Piattaforma AlmaChannel e laboratori virtuali

Tecnologie per la Sicurezza L-S
AA 2005-2006

Anna Riccioni
anna.riccioni@gmail.com

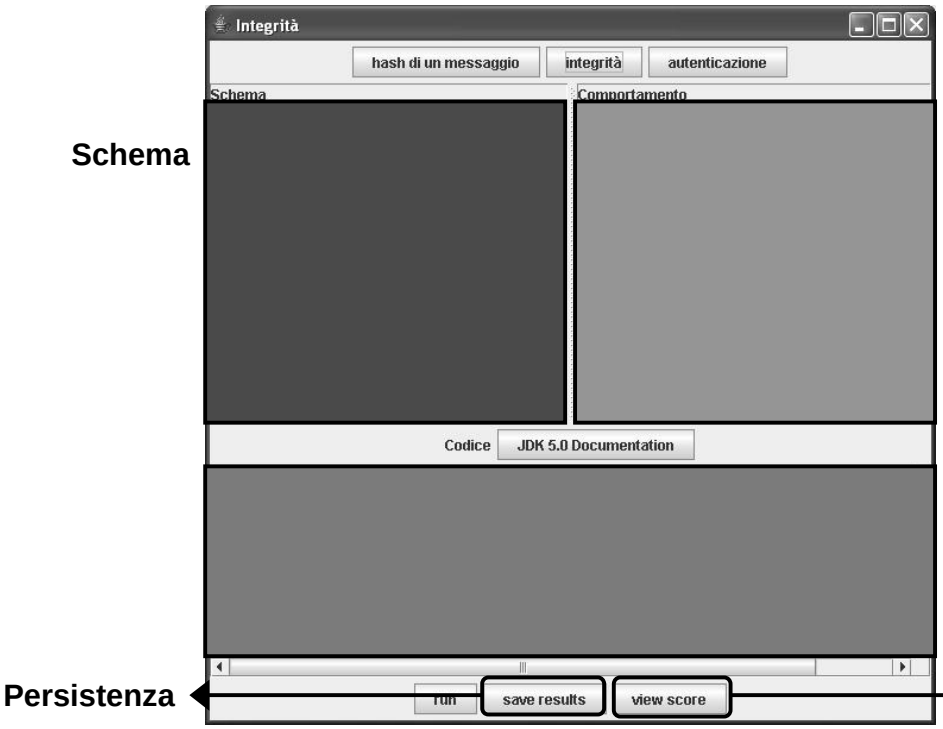
Laboratori virtuali

- Supporto alla didattica tradizionale
- Simulatore come ambiente di lavoro
  - esemplificazione di situazioni reali
  - riproduzione di meccanismi e servizi studiati sotto forma di schemi
  - possibilità di azione e di esecuzione di prove mirate
  - verifica dei concetti teorici studiati
  - sperimentazione

vlab

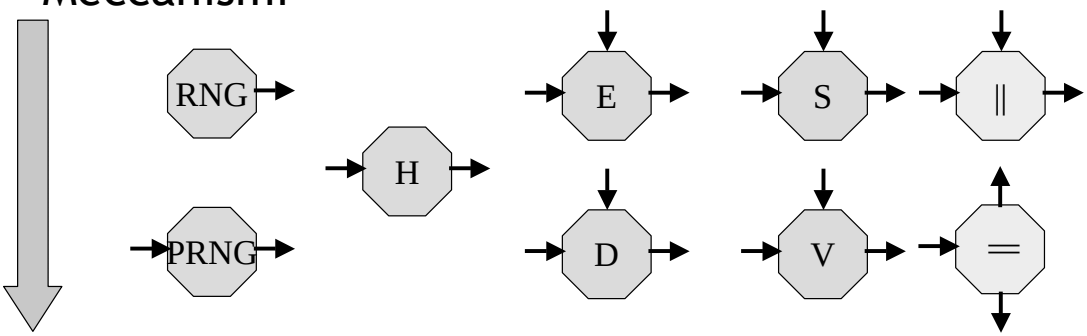
- Framework per l'esecuzione di esercitazioni
- Obiettivi:
  - riprodurre gli schemi visti a lezione
  - costruire il codice Java associato, in funzione dei parametri impostati
  - rendere disponibile un contesto già pronto per l'esecuzione di prove e la sperimentazione di concetti teorici
  - semplificare l'analisi e interpretazione dei risultati grazie ai dati sull'esecuzione

vlab



Esercitazione

- Meccanismi

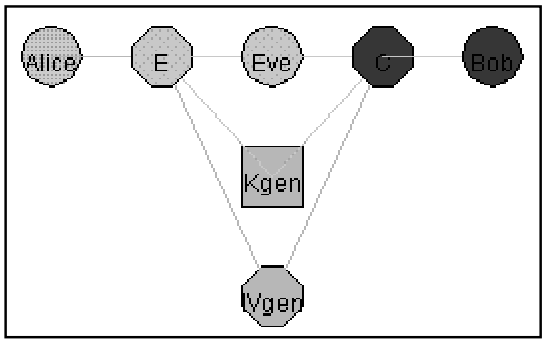


- Componente
  - input
  - output
  - funzione

interattività

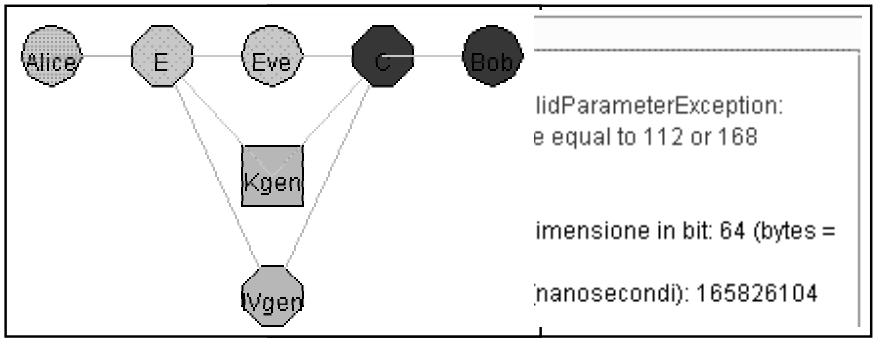
  - parametri
    - funzione specifica
  - esecuzione
    - codice Java
    - comportamento

