



TECNICHE DI COMPRESSESIONE CRITTOGRAFATA

CONTENT

01

Cos'è un Compression Crypto-
System?

02

A Huffman Cose Based Crypto-System

03

Integrated Encryption in Dynamic Arithmetic
Compression

04

Privacy in lossless compression with
Burrows-Wheeler Transform

PREVIEW

Paper 1

Utilizzo di permutazioni pseudocasuali nell'albero di Huffman per garantire resistenza su attacchi crittografici.

Paper 2

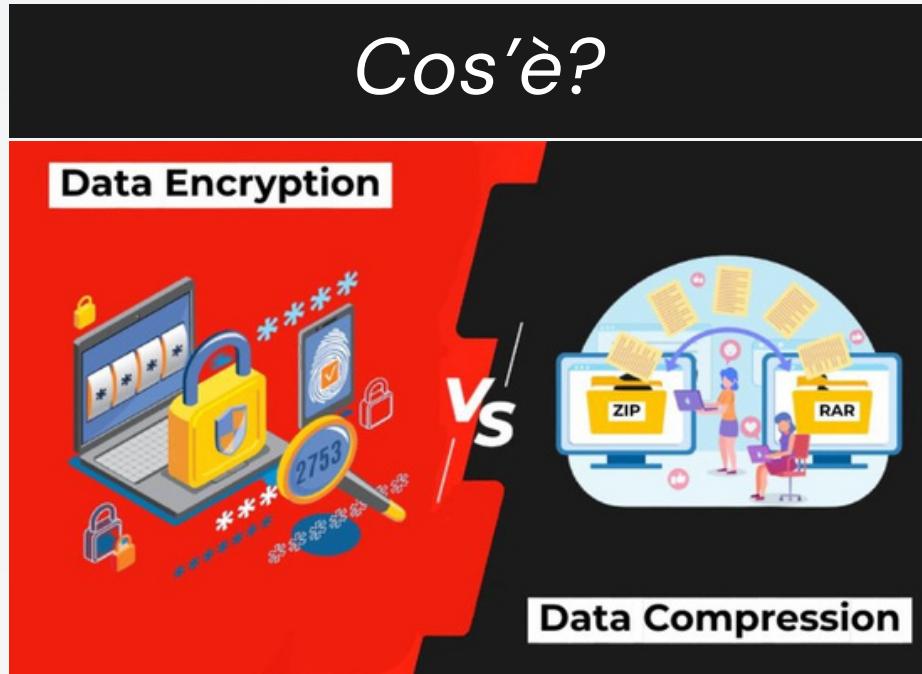
Aggiunta di caratteristiche crittografiche in una variante dell'Adaptive Arithmetic Coding (AAC).

Paper 3

Integrazione della trasformata di Burrows-Wheeler per garantire privacy in compressioni lossless.

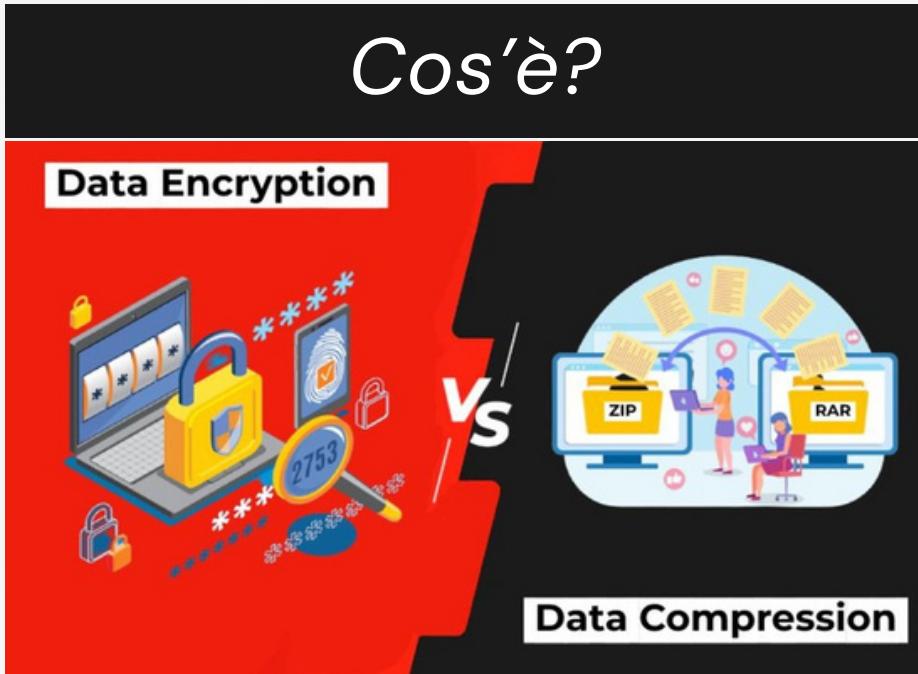


COMPRESSION CRYPTO-SYSTEM



Un Compression Crypto-System è una struttura che unisce compressione e cifratura in un unico processo, allo scopo di ridurre lo spazio di archiviazione e garantire la sicurezza dei dati durante la trasmissione o l'archiviazione. Questo sistema si basa sull'applicazione di algoritmi di compressione che, tramite l'uso di chiavi segrete, rendono i dati indecifrabili senza autorizzazione.

COMPRESSION CRYPTO-SYSTEM



Un Compression Crypto-System è una struttura che unisce compressione e cifratura in un unico processo, allo scopo di ridurre lo spazio di archiviazione e garantire la sicurezza dei dati durante la trasmissione o l'archiviazione. Questo sistema si basa sull'applicazione di algoritmi di compressione che, tramite l'uso di chiavi segrete, rendono i dati indecifrabili senza autorizzazione.

Tale approccio trova impiego in settori dove è essenziale proteggere grandi quantità di informazioni sensibili, come i sistemi di comunicazione e le reti aziendali. Combinando compressione e crittografia, i crypto-sistemi riescono a ottimizzare la velocità di trasmissione, risparmiare spazio e offrire protezione contro attacchi cibernetici.

A cosa servono?



LA CRYPTO COMPRESSION

La Crypto Compression è una tecnologia che combina simultaneamente la compressione dei dati e la crittografia. L'obiettivo è ridurre la dimensione dei file e, allo stesso tempo, proteggerli da accessi non autorizzati. Questo processo consente di ottenere dati compressi crittografati senza dover applicare separatamente le due operazioni.

