Database and Security

Gary KL Tam

Department of Computer Science
Swansea University

Security

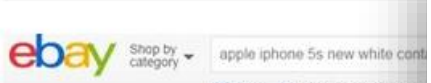
- Security important?

Sony fined over 'preventable' PlayStation data hack

Sony Computer Entertainment Europe has

eBay redirect attack puts buyers' credentials at risk

By Leo Kelion
Technology desk editor



USB 'critically flawe researchers say

By Dave Lee
Technology reporter, BBC News

Karsten Nohl shows Dav

Cyber-security exper safety and security o

Heartbleed bug: What you need to know

By Jane Wakefield
Technology reporter

This week it has emerged that a major security flaw at the heart of the internet may have been exposing users' personal information and passwords to hackers for the past two years.

Shellshock: 'Deadly serious' new vulnerability found

By Dave Lee
Technology reporter, BBC News

Apple confirms accounts compromised but denies security breach

Apple say that hackers targeted celebrities' passwords, as Rory Cellan-Jones reports

Topics

- DBMS security
  - User accounts, privileges and views
  - SQL Injections
  - Network and Encryption

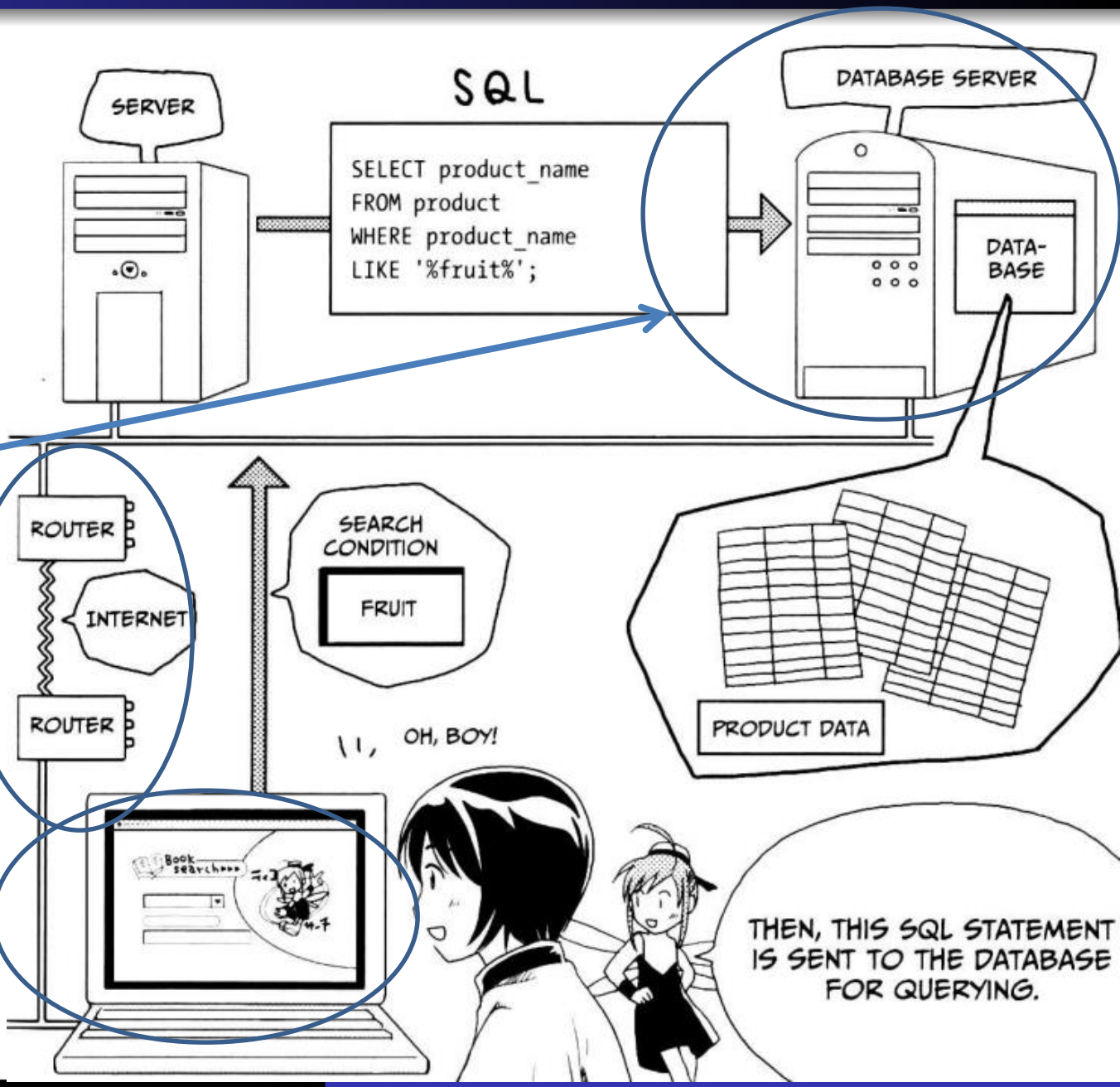
Database and Web

- Online Bookstore

Database security

Network security
Encryption

User accounts,
Privileges and Views



SQL Revisit

- Recall, SQL provides:
  - Data Manipulation Language (DML)
  - Data Definition Language (DDL)
  - Data Control Language (DCL)

Specify access controls

DBMS Security Support

- DBMS can provide some security
  - Each database user has an account, username & password
  - These are used to identify a user and control their access to information
- DBMS verifies password and checks a user's permissions when they try to