Codice dei colori: componente



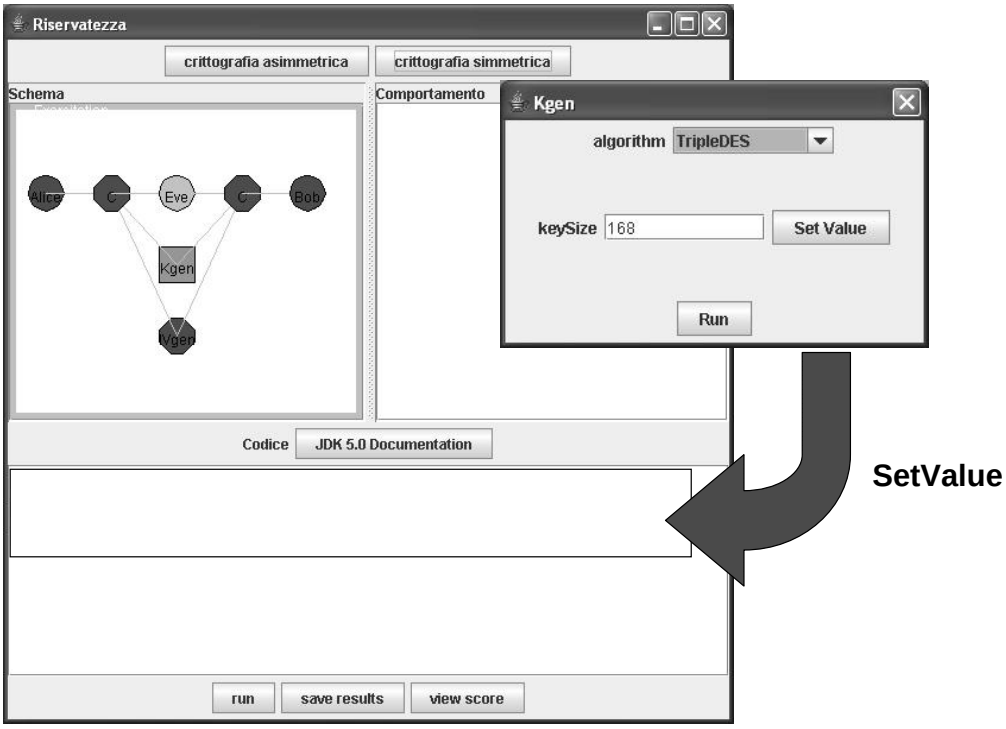
- Rosso: parametri non impostati
- Giallo: input non disponibile o non letto
- Verde: componente eseguito
- Grigio: attori coinvolti nel processo descritto

Codice dei colori: connessione

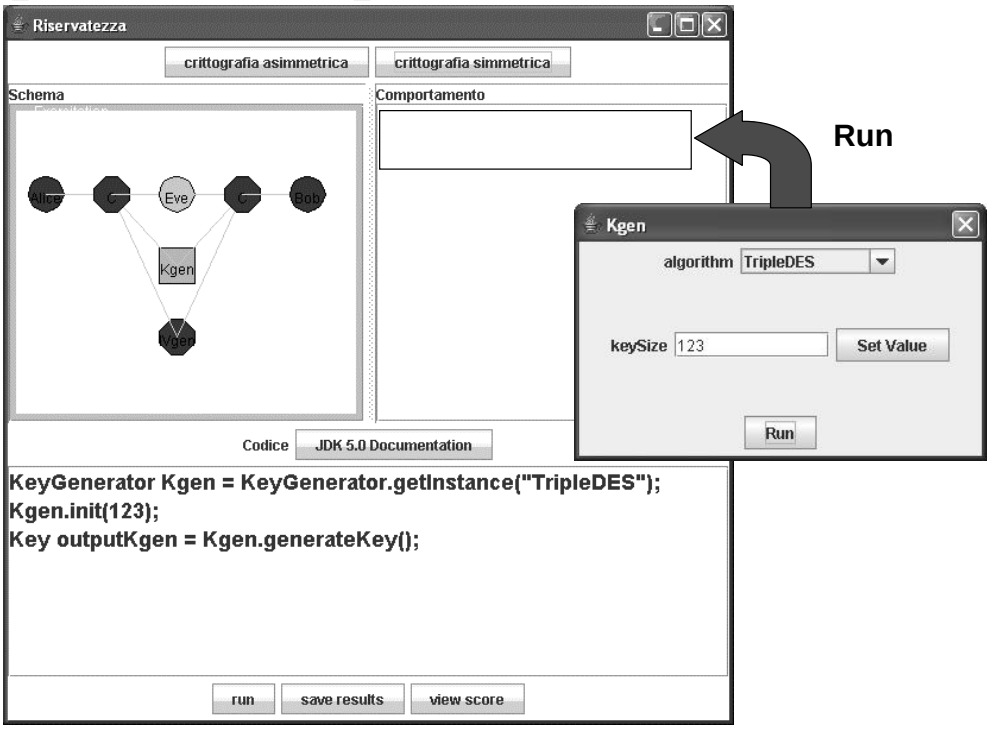


- Verde, a valle di un componente verde: esecuzione andata a buon fine
- Giallo, a valle di un componente giallo o rosso: input non disponibile
- Giallo, a valle di un componente verde: esecuzione terminata con un'eccezione