LA CRYPTO COMPRESSION

La Crypto Compression è una tecnologia che combina simultaneamente la compressione dei dati e la crittografia. L'obiettivo è ridurre la dimensione dei file e, allo stesso tempo, proteggerli da accessi non autorizzati. Questo processo consente di ottenere dati compressi crittografati senza dover applicare separatamente le due operazioni.

Vantaggi

- Efficienza di spazio e sicurezza.
- Velocità rispetto a "Compress then Encrypt".
- Riduzione rischi di attacchi.
- Integrità dei dati.

Svantaggi

- Complessità Computazionale.
- Riduzione dell'efficienza di compressione.
- Non adottata universalmente.

LA CRYPTO COMPRESSION

Limiti e Sfide

Vulnerabilità agli Attacchi CPA (Chosen Plaintext Attack): Alcune tecniche di crypto compression possono essere vulnerabili agli attacchi su testo scelto (CPA), richiedendo l'uso di meccanismi avanzati per migliorare la sicurezza.

LA CRYPTO COMPRESSION

Limiti e Sfide

Vulnerabilità agli Attacchi CPA (Chosen Plaintext Attack): Alcune tecniche di crypto compression possono essere vulnerabili agli attacchi su testo scelto (CPA), richiedendo l'uso di meccanismi avanzati per migliorare la sicurezza.

Compressione su Testi Crittografati: Non è possibile applicare la compressione dopo la crittografia tradizionale, poiché i dati crittografati sono simili a dati casuali. Pertanto, è necessario integrare la crittografia nel processo di compressione.

LA CRYPTO COMPRESSION

Limiti e Sfide

Vulnerabilità agli Attacchi CPA (Chosen Plaintext Attack): Alcune tecniche di crypto compression possono essere vulnerabili agli attacchi su testo scelto (CPA), richiedendo l'uso di meccanismi avanzati per migliorare la sicurezza.

Compressione su Testi Crittografati: Non è possibile applicare la compressione dopo la crittografia tradizionale, poiché i dati crittografati sono simili a dati casuali. Pertanto, è necessario integrare la crittografia nel processo di compressione.

Implementazione Complessa: Integrare perfettamente compressione e crittografia in modo da mantenere l'efficienza di entrambe richiede algoritmi avanzati, come il Crypto-Huffman e l'Arithmetic Coding, che richiedono competenze specializzate per la configurazione e la manutenzione.



A HUFFMAN CODE BASED CRYPTO-SYSTEM

Yoav Gross, Shmul T. Klein, Elina Opalinsky,
Rivka Revivo, Dana Shapira

(Data Compression Conference (DCC), 2022)

I IDEA DI BASE

SCELTO UN NODO V DELL'ALBERO DI HUFFMAN SI TRASFORMA IL SOTTOALBERO AD ESSO
ASSOCIATO



IDEA DI BASE

SCELTO UN NODO V DELL'ALBERO DI HUFFMAN SI TRASFORMA IL SOTTOALBERO AD ESSO ASSOCIATO

VANTAGGIO

È possibile generare output unici e sicuri senza alterare la lunghezza delle codeword.

Canva



IDEA DI BASE

SCELTO UN NODO **V** DELL'ALBERO DI HUFFMAN SI TRASFORMA IL SOTTOALBERO AD ESSO ASSOCIATO

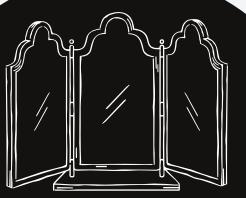
VANTAGGIO

È possibile generare output unici e sicuri senza alterare la lunghezza delle codeword.

SVANTAGGIO

Una volta identificato il nodo **V** scelto, è possibile ricostruire l'albero originale.

APPLICAZIONI

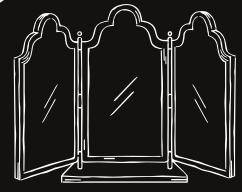


**Il sottoalbero radicato in
v viene sostituito dalla
sua immagine speculare**

**Il codeword ab viene
trasformata in ab^-**

**TRASFORMAZIONE A
SPECCHIO**

APPLICAZIONI



Il sottoalbero radicato in
 v viene sostituito dalla
sua immagine speculare

Il codeword ab viene
trasformata in ab^-

TRASFORMAZIONE A
SPECCHIO

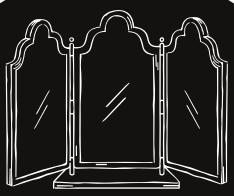


Si scambiano i sottoalberi
sinistro e destro radicati
in v

Il codeword $ab\gamma$ viene
trasformato in $ab\gamma^-$

TRASFORMAZIONE A
SWAP

APPLICAZIONI



Il sottoalbero radicato in v viene sostituito dalla sua immagine speculare

Il codeword ab viene trasformata in ab^-

TRASFORMAZIONE A SPECCHIO



Si scambiano i sottoalberi sinistro e destro radicati in v

Il codeword aby viene trasformato in aby^-

TRASFORMAZIONE A SWAP

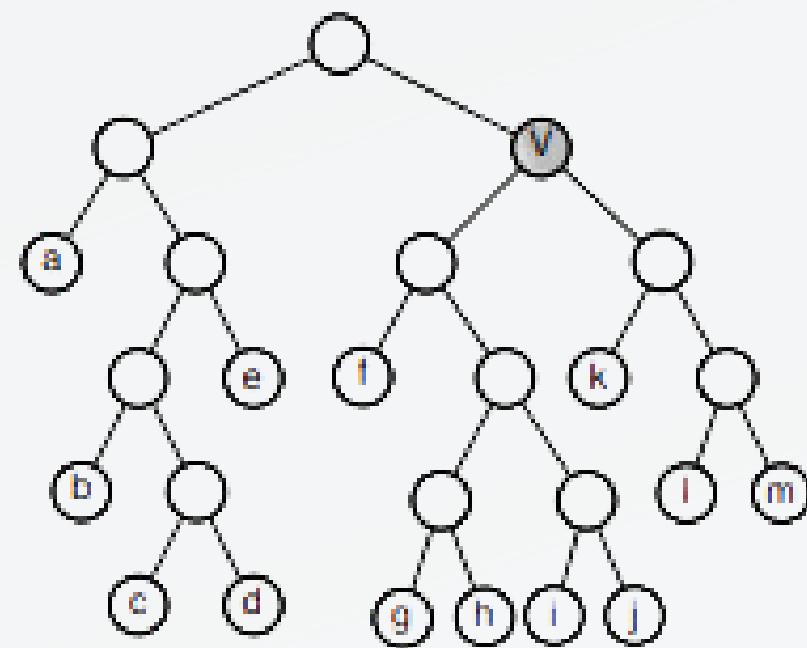


Si scambiano i sottoalberi dei nodi v e $\text{right}(v)$

I codeword $\alpha_1\beta$ e $\alpha_2\gamma$ diventano rispettivamente $\alpha_1\gamma$ e $\alpha_2\beta$

