata
- La parte fidata riceve una prova dell'identità dell'entità e certifica che la chiave pubblica appartiene a quella entità.
- Le autorità di certificazione sono parti fidate – le loro chiavi pubbliche sono incluse nei browser
  - Esse garantiscono per altre autorità, firmando digitalmente le loro chiavi e così via, creando una rete fidata

SSL

- Inserimento della crittografia in un livello del modello di rete ISO (il livello di trasporto)
- SSL – Secure Socket Layer (anche chiamato TLS)
- Protocollo crittografico che permette a due computer di scambiarsi messaggi in modo sicuro
  - Complesso, con molte varianti
- Usato tra web server e browser per comunicazioni sicure (e.g., invio di numeri di carte di credito)
- Il server viene verificato con un **certificato** che assicura il client di parlare con il server giusto
- La crittografia asimmetrica viene usata per stabilire una chiave di sessione (per un cifrario simmetrico) per cifrare i dati durante la sessione (approccio ibrido)

Autenticazione dell'utente

- Cruciale per identificare correttamente l'utente, poichè la protezione del sistema dipende dall'ID dell'utente
- L'identità dell'utente è spesso accertata tramite **password**
- Le password debbono essere tenute segrete
  - Cambio frequente di password
  - Uso di password "difficili da indovinare"
  - Log di tutti i tentativi di accesso invalidi
- Password possono anche essere cifrate o utilizzabili una sola volta (**one-time** password)

Autenticazione dell'utente

- Protocolli sfida – risposta (challenge – response)
- Code book
- Tecniche biometriche
  - Impronta digitale
  - Misure sulla retina
  - Temperatura
- Dispositivi fisici (smart card, chiavi, penne …)
- Autenticazione con più tecniche contemporaneamente