Impostazione parametri



Impostazione parametri



Vincoli di precedenza

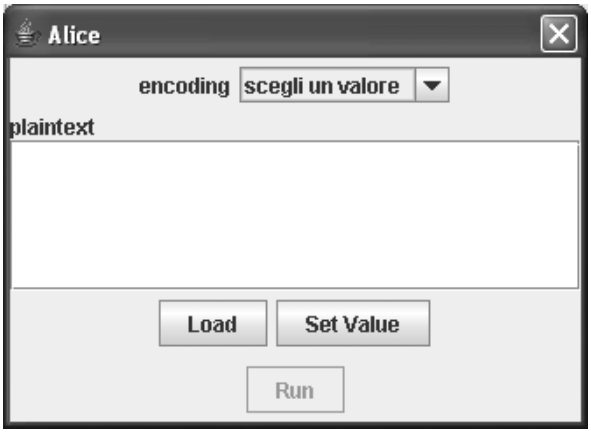
- Esercitazione:

  sequenza ordinata di componenti collegati

- Esecuzione dei componenti:
  - componente eseguibile se tutti i parametri sono stati impostati e tutti gli input sono disponibili
  - un componente non può essere eseguito prima di quello a monte nello schema
- Configurazione dei componenti:
  - nessun vincolo nell'ordine di impostazione dei parametri di componenti diversi
  - alcuni parametri vanno impostati prima che vengano definiti i dati a cui fanno riferimento

Vincoli di precedenza

- Componente di input (Alice)
  - parametro Encoding:
    - UTF-8
    - UTF-16
    - HEX
    - BASE-64
  - dato Plaintext:
    - composto nell'area di testo
    - caricato da file



Vincoli di precedenza

- Testo composto all'interno della text area:
  - al momento dell'esecuzione del componente viene interpretato in funzione della codifica specificata
- Testo importato da file:
  - al momento del caricamento viene visualizzato all'interno della text area e quindi convertito in stringa in funzione della codifica specificata
  - per evitare perdita di informazione occorre prima selezionare un'opportuna codifica, soprattutto nel caso in cui si importi un file non UTF-8 o UTF-16

Encoding

- Codifiche disponibili:
  - UTF-8
  - UTF-16

  stringhe, file di testo
  - HEX
  - BASE-64

  file binari, file in formati complessi
- HEX e BASE-64 consentono di rappresentare in formato testuale il contenuto di byte che le codifiche ASCII o UTF associano a caratteri non stampabili

HEX

- Codifica esadecimale:
  - anziché rappresentare un intero byte, se ne considera la metà
  - le 2^4 possibili combinazioni sono rappresentate dai caratteri [0..9] U [A..F]
  - usando un carattere ASCII per ogni 4 bit si codifica una qualsiasi sequenza di byte in modo stampabile, ma se ne raddoppia la dimensione

BASE-64

- Codifica in BASE-64:
  - anziché considerare un byte alla volta, se ne considerano 3

01010101	00000000	11111111

  - i 24 bit che compongono i 3 byte possono essere suddivisi in 4 blocchi da 6 bit l'uno

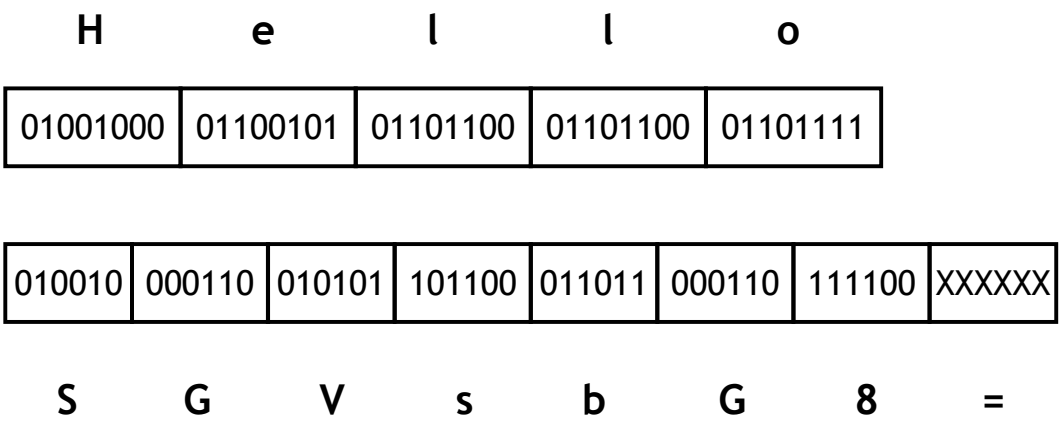
010101	010000	000011	111111

  - un blocco di 6 bit ha 2^6 possibili configurazioni, ognuna delle quali è associata, nella codifica BASE-64, ad uno dei 64 caratteri stampabili nell'insieme [A..Z] U [a..z] U [0..9] U {+} U {/}

BASE-64

- una codifica in BASE-64 trasforma la sequenza di byte di partenza in una nuova sequenza di byte, leggibile in forma testuale, ma la cui dimensione risulta incrementata di un terzo
- poiché si considerano 3 byte per volta, nel caso in cui la sequenza di byte di partenza non abbia una lunghezza multipla di 3 occorre utilizzare un "padding":
  - un eventuale blocco in BASE-64 incompleto viene riempito di "0" fino a raggiungere la dimensione di 6 bit
  - eventuali blocchi da 6 bit mancanti vengono sostituiti dal carattere speciale "="