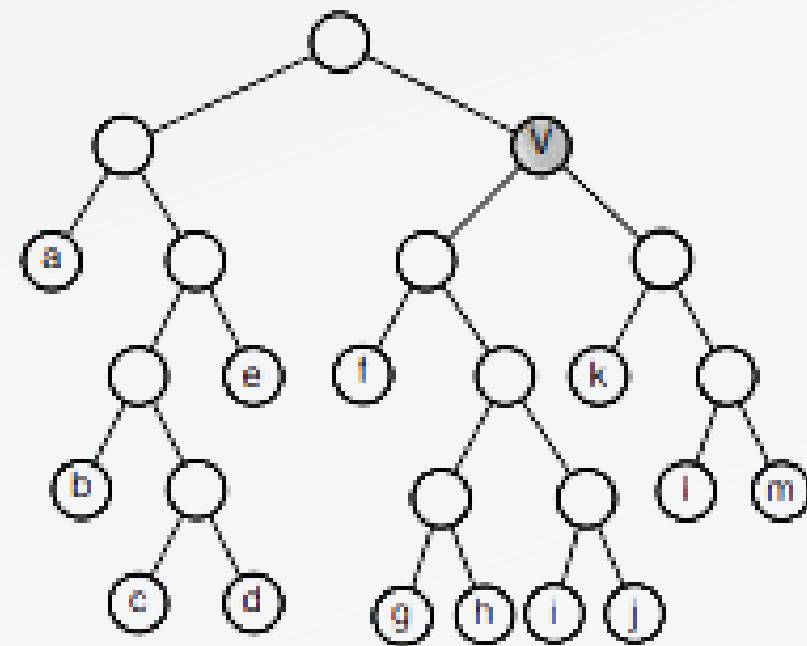
TRASFORMAZIONE DI SCAMBIO LIVELLO

APPLICAZIONI (2)

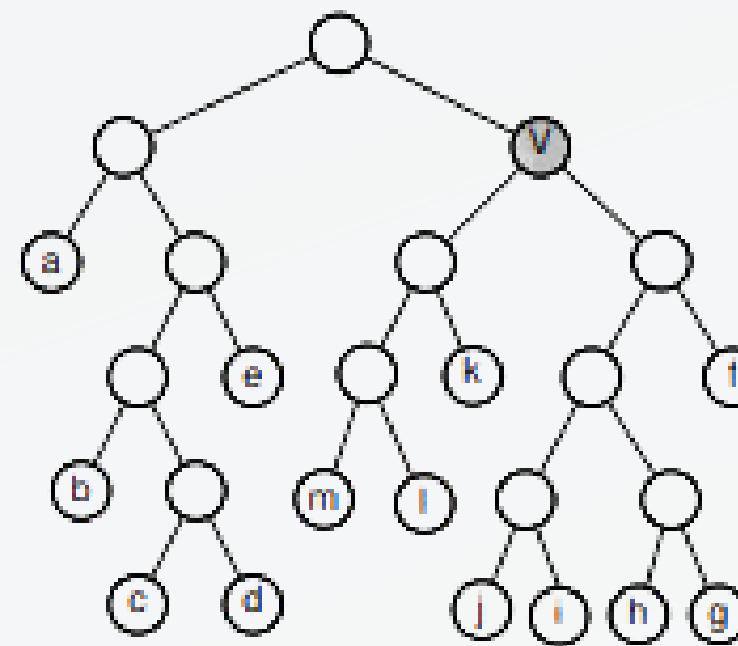


(a) *Original*.

APPLICAZIONI (2)

