

به نام خدا
گزارش کار پروژه درس پایگاه داده
CryptDB موضوع پروژه
استاد راهنما: دکتر ایزدی

شایان فاضلی ۹۱۱۰۲۱۷۱
فؤاد جعفری نژاد ۹۳۱۰۰۷۸۵
سپیده برنگی ۹۰۱۱۰۷۳۷

تابستان ۱۳۹۵

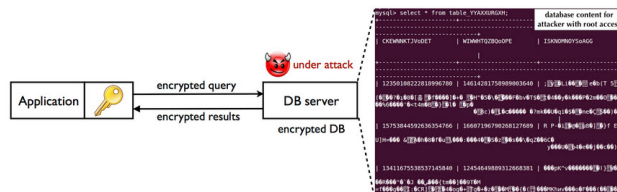
فهرست مطالب

۳	۱ آشنایی با CryptDB
۳	۲ دلایل استفاده از CryptDB
۳	۳ رمزنگاری پیازی
۳	۱.۳ لایه های پیازی
۴	۱.۱.۳ دستور GroupBy
۴	۲.۱.۳ دستور sum
۴	۴ شیوه های رمزنگاری
۴	۱.۴ RND
۴	۲.۴ HOM
۴	۳.۴ SEARCH
۵	۴.۴ DET
۵	۵.۴ OPE
۵	۶.۴ OPE-JOIN ، JOIN
۵	۵ اجرای CryptDB
۶	۶ موارد استفاده CryptDB
۷	۷ Query ها در CryptDB
۷	۸ نتایج تست CryptDB در BenchMark ها
۸	۹ JDBC
۸	۱۰ about MySQL
۸	۱۱ موارد مورد نیاز

۸	۱۲ چگونه نصب در Linux
۹	۱۳ MySQL در java
۱۰	۱۴ نحوه اتصال به MySQL
۱۰	۱۵ برقراری ارتباط با DataBase
۱۰	۱۶ نحوه فرستادن Query
۱۱	۱۷ دریافت result
۱۱	۱۸ گرفتن خطاها

۱ آشنایی با CryptDB

وقتی یک client از یک دیتابیس استفاده می کند به ازای query هایی که به سیستم می دهد شخصی می تواند بین System و client قرار بگیرد و تمام query ها جواب های آن ها را نگه دارد اما وقتی یک Client از cryptDB استفاده می کند وقتی query را می فرستد query رمز می شود و می رود و هنگامی که جواب آن query می آید جواب نیز رمز می شود و می آید و حال با این اوصاف اگر شخصی در وسط قرار بگیرد از داده هایی که می بیند هیچ چیز متوجه نمی شود.



شکل ۱: شکل مربوط به ارتباط پایگاه داده رمز نگاری شده

این پایگاه داده اولین بار توسط دانشگاه MIT پیاده سازی شده است. (<http://css.csail.mit.edu/cryptdb>) چند نکته در مورد پایگاه داده رمز نگاری شده:

1. Provides practical and provable confidentiality in many ways.
2. It work by executing SQL queries over encrypted data.
3. It uses a collection of SQL-aware encryption schemes.
4. It chains encryption keys to user passwords.

۲ دلایل استفاده از CryptDB

دلایل استفاده زیادی دارد: (دو مورد از آن در زیر توضیح داده می شود)

۱. در سیستم دیتابیس administrator ممکن است بخواهد که دیتا را بگیرد یا نشر دهد که اگر سیستم این گونه باشد دیتایی که administrator می بیند نمی تواند قابل استفاده باشد.
۲. بعضی از برنامه های آنلاین قابل نفوذ برای سرقت اطلاعات هستند اما هنگامی که از این دیتا بیس استفاده کنیم داده هایی که سرقت می روند قابل استفاده نیستند.

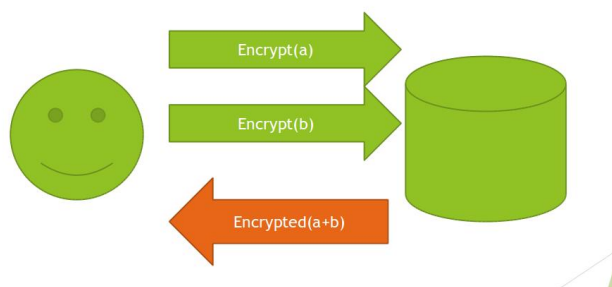
یک مثال مثالی از CryptDB را با شکل نشان می دهیم.