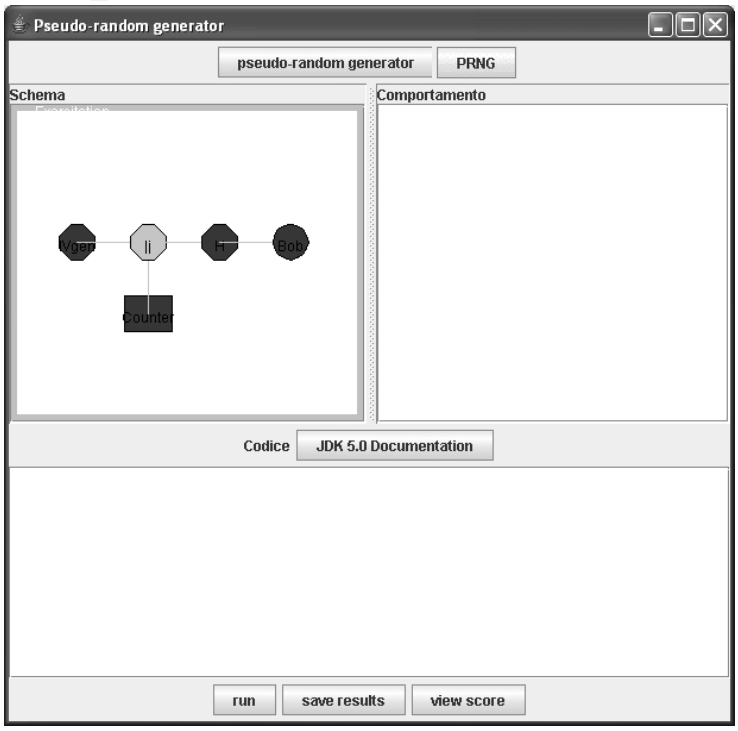
BASE-64: esempio



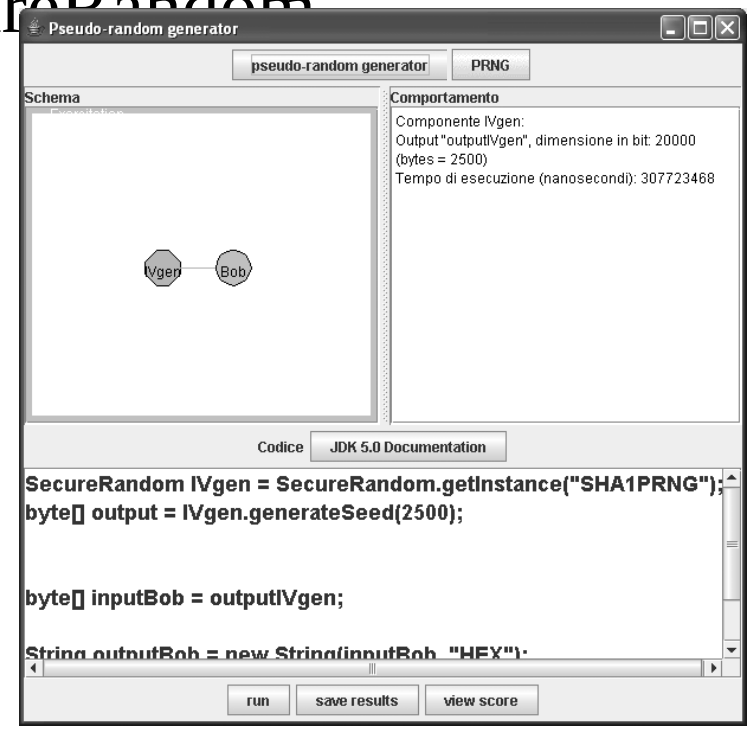
Esercitazioni disponibili

- Numeri pseudocasuali
  - generatore basato su SHA-1
  - classe java.security.SecureRandom
- Integrità
  - hash di un messaggio
  - verifica dell'integrità di un messaggio
  - verifica dell'integrità e autenticazione di un messaggio
- Riservatezza
  - crittografia simmetrica
  - crittografia asimmetrica

Numeri pseudocasuali: SHA-1



Numeri pseudocasuali: SecureRandom



Classe SecureRandom

- Creazione di oggetti SecureRandom attraverso il Factory Pattern
  - i vari provider possono realizzare le proprie implementazioni, non necessariamente basate sull'hash SHA-1

- Istanziare un oggetto SecureRandom:

```
SecureRandom rand =
  SecureRandom.getInstance("SHA1PRNG");
```

  - consente di istanziare un oggetto SecureRandom basato sull'algoritmo specificato, se reso disponibile da almeno uno dei provider installati

Classe SecureRandom

- Istanziare un oggetto SecureRandom:

```
SecureRandom rand =
  SecureRandom.getInstance("SHA1PRNG","BC");
```

  - consente di istanziare un oggetto SecureRandom basato sull'algoritmo specificato e nell'implementazione offerta dal provider richiesto. Se il provider non è disponibile viene sollevata un'eccezione
- Inizializzare un oggetto SecureRandom:

```
rand.setSeed(seed);
```

  - inizializzazione forzata, basata su byte random precedentemente raccolti. In alternativa, l'oggetto può inizializzare autonomamente il proprio stato (meno efficiente)

Classe SecureRandom

- Generare byte random:

```
byte[] bytes = new byte[20];
rand.nextBytes(bytes);
```

  - genera il numero di byte random desiderato dall'utente

- Generare byte di seed:

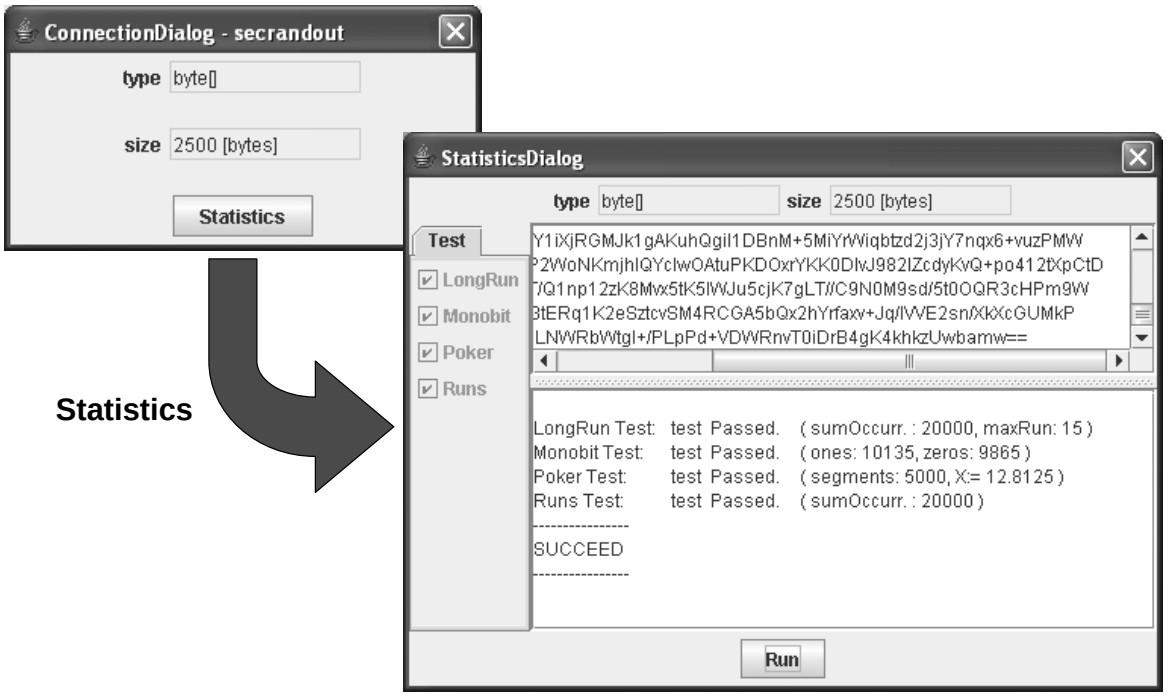
```
byte seed[] = rand.generateSeed(20);
```