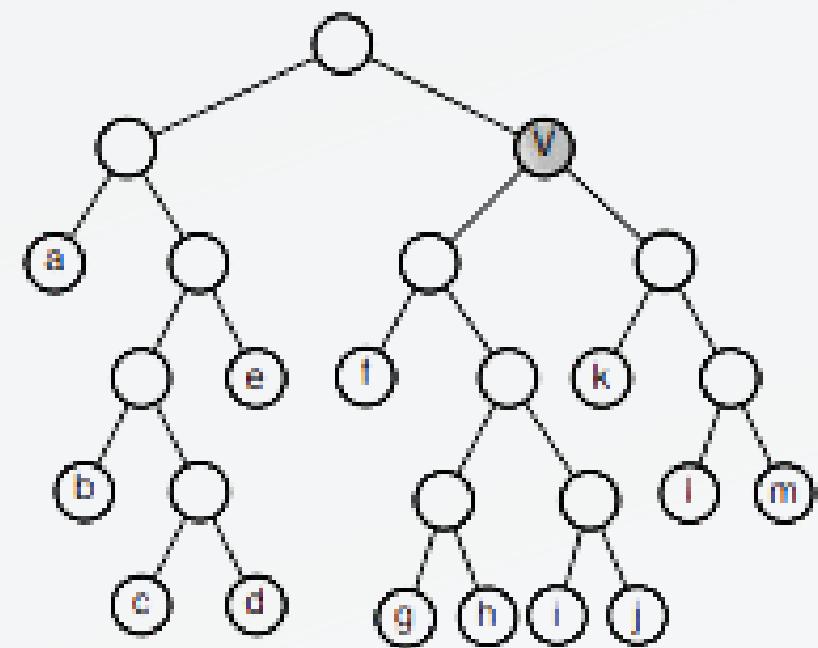


(a) *Original.*

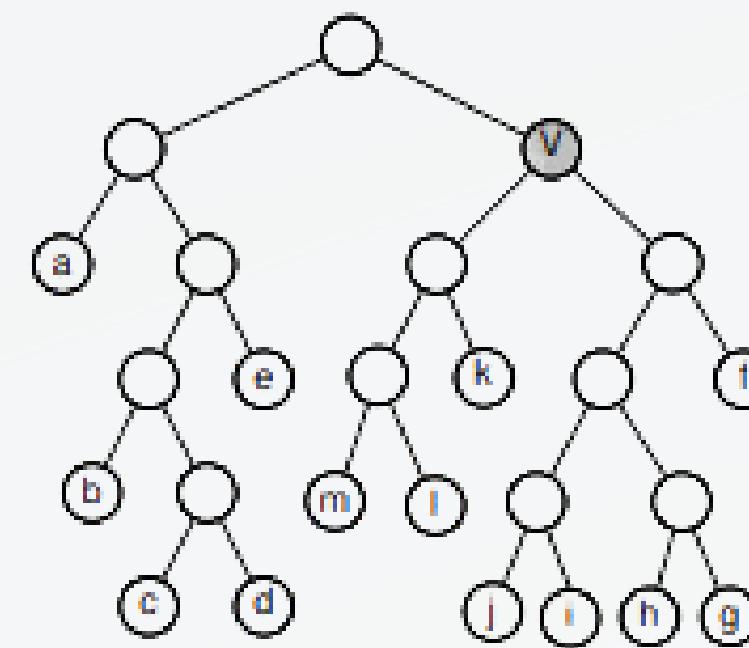


(b) *Mirror.*

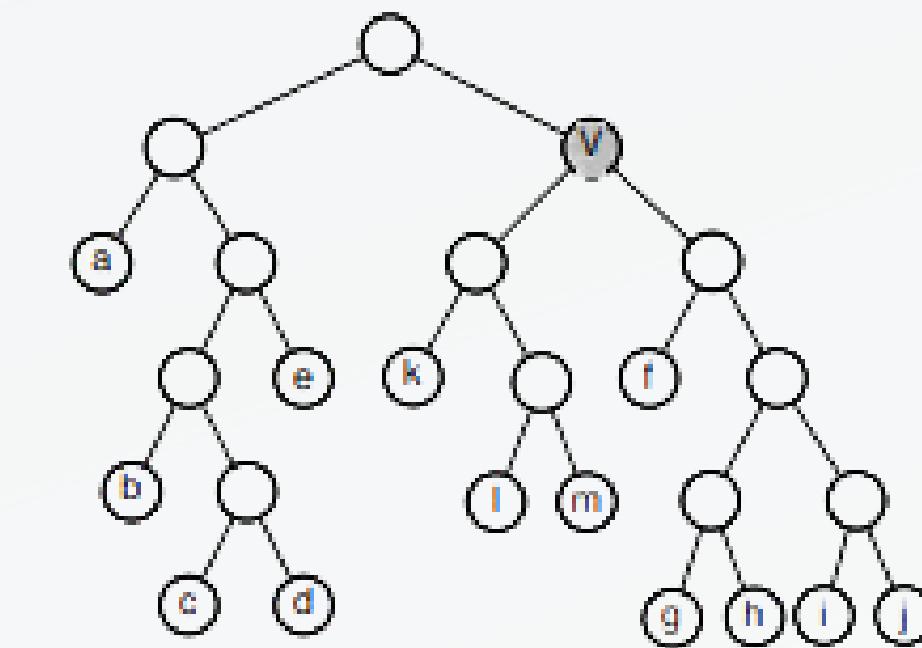
APPLICAZIONI (2)



(a) Original.



(b) Mirror



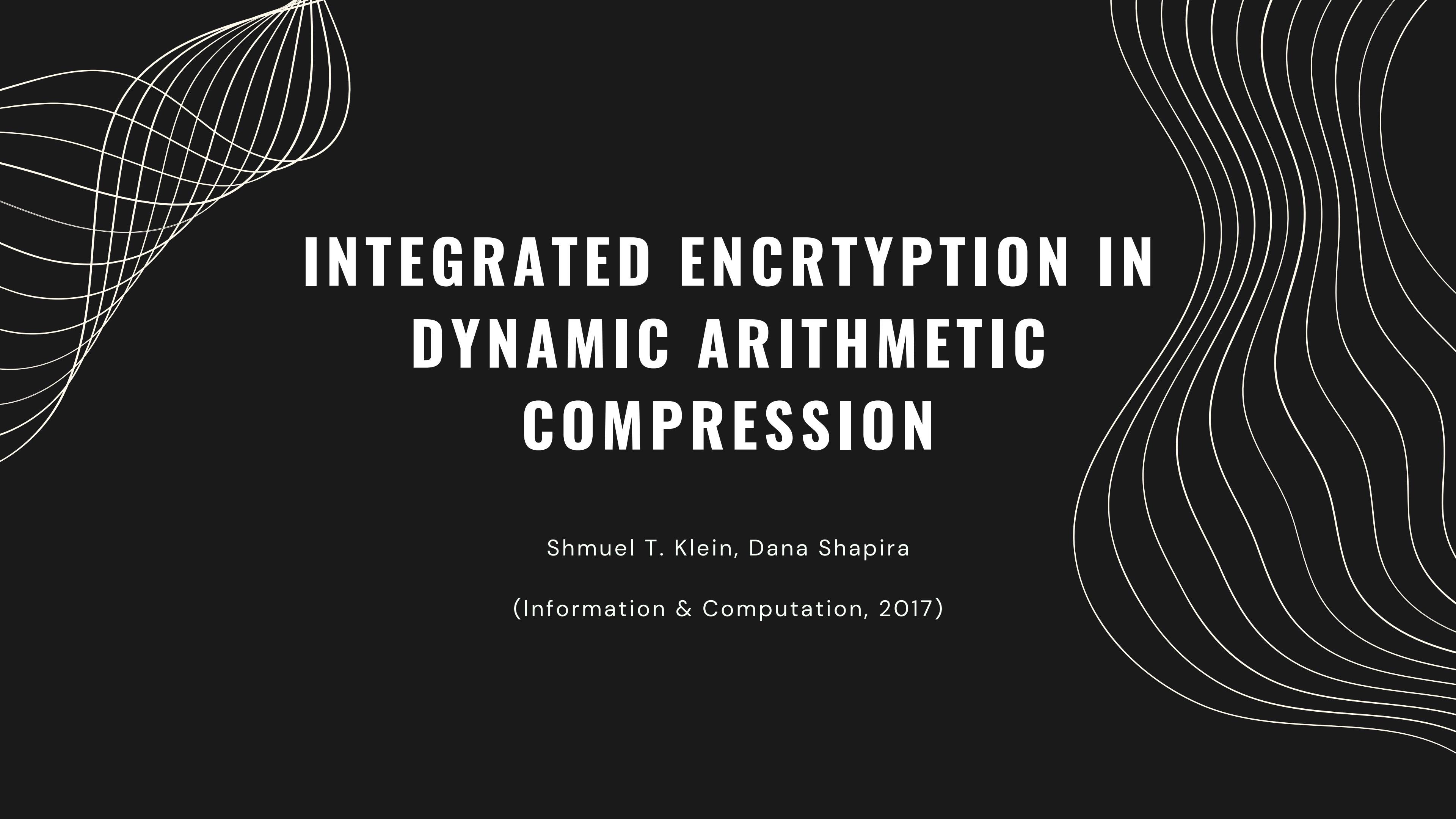
(c) Swap.