۳ رمز نگاری پیازی

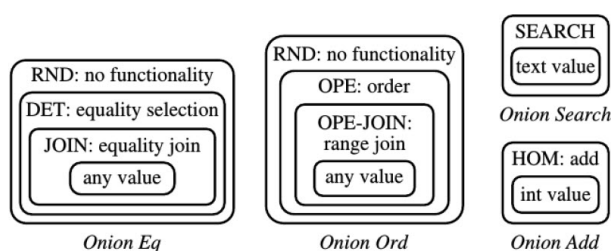
این رمز نگاری این گونه است که داده های ما در هر لایه رمز می شوند و به لایه بعدی می روند و هنگامی که می خواهد رمز آن باز رمز هر لایه آن باز می شود و مزیت اصلی این شیوه رمز نگاری این است که داده های ما چندیدن بار رمز شده اند.

۱.۳ لایه های پیازی

هر query نیاز به یک عملیات برای اجرا شدن دارد.



شکل ۲: یک مثال برای cryptDB



شکل ۳: رمز نگاری پیازی

۱.۱.۳ دستور GroupBy

در دستور GroupBy برابری حتماً باید چک شود.

۲.۱.۳ دستور sum

دستور sum وابسته به افزایش داده های رمز نگاری شده می باشد.

۴ شیوه های رمز نگاری

۱.۴ RND

تضمین های امنیتی قوی را فراهم می کند
این مشکل را دارد که یک متن با متن رمز نگاری شده اش یکسان باشد.

۲.۴ HOM

بطور خاص برای ستون از نوع داده صحیح طراحی شده

$$E(a + b) = E(a) + E(b)$$

۳.۴ SEARCH

این لایه منحصر به فرد برای نوع داده "text" طراحی شده است.

۴.۴ DET

این لایه دومین قوی ترین لایه است و dtermenstic می باشد و در دستورهایی مانند GroupBy کاربرد دارد.

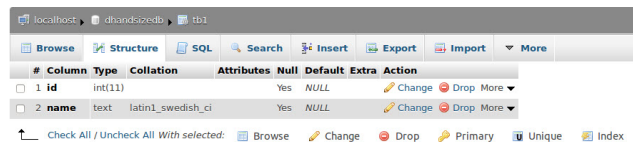
۵.۴ OPE

مخفف کلمه ی encryption Order-preserving می باشد .
اگر مقدار x کمتر از y باشد آنگاه می توان گفت که $OPE(x)$ از $OPE(y)$ کمتر است و در Query هایی مانند min و max و ORDERBY کاربرد دارد.

۶.۴ OPE-JOIN ، JOIN

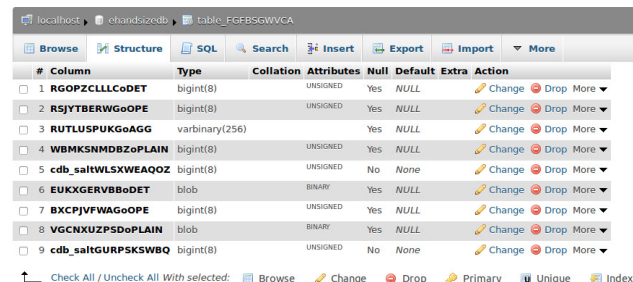
مانند OPE و DET می باشد و در Query هایی که join دارند نیز کاربرد دارد.

یک مثال به بیان یک مثال برای رمز نگاری می پردازیم و سپس آن را تحلیل می کنیم.



#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)		Yes	NULL			Change Drop More
2	name	text	latin1_swedish_ci	Yes	NULL			Change Drop More

شکل ۴: یک مثال



#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	RGOPZCLLLCoDET	bigint(8)		UNSIGNED	Yes	NULL		Change Drop More
2	RSJYTBerWGoOPE	bigint(8)		UNSIGNED	Yes	NULL		Change Drop More
3	RUTLUSPUKGoAGG	varbinary(256)			Yes	NULL		Change Drop More
4	WBMKSNMdBZoPLAIN	bigint(8)		UNSIGNED	Yes	NULL		Change Drop More
5	cdB_saltWLSXWEAQOZ	bigint(8)		UNSIGNED	No	None		Change Drop More
6	EUKGGERVBBoDET	blob		BINARY	Yes	NULL		Change Drop More
7	BXCPJVfWAGoOPE	bigint(8)		UNSIGNED	Yes	NULL		Change Drop More
8	VGCNXUZPSDoPLAIN	blob		BINARY	Yes	NULL		Change Drop More
9	cdB_saltGURPSKSWBQ	bigint(8)		UNSIGNED	No	None		Change Drop More

شکل ۵: یک مثال

<input type="checkbox"/> dhandsizedb	latin1_swedish_ci	1	203	16.0 KiB	0 B	16.0 KiB	0 B
<input type="checkbox"/> ehandsizedb	latin1_swedish_ci	1	194	128.0 KiB	0 B	128.0 KiB	0 B