  - genera il numero di byte di seed richiesto, utilizzando l'algoritmo di generazione del seme che l'oggetto SecureRandom usa per inizializzare se stesso.

Test statistici

- Verifica della casualità di una sequenza di bit prodotta da un generatore di numeri casuali
- Standard FIPS 140-1 (NIST 1994):
  - campione di 20000 bit
  - superamento di quattro test
    - **monobit**: occorrenze di "0" e "1" circa equivalenti
    - **long run**: corsa più lunga di dimensione inferiore a 26
    - **runs**: occorrenze di tutte le corse di varie dimensioni (1, 2, 3, 4, 5, 6 e superiori) incluso in intervalli prefissati che rispecchiano i valori che si avrebbero in una sequenza casuale
    - **poker**: test del chi quadro per valutare se le occorrenze delle possibili stringhe di 4 bit approssimano il valore atteso per una sequenza casuale

Test statistici

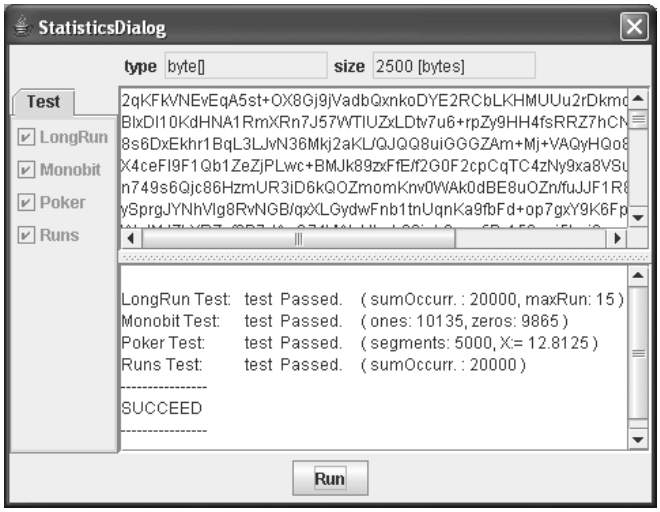


Test statistici

- Prerequisiti
  - standard FIPS 140-1: campione di 20000 bit
  - connessione che contiene un dato di almeno 2500 byte
- Quando utilizzarli
  - verifica dell'output di un generatore di numeri pseudocasuali
  - verifica dell'output di un'istanza della classe SecureRandom
  - analisi dei byte prodotti da un'operazione di cifratura
  - …

Test statistici

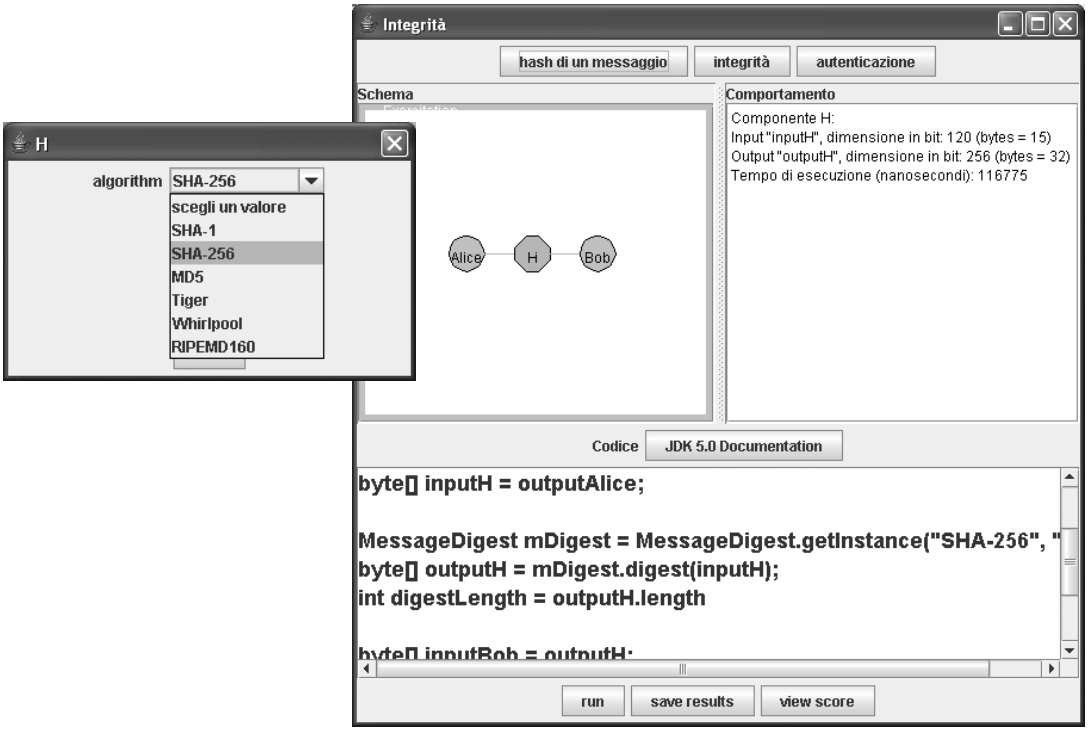
- Come interpretare i risultati
  - confronto dei parametri caratteristici dei test con quelli considerati ammissibili dal FIPS 140-1