APPLICAZIONI (3)

Dato un testo $T = x_1 x_2 \cdots x_n$ da comprimere e cifrare, dove x_i sono caratteri appartenenti a un alfabeto Σ di dimensione σ , viene scelto un intero $1 \leq k < \sigma$ secondo il quale si effettua la trasformazione dei sottoalberi interni

Dopo la lettura di un carattere, k nodi interni differenti sono selezionati in base a una chiave segreta e su ciascuno viene applicata una trasformazione sul sottoalbero con radice in v_i (per $1 \leq i \leq k$)

La scelta di k permette di controllare il trade-off tra sicurezza e complessità computazionale



INTEGRATED ENCRYPTION IN DYNAMIC ARITHMETIC COMPRESSION

Shmuel T. Klein, Dana Shapira

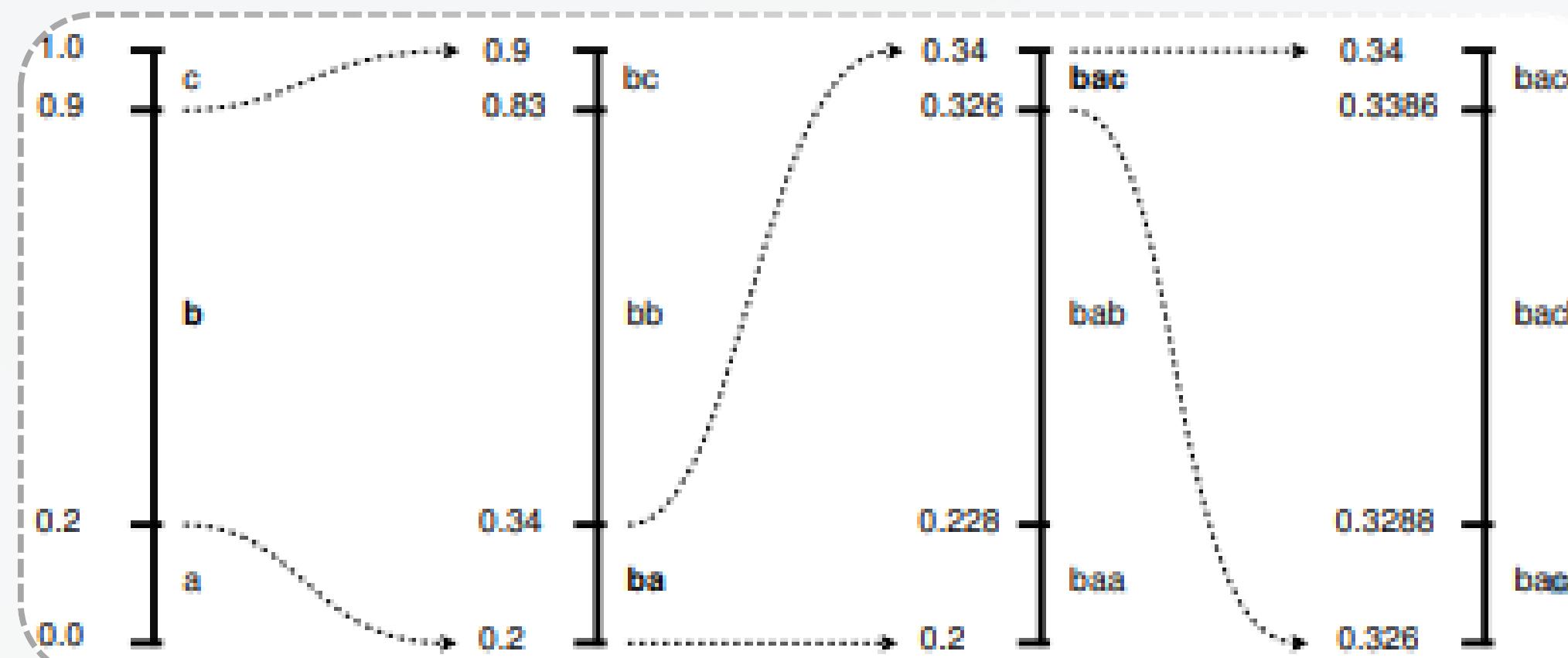
(Information & Computation, 2017)

ARITHMETIC CODING

Rappresentare una sequenza di simboli restringendoli ad un sottointervallo $[low, high) \rightarrow [0, 1]$.

Per risparmiare spazio, invece di trasmettere l'intero intervallo risultante (con i valori di estremità "low" e "high"), è sufficiente inviare un solo numero reale che appartiene a quell'intervallo finale.

$$\Sigma = \{a, b, c\} \text{ e } P(x) = 0.2 - 0.7 - 0.1$$



VARIANTE DINAMICA

Nella variante dinamica, le probabilità dei simboli non sono fisse, ma vengono aggiornate durante il processo di codifica, adattando la probabilità degli stessi ai simboli incontrati sino a quel punto.

IN PARTICOLARE:

- **SI CALCOLA COSTANTEMENTE IL NUOVO INTERVALLO**
- **SI AGGIORNA LA DISTRIBUZIONE CORRENTE**

IDEA DI BASE

Invece di aggiornare il modello ad ogni simbolo codificato, gli aggiornamenti vengono effettuati solo in determinati passi.