شکل ۶: یک مثال

۵ اجرای CryptDB

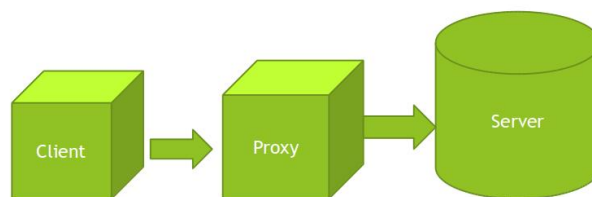
برای اجرای CryptDB باید مراحل زیر را انجام دهیم:

توجه: به دو عدد لپ تاپ یا VM که دارای Ubuntu هستند نیازمندیم

۱. ابتدا دو عدد لپ تاپ را به یک شبکه وصل می کنیم
۲. سرور باید دارای پورت شماره ۳۳۰۷ و Proxy باید دارای پورت شماره ۳۳۰۶ باشد.
۳. حال در Browser لپ تاپ سرور دستور
`http://localhost/phpmyadmin/`
را می زنیم و در قسمت Username و password باید Username و Password مربوط به خودمان که در هنگام نصب داده ایم را وارد کنیم.
۴. بعد از آن باید در ترمینال لپ تاپ proxy دستورات زیر را اجرا کنیم:

```
> /path/to/cryptdb/bins/proxy-bin/bin/mysql-proxy  
-plugins=proxy -event-threads=4  
-max-open-files=1024  
-proxy-lua-script=$EDBDIR/mysqlproxy/wrapper.lua  
-proxy-address=127.0.0.1:3307  
-proxy-backend-addresses=localhost:3306
```

که IP های Proxy و Server را در قسمت مربوط به خودشان قرار می دهیم.
۵. حال باید در terminal دیگری مربوط به Proxy دستور
`mysql -u root -p letmein -h 127.0.0.1 -P 3307`
userName ای که ما با آن کار می کردیم root و password متناظر با آن letmein می بود جناب عالی می توانید username و password خود را انتخاب و وارد کنید.
این دستور ما را به terminal قبلی وصل می کند و از آن جا به سرور وصل می شویم.
۶. حال وارد SQL شده ایم و می توانیم Query های مربوطه را بدهیم.



شکل ۷: شکل مربوط به اجرای CryptDB

۶. موارد استفاده CryptDB

۱. BigQuery که در شرکت های بزرگ مانند google کاربرد دارد.
۲. رمزنگاری SQL-Server شرکت ماکروسافت
۳.

۷ Query ها در CryptDB

تمام Query ها در CryptDB با SQL یکساز هستند و هیچ Syntax جدیدی ندارد.

۸ نتایج تست CryptDB در BenchMark ها

توجه: نتایج زیر بر روی یک ماشین با RAM ۶۴GB و CPU ۲,۴GHz اجرا شده است.

Query	MySQL	Server	Proxy	Proxy*
Select by =	0.10	0.11	0.86	0.86
Select join	0.10	0.11	0.75	0.75
Select range	0.16	0.22	0.78	28.7
Select sum	0.11	0.46	0.99	0.99
Delete	0.07	0.08	0.28	0.28
Insert	0.08	0.10	0.37	16.3
Update set	0.11	0.14	0.36	3.80
Update inc	0.10	0.17	0.30	25.1
Overall	0.10	0.12	0.60	10.7

شکل ۸: نتایج اجرای BenchMark بر روی CryptDB

در این بخش به مطالعه و بررسی MySQL java می پردازیم .
در این بخش ابتدا به مطالعه MySQL driver for JDBC می پردازیم و یک مثال برای آن می زنیم.
توجه: تمامی مطالب در Ubuntu ساخته شده و نیز تست شده اند.

۹ JDBC

یک API برای زبان برنامه نویسی Java می باشد که در با استفاده از آن می توان به عنوان یک client به DataBase متصل شد.
در این API می توان از Method های مختلفی برای فرستادن query به DataBase استفاده کرد.
این API در پکیج java.sql قرار دارد.
توجه: برای استفاده از JDBC به یک درایور JDBC برای DataBase نیاز داریم.

۱۰ about MySQL

MySQL یک برنامه مدیریت DataBase متن باز می باشد که multiUser و multiThread است که هر دو این ویژگی ها ویژگی های بسیار مهمی هستند. (MySQL در برنامه نویسی وب و کار های مربوط به وب بیشتر استفاده می شود).
MySQL متعلق به شرکت Oracle می باشد و در بسیاری از سیستم عامل ها (مانند Unix ، Linux ، Windows ، Mac ، ...) به صورت پیش فرض موجود می باشد.
MySQL در دو نسخه MySQL server و MySQL embedded توضیح می شود.

۱۱ موارد مورد نیاز

قبل از شروع به راه اندازی ما به چند libraries مهم نیاز داریم که حتماً باید نصب شوند.