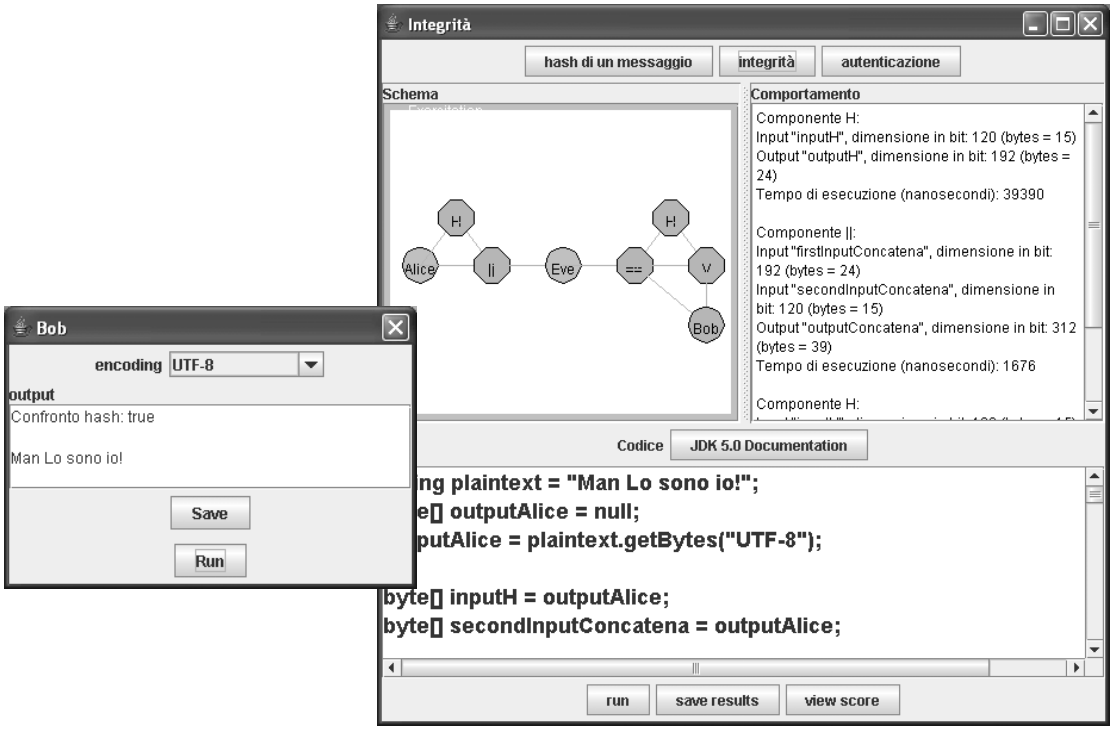
- **long run**: X < 26
- **monobit:** 9725 < X < 10275
- **poker:** 2.16 < X < 46.17
- **runs:**

Lunghezza della corsa	Intervallo ammissibile
1	2315 - 2685
2	1114 - 1386
3	527 - 723
4	240 - 384
5	103 - 209
6+	103 - 209