Se infatti vediamo la chiave come una sequenza di bit

$$k_0, k_1, k_2 \dots, k_{r-1}$$

l'aggiornamento dovrà essere effettuato solo se:

**AD OGNI PASSO i , IL BIT $k_{(i-1) \bmod r}$
DELLA CHIAVE È 1
IL MODELLO È VISTO COME
SOTTOINTERVALLI CONTIGUI.**

```
encode( $T, K$ )
1  $[low, high] \leftarrow [0, 1]$ 
2 initialize the interval partition distribution  $\{[\ell_0, h_0], \dots, [\ell_{s-1}, h_{s-1}]\}$ 
3 for  $i \leftarrow 1$  to  $n$ 
4.1  $range \leftarrow high - low$ 
4.2  $high \leftarrow low + range \cdot h_{t_i}$ 
4.3  $low \leftarrow low + range \cdot \ell_{t_i}$ 
4.4 if  $k_{(i-1) \bmod r} = 1$  then
4.4.1 update  $\{[\ell_0, h_0], \dots, [\ell_{s-1}, h_{s-1}]\}$ 
4 return some value in the current interval
```

POSSIBILI ATTACCHI

Attacco 1: Indovinare la chiave

Se fosse lunga quanto il messaggio -> OTP (One Time Pad) nella pratica, non può essere così.

k non sarà quindi una chiave, bensì un seme che scegliamo (un grande primo P e una radice a mod P):

$$K \leftarrow a^K \bmod P.$$

In questo modo sfruttiamo il logaritmo discreto.

POSSIBILI ATTACCHI

Attacco 2: Ricostruzione Testo CPA

Il punto debole è l'inizializzazione, nota, e un attaccante potrebbe sfruttarla per osservare come i segmenti di codifica cambino in funzione della chiave.

Per risolvere questo problema, possiamo aggiungere un testo noto all'inizio, al fine di minimizzare le possibilita' di decodifica e indurre un "Effetto valanga".

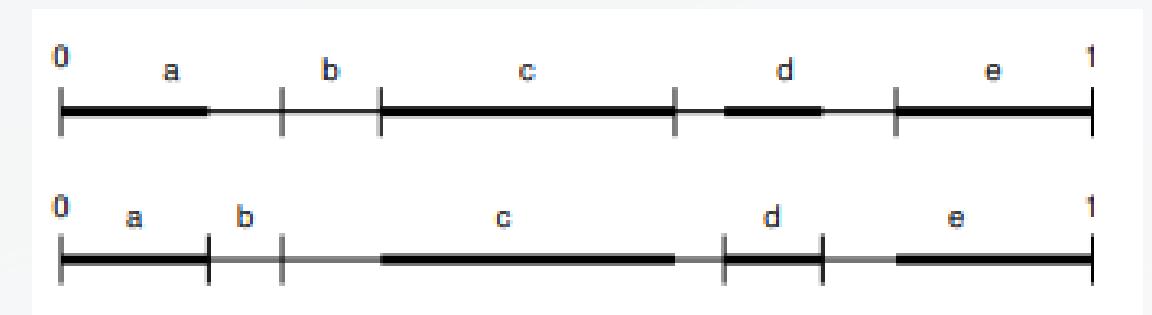
POSSIBILI ATTACCHI

Attacco 3: Indovinare i limiti degli intervalli

Un avversario potrebbe cercare di prevedere come i sotto-intervalli cambino nel tempo, se conosce le distribuzioni dei caratteri.

Sebbene la sovrapposizione sia altissima (83% dopo 1000 caratteri), al crescere dei caratteri codificati la probabilità d'errore aumenta a sua volta.

Dopo 1000 caratteri codificati, l'avversario ha una probabilità d'errore del 70% per ogni carattere.



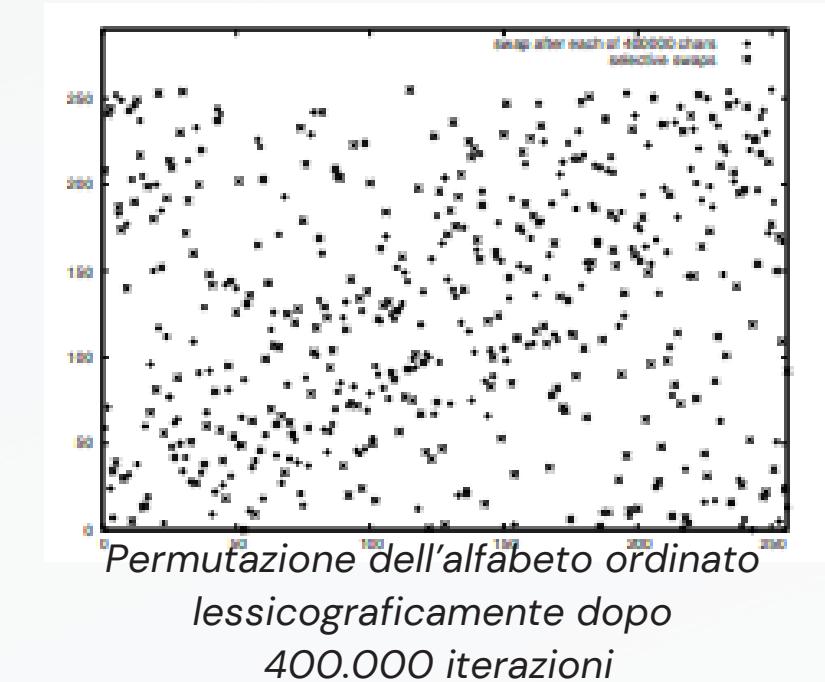
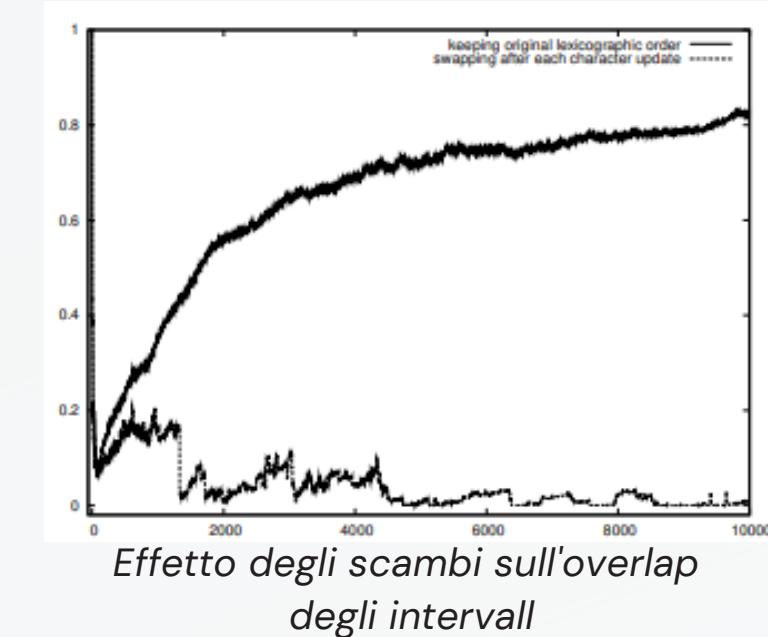
Sovrapposizione tra un modello che aggiorna il partizionamento dopo ogni carattere letto, e uno che lo fa in base ad un segreto K

