due nodi che comunicano (ad esempio nel gateway d'uscita del server) o che riesce a forzare il redirect del traffico verso la sua postazione, può in pratica ricostruire con estrema semplicità il contenuto delle sessioni applicative, intercettando e ricostruendo il flusso dei dati in chiaro. Nessuna procedura di decrypting è necessaria per appropriarsi delle informazioni trasmesse. Questo tipo di attacco è anche noto come "Man In The Middle" (MITM).
Cifrare i dati, tuttavia, potrebbe non essere sufficiente a impedire lo sniffing. Anche in presenza di sessioni cifrate, infatti, un aggressore può intercettare ed archiviare tutto il traffico per cercare di decifrarlo in modalità offline, ovvero a sessione client/server terminata. Il tipo di algoritmo che l'applicazione implementa e la dimensione della chiave di cifratura utilizzata giocano un ruolo fondamentale nel garantire un'adeguata protezione da questo tipo di attacchi. Se l'applicazione implementa un algoritmo semplice e/o fa uso di una chiave crittografica di dimensioni non adeguate, un aggressore può riuscire a decifrare i dati scambiati, persino in tempo reale. Le principali tecniche utilizzate per violare una chiave crittografica generata attraverso algoritmi simmetrici o di hashing vengono descritte nei paragrafi Brute Forcing e Rainbow Table.

Nella crittografia simmetrica, un messaggio viene cifrato dal mittente con una chiave e decifrato dal destinatario con la stessa chiave attraverso questi semplici passaggi:
il messaggio viene criptato dal mittente:

```
messaggio_cifrato = funzioneCrittografica(messaggio_in_chiaro,
chiave_condivisa);
```

e poi decriptato dal destinatario:

```
messaggio_in_chiaro = funzioneCrittografica(messaggio_cifrato,
chiave_condivisa);
```

La crittografia simmetrica è un esempio di cifratura debole, poiché la chiave può essere divulgata, intenzionalmente o per errore, con molta facilità.

Contromisure

La soluzione è la crittografia asimmetrica, a chiave pubblica/privata, come nelle connessioni SSL/TLS (https).

6.3.2 Brute forcing

Il brute forcing è la tecnica principalmente utilizzata da un aggressore per "rompere" la chiave crittografica di un messaggio testuale o di una sequenza di byte cifrata (ad esempio una password).
Un attacco di brute forcing può, tra l'altro, palesarsi tramite ripetuti tentativi di accesso ad un servizio, utilizzando una lista di username o password predefiniti. Vengono tentate in modo sistematico tutte le possibili combinazioni di un valore crittografato.
Un eventuale match identifica la chiave che può essere impiegata per riportare l'intero messaggio o la sequenza di byte in chiaro (clear-text). Il brute forcing è una tecnica che a seconda dell'algoritmo crittografico utilizzato per cifrare un messaggio, e soprattutto della dimensione della chiave, può non raggiungere l'intento di un aggressore in tempi ragionevoli. Viene solitamente sfruttata per decifrare password o chiavi cifrate con algoritmi simmetrici.

L'attacco di brute force può essere facilitato nei seguenti casi:

- Weak Keys (chiavi deboli): il meccanismo di generazione automatico delle chiavi crittografiche di un'applicazione produce delle Weak Keys. Si tratta di chiavi che, quando utilizzate per cifrare un messaggio, generano in output lo stesso messaggio in chiaro. Questa problematica è strettamente correlata al tipo di algoritmo crittografico utilizzato e può essere occasionalmente riscontrata durante la generazione di chiavi DES, 3DES, RC4, Blowfish, IDEA, etc.
- Collisioni: si tratta di una particolarità che si verifica nel caso degli algoritmi di hashing one-way (MD5, SHA-1, ecc…). Quando un'applicazione utilizza questo genere di algoritmi, ad esempio per confrontare la password fornita da un utente con il valore hash presente in un database, il valore in chiaro proveniente da input viene convertito in hash (una stringa cifrata). L'hash viene poi confrontato direttamente con il valore, sempre cifrato, mantenuto nel database. Per alcuni