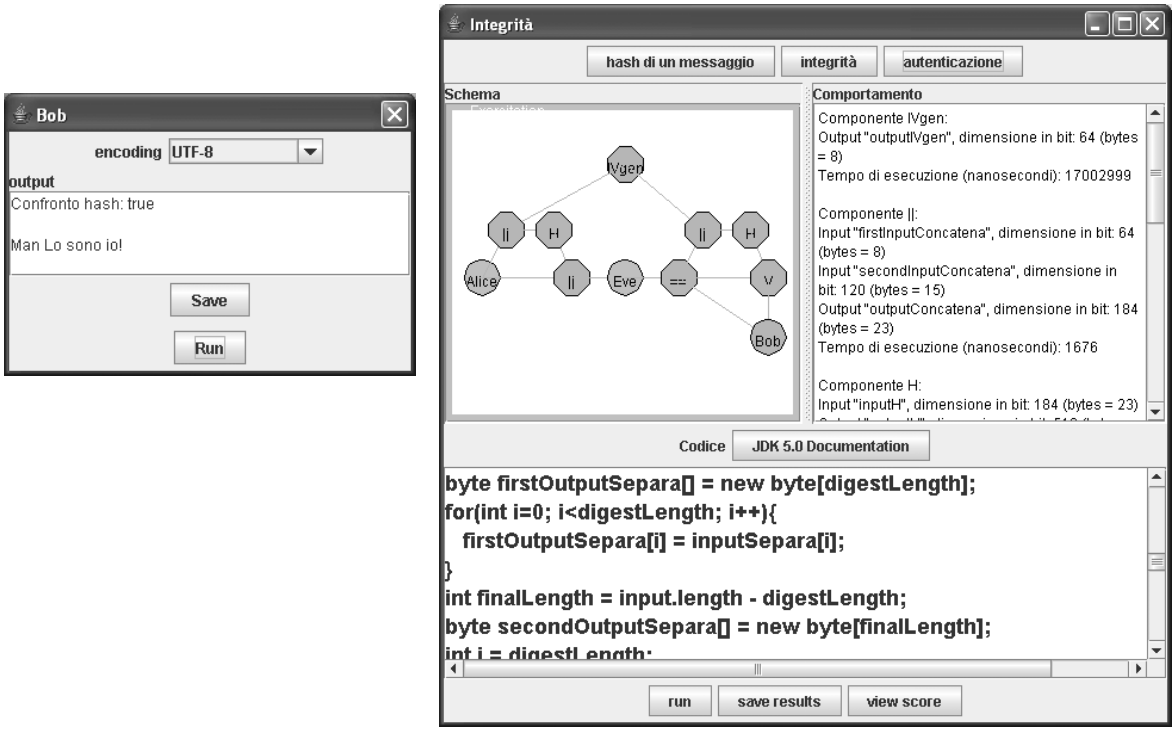
Integrità: hash



Integrità: verifica hash



Integrità: autenticazione



Piattaforma AlmaChannel

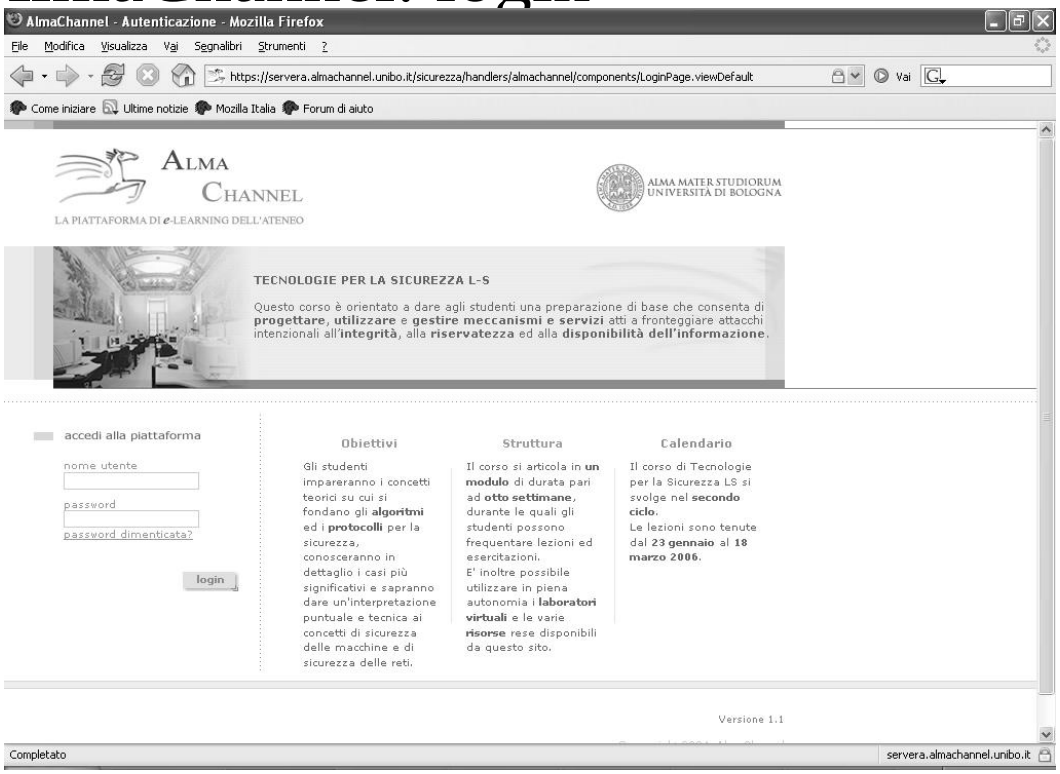
- Piattaforma di e-learning d'Ateneo
- Supporto alla didattica attraverso:
  - strumenti di comunicazione
    - sincrona (chat)
    - asincrona (forum, FAQ, annunci)
  - aree di scambio e condivisione risorse
    - area pubblica
    - area privata
    - area elaborati (visibile all'utente e ai docenti o ai tutor)
  - aree per la pubblicazione di materiali didattici e risorse addizionali (ad es. laboratori)
  - test di autovalutazione
  - …

Istanza del corso su AlmaChannel

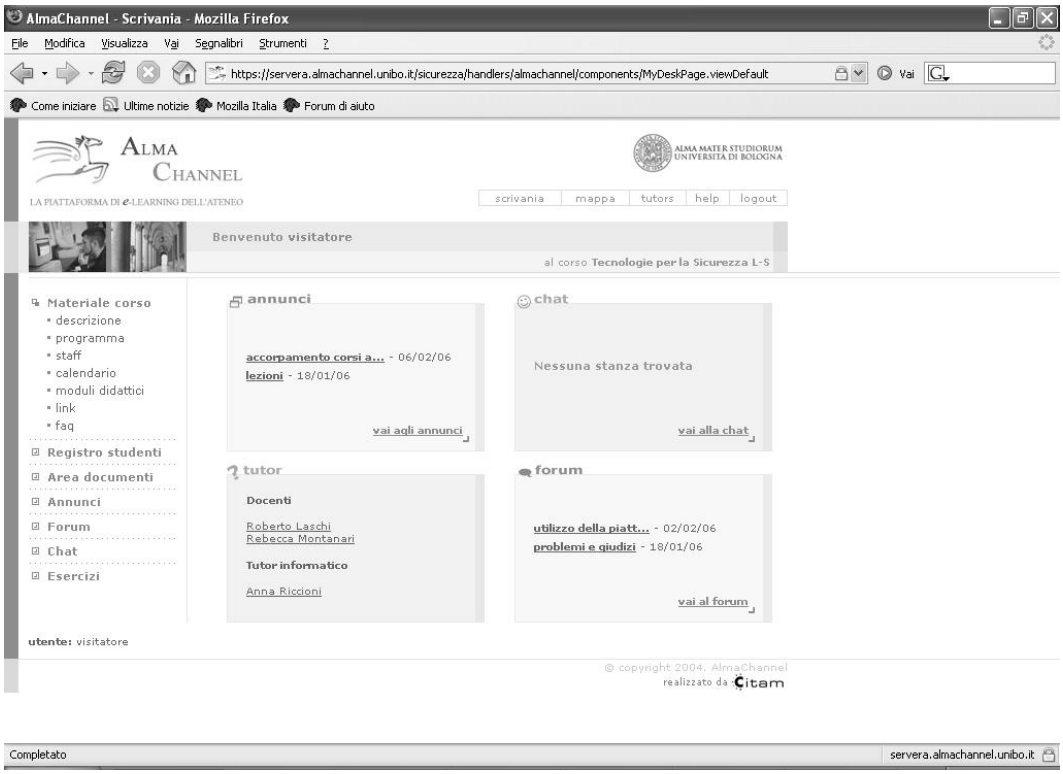
- URL:
  - http://servera.almachannel.unibo.it/sicurezza