POSSIBILI ATTACCHI

Attacco 4: Indovinare ordine dei caratteri

Mentre nella codifica di Huffman i caratteri sono ordinati in base alla loro frequenza, nella codifica aritmetica l'ordine dei caratteri nell'intervallo può essere scelto liberamente (spesso, lessicografico).

Idea: permutare i caratteri selettivamente, sfruttando la chiave k. Ogni carattere si scambia, quindi, col suo vicino nell'ordine corrente.



IN SINTESI

Quanto ci convince davvero questa metodologia di
cifratura?

Perdita di prestazioni
presente, ma
considerabile
NULLA.

PERFORMANCES

IN SINTESI

Quanto ci convince davvero questa metodologia di cifratura?

Perdita di prestazioni
presente, ma
considerabile
NULLA.

PERFORMANCES

Le distribuzioni
ottenute utilizzando
questa codifica sono
vicinissime a quella
UNIFORME.

UNIFORMITÀ



IN SINTESI

Quanto ci convince davvero questa metodologia di cifratura?

Perdita di prestazioni
presente, ma
considerabile
NULLA.

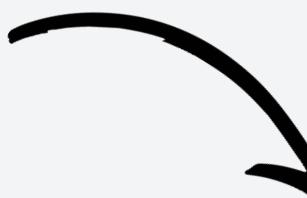
PERFORMANCES

Le distribuzioni
ottenute utilizzando
questa codifica sono
vicinissime a quella
UNIFORME.

UNIFORMITÀ

Il risultato finale
risente
ENORMEMENTE
delle minime
differenze di chiave.

SENSIBILITÀ



ON SCRAMBLING THE BURROWS- WHEELER TRANSFORM TO PROVIDE PRIVACY IN LOSSLESS COMPRESSION

M. Oğuzhan Külekci

(Computer & Security, 2012)

INTRODUZIONE

Contesto e problema



- La compressione riduce spazio e banda.
- Ampiamente usata in web, database, file multimediali.
- Non garantisce la sicurezza dei dati.

**COMPRESSEIONE DATI E
NECESSITÀ DI SICUREZZA**

INTRODUZIONE

Contesto e problema



- La compressione riduce spazio e banda.
- Ampiamente usata in web, database, file multimediali.
- Non garantisce la sicurezza dei dati.

**COMPRESSEIONE DATI E
NECESSITÀ DI SICUREZZA**



- I dati compressi possono essere vulnerabili.
- Mancanza di crittografia intrinseca dei metodi standard.
- Critico per dati sensibili o riservati

**LIMITI DELLE TECNICHE DI
COMPRESSEIONE STANDARD**

INTRODUZIONE

Contesto e problema



- La compressione riduce spazio e banda.
- Ampiamente usata in web, database, file multimediali.
- Non garantisce la sicurezza dei dati.

**COMPRESSEIONE DATI E
NECESSITÀ DI SICUREZZA**



- I dati compressi possono essere vulnerabili.
- Mancanza di crittografia intrinseca dei metodi standard.
- Critico per dati sensibili o riservati

**LIMITI DELLE TECNICHE DI
COMPRESSEIONE STANDARD**



- Diffuso ma inefficiente.
- Svantaggi: carico computazionale, ricerche lente.
- Serve una soluzione unificata tra compressione e sicurezza.

COMPRESS-THEN-ENCRYPT

VARIANTE DELLA TRASFORMATA DI BURROWS-WHEELER (BWT)

Soluzione proposta: Scrambled Burrows- Wheeler Transform (sBWT)

Aspetto	BWT TRADIZIONALE	sBWT
Ordinamento	Lessicografico fisso	Ordine casuale e segreto
Sicurezza	Nessuna sicurezza	Basata sull'ordine casuale dei caratteri
Operazioni	Solo compressione	Compressione e sicurezza
Ricerche	Necessaria la decompressione	Ricerche dirette senza decompressione

PROCESSO DI GENERAZIONE DELLA BURROWS-WHEELER TRANSFORM (BWT)

$T=mela\$$

BWT	ROTAZIONI CICLICHE
1	mela\$
2	ela\$m
3	la\$me
4	a\$mel
5	\$mela

PROCESSO DI GENERAZIONE DELLA BURROWS-WHEELER TRANSFORM (BWT)

$T=mela\$$

BWT	ROTAZIONI CICLICHE
1	mela\$
2	ela\$m
3	la\$me
4	a\$mel
5	\$mela

BWT	ORDINAMENTO LESSICOGRAFICO
1	\$mela
2	a\$mel
3	ela\$m
4	la\$me
5	mela\$

PROCESSO DI GENERAZIONE DELLA BURROWS-WHEELER TRANSFORM (BWT)

$T=mela\$$

BWT	ROTAZIONI CICLICHE
1	mela\$
2	ela\$m
3	la\$me
4	a\$mel
5	\$mela

BWT	ORDINAMENTO LESSICOGRAFICO
1	\$mela
2	a\$mel
3	ela\$m
4	la\$me
5	mela\$

BWT	ESTRAZIONE COLONNA FINALE
1	\$mela
2	a\$mel
3	ela\$m
4	la\$me
5	mela\$

OUTPUT: ALMES\$

PROCESSO DI GENERAZIONE DELLA SCRABLED BURROWS-WHEELER TRANSFORM (SBWT)

$T=mela\$$

ordine segreto : $(\$ < l < m < a < e)$

sBWT	ROTAZIONI CICLICHE
1	mela\$
2	ela\$m
3	la\$me
4	a\$mel
5	\$mela

PROCESSO DI GENERAZIONE DELLA SCRABLED BURROWS-WHEELER TRANSFORM (SBWT)