Implementazione di misure di difesa

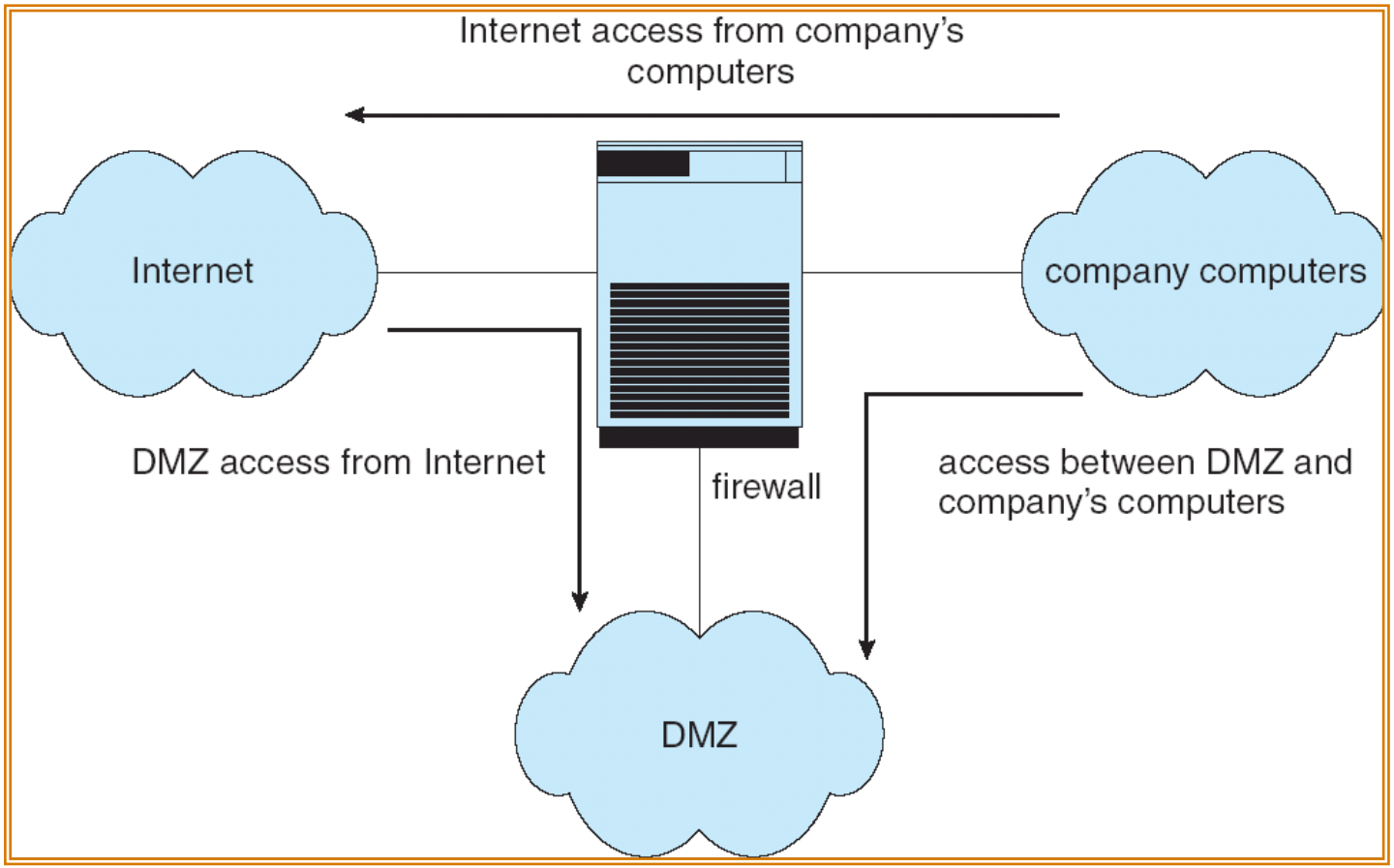
- Una politica di sicurezza descrive cosa deve essere protetto
- La valutazione delle vulnerabilità confronta lo stato reale del sistema/rete con la politica di sicurezza
- La ricognizione delle intrusioni controlla i tentativi di intrusione e i successi
  - La ricognizione basata su firme (Signature-based detection) usa pattern conosciuti
  - La ricognizione delle anomalie (Anomaly detection) controlla le differenze con comportamenti normali
    - Permette di individuare attacchi nuovi (zero-day attacks)
  - Falsi positivi e falsi negativi sono un problema
- Protezione dai virus
- Auditing, accounting, and logging di tutte le attività del sistema e della rete

Firewall

- Un firewall viene posizionato tra computer fidati e non fidati
  - Il firewall limita l'accesso alla rete tra questi due domini di sicurezza
- Può essere aggirato attraverso tunnelling o spoofing
  - Il tunneling permette ad un protocollo vietato di essere usato all'interno di un protocollo permesso (i.e. telnet all'interno di HTTP)
  - Le regole di quei firewall che si basano sul nome dell'host o sull'indirizzo IP sono soggette a spoofing
- Firewall personali: livello software su un host
  - Monitorizza/limita il traffico da e verso l'host
- Un firewall proxy per un'applicazione conosce l'applicazione e la può controllare (i.e. SMTP)

- Un firewall per le chiamate di sistema monitorizza tutte le chiamate di sistema importanti e ne controlla l'uso attraverso regole (i.e. questo programma può eseguire questa system call)

Sicurezza di rete attraverso la separazione dei domini attraverso un Firewall



Classificazione della sicurezza

- Il dipartimento della difesa U.S. usa quettro categorie per la sicurezza dei computer: **A**, **B**, **C**, e **D**.
- **D** – Requisiti di sicurezza minima.
- **C** – Fornisce protezione discrezionale attraverso auditing. Divisa in **C1** e **C2**. **C1** identifica utenti che cooperano con lo stesso livello di protezione. **C2** permette controllo dell'accesso a livello utente.
- **B** – Tutte le proprietà di **C**, ma ogni oggetto ha una specifica label che ne definisce la protezione. Divisa in **B1**, **B2**, e **B3**.
- **A** – Usa tecniche formali per il disegno e la verifica per garantire la sicurezza.