- Login
  - Personale:
    - Username / password possono essere richiesti a lezione o via mail (anna.riccioni@gmail.com)
  - Anonimo:
    - Username = **visitatore**
    - Password = **visitatore**

AlmaChannel: login

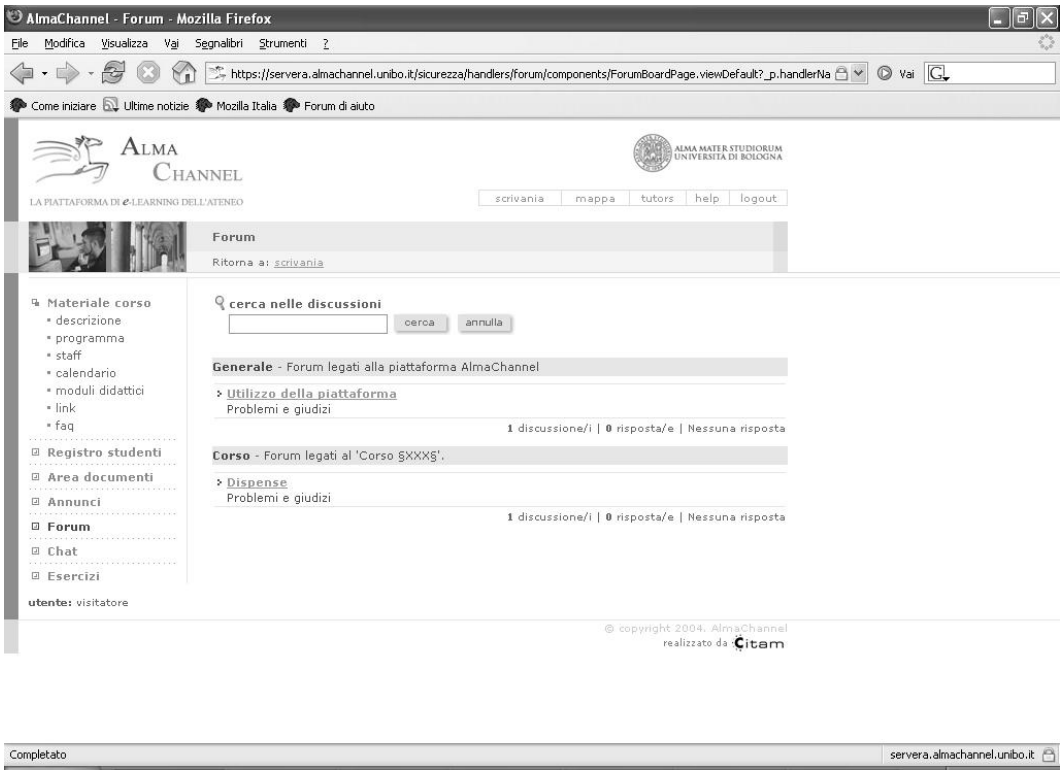


AlmaChannel: scrivania

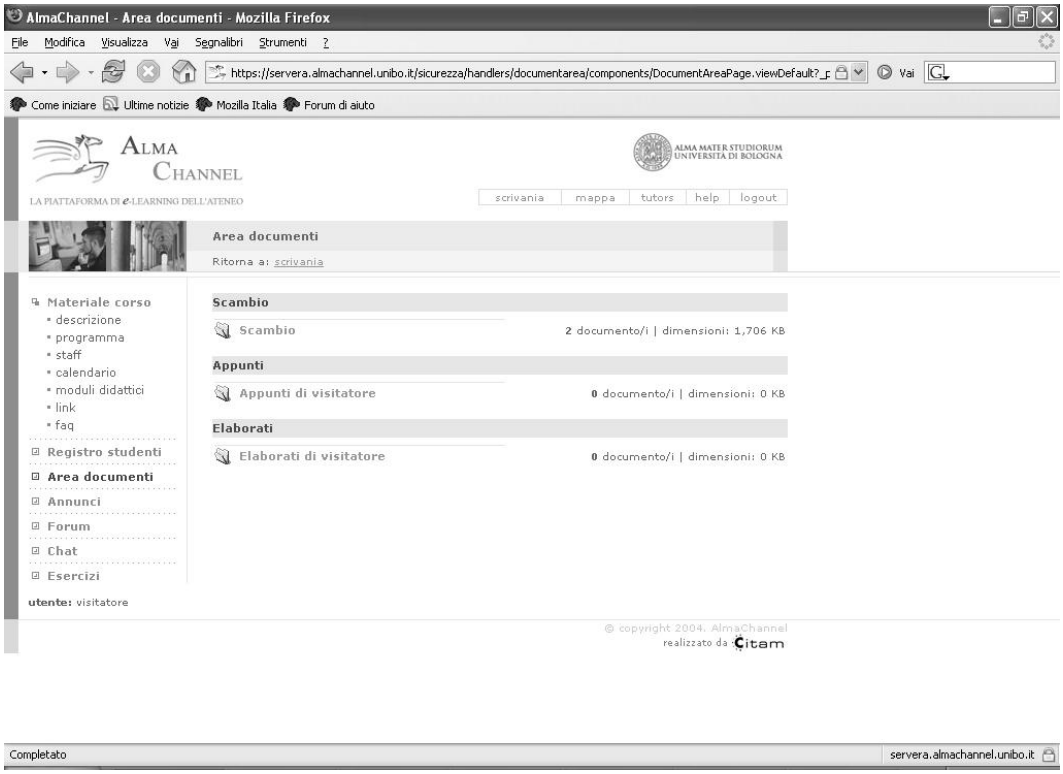


AlmaChannel: moduli didattici

AlmaChannel: forum



AlmaChannel: aree documenti



AlmaChannel: documenti condivisi