- mySQL-server
- mySQL-client
- JDK با این پکیج باید برنامه ها را compile کنیم.
- JRE با این پکیج برنامه های java را اجرا می کنیم.
- JDBC

۱۲ چگونگی نصب در Linux

- install MySQL
sudo apt-get install mysql-server
- connect to SQL
mysql -u root -p
after p we must enter password

- in MySQL
 - 1- mysql> CREATE DATABASE TEST;
 - 2- mysql> CREATE USER 'testuser'@'localhost' IDENTIFIED BY 'test2';
 - 3- USE TEST;
 - 4- GRANT ALL ON TEST.* TO 'testuser'@'localhost';
 - 5-

۱۳ در MySQL جاوا

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;

public class MyClient {

    public static void main(String[] args) {

        Connection con = null;
        Statement st = null;
        ResultSet rs = null;

        String url =
            "jdbc:mysql://192.168.20.154:3307/YourQuery";
        String user = "you user name";
        String password = "your password";

        try {
            con = DriverManager.getConnection(url, user,
                password);
            st = con.createStatement();
            rs = st.executeQuery("SELECT MyClient()");

            if (rs.next()) {
                System.out.println(rs.getString(1));
            }

        } catch (SQLException ex) {
            Logger lgr =
                Logger.getLogger(MyClient.class.getName());
            lgr.log(Level.SEVERE, ex.getMessage(), ex);
        } finally {

            try {
                if (rs != null || st != null || con != null) {
                    rs.close();
                }
            }
        }
    }
}
```

```

    }

    } catch (SQLException ex) {
        Logger lgr =
            Logger.getLogger(MyClient.class.getName());
        lgr.log(Level.WARNING, ex.getMessage(), ex);
    }
}
}
}
}

```

که توضیح بخش های مختلف کد در زیر قرار دارد.

۱۴ نحوه اتصال به MySQL

برای اتصال به MySQL باید از یک query به شرح زیر استفاده کرد.

String url = "jdbc:mysql://localhost:3306/testdb";
 که در این Query باید حتماً Host و port و نام دیتابیس را مشخص کنیم.

۱۵ برقراری ارتباط با DataBase

برای برقراری ارتباط با DataBase باید از یک Connection URL استفاده کنیم که به صورت زیر می باشد.

con = DriverManager.getConnection(url, user, password);
 برای اتصال به دیتابیس باید از url (که در مورد قبل توضیح داده شد) و userName و password استفاده کنیم.
 همان نام کاربری است که ما در دیتابیس داریم و Password نیز همان رمز عبور ما می باشد.

حال برای برقراری ارتباط باید از Method ای با نام createStatment() استفاده کنیم که شرح آن نیز در زیر قرار دارد:

st = con.createStatement();
 این method یک Object از نوع Statment می سازد که برای ارسال SQLStatment به DataBase استفاده می شود.

۱۶ نحوه فرستادن Query

برای فرستادن Query باید از Statment ای که در مرحله قبل بدست آمد استفاده کنیم به این صورت که
 rs = st.executeQuery("SELECT MyClient()");
 با استفاده از آن Statment می توانیم ای Method به نام executeQuery را برای آن صدا بزنیم که ورودی Method را Query که می خواهیم می دهیم.
 خروجی این Method یک ResultSet می باشد که همان ستون های خروجی ما هستند.

۱۷ دریافت result

```
if (rs.next()) {  
    System.out.println(result.getString(1));  
}
```

حال با استفاده از resultSet ای که از مرحله قبل داشتیم شروع به چاپ خروجی می کنیم به این صورت که:
cursore موجود در جدول به اولین خانه اشاره می کند که تابع next() آن را به سطر بعدی می برد و اگر سطر دیگری باقی نمانده باشد این تابع مقدار false را بر می گرداند.
و تابع getString() مقدار ستون را بر می گرداند.

۱۸ گرفتن خطاها

```
} catch (SQLException ex) {  
    Logger lgr =  
        Logger.getLogger(MyClient.class.getName());  
    lgr.log(Level.SEVERE, ex.getMessage(), ex);  
}
```

تمام خطاهایی که بوجود می آیند را در کنسول چاپ می کنیم.

```
} finally {  
  
    try {  
        if (rs != null || st != null || con != null) {  
            rs.close();  
        }  
  
        } catch (SQLException ex) {  
            Logger lgr =  
                Logger.getLogger(MyClient.class.getName());  
            lgr.log(Level.WARNING, ex.getMessage(), ex);  
        }  
  
    }  
}
```

در قسمت finally سعی در بستن resource های دیتابیس می کنیم.

```
} catch (SQLException ex) {  
    Logger lgr =  
        Logger.getLogger(Version.class.getName());  
    lgr.log(Level.WARNING, ex.getMessage(), ex);  
}
```

در این قسمت error هایی که نمی گذارند resource های دیتابیس بسته شود را چاپ می کنیم.