POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea

Format Preserving Encryption for databases

Relatori
prof. Antonio Lioy
prof. Andrea Atzeni

Francesco Vaccaro

Supervisore Aziendale
ing. Marco Mangiulli, ing. Luca Castello

MESE 20**

A mio padre

† A mio nonno Pino

Summary

Inserire qui un breve sommario della tesi. (INSERITO QUELLO DELLA TRACCIA)

Companies migration to the cloud implies protection of their sensitive and private data. Encryption is the key tool for business's confidential data protection against cyber security threats. However, storing the data in an encrypted format requires to address critical issues: performance, format and ordering of data, protection of encryption keys. Format-preserving encryption (FPE) allows to encrypt the data in such a way that the output (the ciphertext) is in the same format as the input (the plaintext).

Acknowledgements

Opzionali, solo nel caso si sia ricevuto un aiuto speciale e particolarmente rilevante.

Contents

Chapter 1

State of the art

1.1 Sezione 1 - Modes of operation

A block cipher mode of operation is an algorithm that is used in the scope of a specific symmetric key block cipher algorithm in order to provide an information service, such as confidentiality or authentication.

The reason behind the definition of a block cipher mode of operation comes from the need to have a working block cipher even if the input data of the block cipher is different from the algortihm's block size.

The very first modes of operation were published in FIPS PUB 81 [1] in 1980 by the National Institute of Standards and Technology (NIST). This publication included four modes of operation: ECB, CBC, OFB e CFB; all of these modes were originally suited for Data Encryption Standard (DES) block cipher, that was withdrawn in 2005.

After this initial publication, NIST starts considering proposals for new modes of operation. Proposals are evaluated by the NIST and when a mode of operation is approved it is published in the 800-38 series of Special Publication (SP 800-38). Currently NIST approved eigth confidentiality modes (ECB, CBC, OCB, CFB, CTR, XTS-AES, FF1 and FF3), one authentication mode (CMAC) and five combined modes for confidentiality and authentication (CCM, GCM, KW, KWP and TKW), for a total fourteen modes.

1.2 Sezione 2 - Format Preserving Encryption mode of operation

Methods for Format Preserving Encryption were published by NIST in the seventh part of the 800-38 series [2]. The modes for encryption defined in the previous six parts are all transformations on binary data, that is, the inputs and the outputs of the modes are bit strings. For sequences of non-binary symbols there is no natural way for these modes to produce encrypted data that has the same format.

A Format Preserving Encryption, given any finite set of symbols, transforms data that is formatted as a sequence of the symbols in such a way that the encrypted form of the data has the same format, including the length, as the original data. A typical example is a Social Security Number (SSN), that consists of nine decimal numbers; consequently The SSN is an integer less then one bilion (1,000,000,000). If we use a non-FPE mode to encrypt an SSN number, we have to convert it to a bit string as input for that mode; then we apply the mode and we obtain an output that is again a bit string. When the bit string is converted back to an integer, it can be the case that the integer will be greater than one bilion, which would be too long fo an SSN and will we break the format defined.

FPE is useful especially for data at rest in database applications, where changes to the length or format of data fields must not be supported. In fact a lot of companies, woriking in the fincance word, as well as, in the healtcare or government, have legacy applications (old-fashioned and expensive applications) requiring a certain format of data. In order to account for the new format the application should be redone from scratch, spending time and money. FPE allows a drop-in replacement of plaintext with the respective ciphertext in legacy applications.

Another advantage of FPE is that it helps in recognizing data encrypted. As an example we can take a credit card number (CCN), typically composed of 16 integer; the number obtained after encryption using FPE will be consist again of 16 integer, so in the contest of a database we will know that we are dealing with a CCN. This aspect is important if we have sensitive data (maybe protected by the GDPR legislation) and we want to perform some statistical researches on these data.

Furthermore FPE, in constrast with other modes of operation, such as the CBC mode, which uses a random seed value to initialize the encryption algorithm, gives the possibilty to use encrypted data as a unique key to identify a row in a database.

SCRIVERE QUALCOSA SU I COMMENTI PUBLICI????

1.3 Sezione 3 - FPE methods

The origins of the FPE problem go back in 1981, when the US National Bureau of Standards (which later became NIST) published FIPS PUB 74 [3], an appendix describing an approach for enciphering arbitrary strings over an arbitrary alphabet. Afterwards, in 1997, Brightwell and Smith were the first authors to describe more generally the FPE problem, calling it *"datatype-preserving encryption"*.