$T=mela\$$

ordine segreto : $(\$ < l < m < a < e)$

sBWT	ROTAZIONI CICLICHE
1	mela\$
2	ela\$m
3	la\$me
4	a\$mel
5	\$mela

sBWT	ORDINAMENTO SEGRETO
1	\$mela
2	la\$me
3	mela\$
4	a\$mel
5	ela\$m

PROCESSO DI GENERAZIONE DELLA SCRABLED BURROWS-WHEELER TRANSFORM (SBWT)

$T=mela\$$

ordine segreto : $(\$ < l < m < a < e)$

sBWT	ROTAZIONI CICLICHE
1	mela\$
2	ela\$m
3	la\$me
4	a\$mel
5	\$mela

sBWT	ORDINAMENTO SEGRETO
1	\$mela
2	la\$me
3	mela\$
4	a\$mel
5	ela\$m

sBWT	ESTRAZIONE COLONNA FINALE
1	\$mela
2	la\$me
3	mela\$
4	a\$mel
5	ela\$m

OUTPUT: AE\$LM

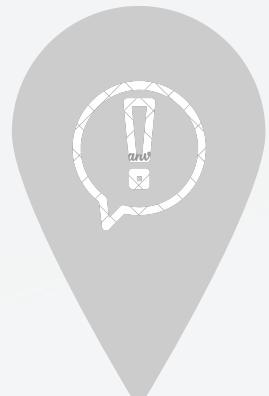
Limiti, vantaggi e applicazioni future



LIMITI DELLA SBWT

- **Sicurezza pratica, non provata:** Meno sicura rispetto alle cifrature provate, adatta a contesti con sicurezza moderata.
- **Dipendenza dall'Ordine segreto:** vulnerabile se l'ordine casuale viene compromesso

Limiti, vantaggi e applicazioni future



LIMITI DELLA SBWT

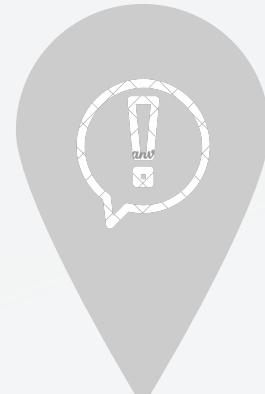
- **Sicurezza pratica, non provata:** Meno sicura rispetto alle cifrature provate, adatta a contesti con sicurezza moderata.
- **Dipendenza dall'Ordine segreto:** vulnerabile se l'ordine casuale viene compromesso



VANTAGGI

- **Compressione e sicurezza:** Unisce compressione e sicurezza in un singolo passaggio, senza cifratura aggiuntiva.
- **Efficienza nelle ricerche:** Consente ricerche dirette sui dati compressi, senza decompressione.
- **Adattabilità:** Applicabile a scenari con esigenze moderate di sicurezza e ricerca

Limiti, vantaggi e applicazioni future



LIMITI DELLA SBWT

- **Sicurezza pratica, non provata:** Meno sicura rispetto alle cifrature provate, adatta a contesti con sicurezza moderata.
- **Dipendenza dall'Ordine segreto:** vulnerabile se l'ordine casuale viene compromesso



VANTAGGI

- **Compressione e sicurezza:** Unisce compressione e sicurezza in un singolo passaggio, senza cifratura aggiuntiva.
- **Efficienza nelle ricerche:** Consente ricerche dirette sui dati compressi, senza decompressione.
- **Adattabilità:** Applicabile a scenari con esigenze moderate di sicurezza e ricerca



APPLICAZIONI

- **Indicizzazione Compressa:** Ricerche veloci su dati compressi, ad esempio in indici FM.
- **Archiviazione Sicura:** Compressione e protezione di backup e dati sensibili.
- **Dispositivi IoT:** Alternativa leggera alla cifratura completa, per dispositivi a risorse limitate
- **Data Mining:** Utile per l'analisi e ricerche su grandi volumi di dati compressi.

OUR TEAM



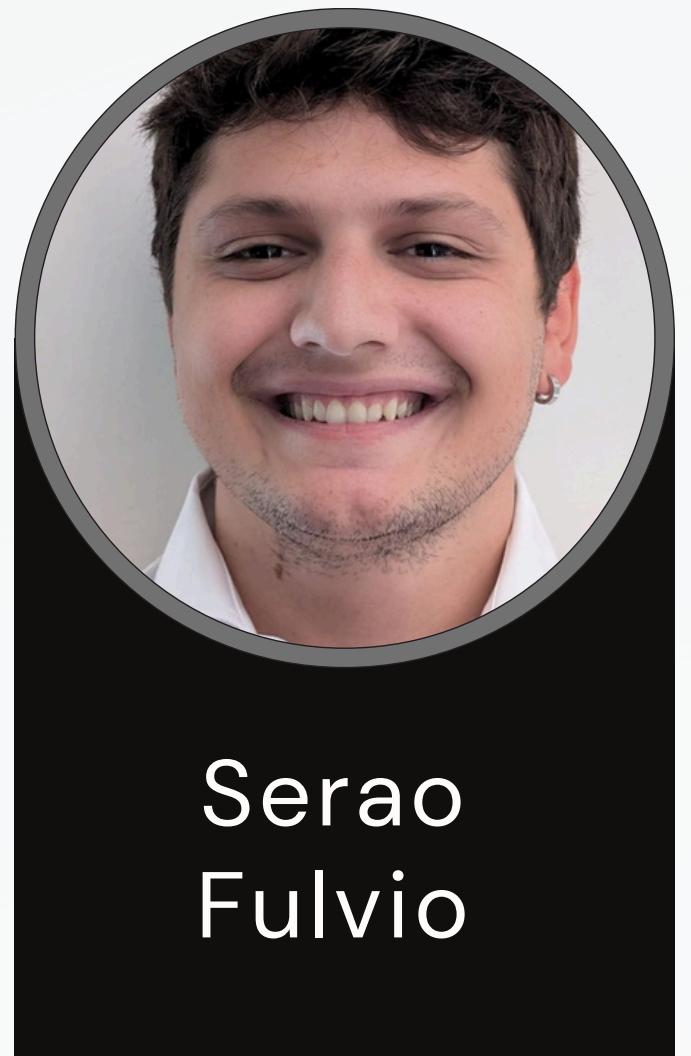
Fusco
Marco



Garofalo
Antonio



Palmisciano
Marco



Serao
Fulvio

**GRAZIE PER
L'ATTENZIONE**