The most important study which increased the interest on FPE is the paper by cryptographers John Black and Phillip Rogaway, *"Cipher with Aribtrary Finite Domains"* [4]. The paper describes three different methods to implement FPE:

- Prefix Cipher;
- Cycle-Walking Cipher;
- Generalized-Feistel Cipher.

Black and Rogaway proved that each of these three methods is as secure as the block cipher used to construct them; thus, if the AES (CITARE) is used to create the FPE algorithm, an adversary can break the FPE algorithm if and only if he can break the AES algorithm too.

The *Prefix Cipher* method fixes some integer k and works on M, the set [0, k-1]. His goal is to build a cipher with domain M. It assigns a pseudorandom weight to each integer, then sort by weight. The weights are defined by applying an existing block cipher to each integer. This method is useful only for small values of k, because the cost in time and space due to the initialization step is $O(k)$, while generally enciphering and deciphering are constant-time operations. The ciphering and deciphering algorithms are given in Figure 1.

```
Prova. Inserire algoritmo di Prefix Cipher.
```

Figure 1.1. Esempio di programma inserito tramite `lstlisting`.

The *Cycle-Walking Cipher* method uses a block cipher whose domain is larger than M, where the points out-of-range are handled by repeatedly applying the block cipher until the result is

within M. More precisely let N be the smallest power of 2 larger or equal to k and n be *log(N)*, the underlying cipher works on blocks of n-bit. The recursion is guaranteed to terminate, because the block cipher is supposed to be ideal, which is in fact a random permutation. If we apply the block cipher enough times we must eventually arrive back at some point in M, even at the initial point itself. This method is quite fisible if k is just smaller than some power of 2, because in this case the number of points we have to traverse during any encipherment is correspondegly small. Instead, in the worst case scenario where k is one larger than a power of 2, the algorithm might require k calls to the underlying block cipher to encipher just one point. There is also another drawback: if the block cipher is of a fixed size, such as AES, this is a severe restriction on the sizes of M for which this method is practical. The ciphering and deciphering algorithms are given in Figure 2.

The *Generalized-Feistel Cipher* method consists in decomposing all the numbers in M into pairs of "similarly sized" numbers and then apply the well-known Feistel (CITARE) construction to produce a cipher. The cipher takes as input r, the number of round used in the Feistel network and two positive numbers a and b such that $ab \geq k$. Whitin the network r random function *F1,...,Fr* are used. This method is an adaptation of Luby-Rackoff construction (CITARE) (with the related security proof) and it shows that when the attacker is limited to access less than $Q = 2minL,R/2$ (RISCRIVERE BENE) plaintext/ciphertext pairs, she has not enough information to distinguish this construction from a random permutation with domain M. The Generalized-Feistel Cipher can be quite efficient, even if the proven bounds are weak when the message space M is small. The ciphering and deciphering algorithms are given in Figure 3.

1.4 Sezione unknown - FPE solutions adopted by Cloud Service Providers

Voltage SecureData is an industry leader in the data security space, where hundreds of enterprises rely on it to secure sensitive data at the application layer and establish the trust of their customers

Chapter 2

Analysis of the approved FPE mode

2.1 Sezione 1 - SP 800-38G

2.2 Spiegare cos'è un 'tweak', (tweakable block cipher paper)

2.3 Sezione 2 - FF1

FF1

2.4 Sezione 3 - FF3

FF3

2.5 Sezione 4 - FF2

FF2

Chapter 3

Proof of concept

3.1 Sezione 1

Scrivere come è stata implementata la Proof of Concept

3.2 Sezione 2

Risultati Proof of Concept

3.3 English Section

Do we want to write the thesis in English?

Chapter 4

Risultati

Chapter 5

Conclusioni

Bibliography

[1] FIPS PUB 81, https://csrc.nist.gov/csrc/media/publications/fips/81/archive/1980-12-02/documents/fips81.pdf

[2] SP 800-38G, DOI http://dx.doi.org/10.6028/NIST.SP.800-38G

[3] FIPS PUB 74, https://nvlpubs.nist.gov/nistpubs/Legacy/FIPS/fipspub74.pdf

[4] J. Black, P. Rogaway, "Ciphers with Arbitrary Finite Domains", https://www.cs.ucdavis.edu/~rogaway/papers/subset.pdf