  - Retrieve data
  - Modify data
  - Modify the database structure

Example

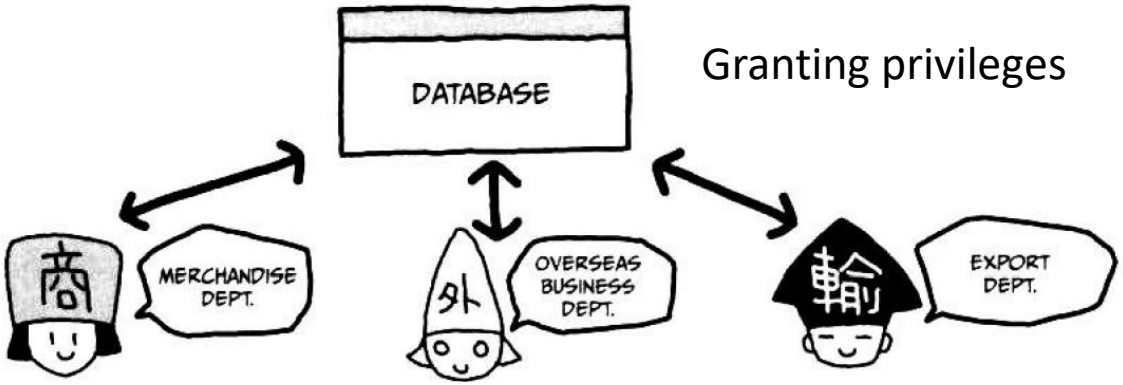
- Login MySQL using root account

C:\Windows\system32\cmd.exe - mysql -u root -p

```
D:\xampp\mysql\bin>mysql -u root -p
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 472
Server version: 5.5.27 MySQL Community Server (GPL)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Permissions and Privilege

- SQL uses privileges to control access to tables and other database objects
  - SELECT privilege
  - INSERT privilege
  - UPDATE privilege
  - DELETE privilege



Granting privileges

- The owner (creator) of a ***database*** has all privileges on all objects in the database, and can ***grant*** these to others
- The owner (creator) of an ***object*** has all privileges on that object and can ***pass*** them on to others

Privileges in SQL

```
GRANT <privileges>
    ON <object>
    TO <users>
[WITH GRANT OPTION]
```

- **<privileges>** is a list of **SELECT <columns>**, **INSERT <columns>**, **DELETE**, and **UPDATE <columns>**, or simply **ALL**
- **<users>** is a list of user names or PUBLIC
- **<object>** is the name of a table or view (discuss later)
- **WITH GRANT OPTION** means that the users can pass their privileges on to others

Privileges Examples

```
GRANT ALL ON Fruit
  TO King
  WITH GRANT OPTION;
```

The user 'King' can do anything to the Fruit table, and can allow other users to do the same (by using **GRANT** statements)

```
GRANT SELECT, UPDATE(Quantity) ON
  Fruit TO Farmer;
```

The user 'Farmer' can view the entire Fruit table, and can change Quantity values, but cannot change other values or pass on their privilege

No insert/delete

Removing Privileges

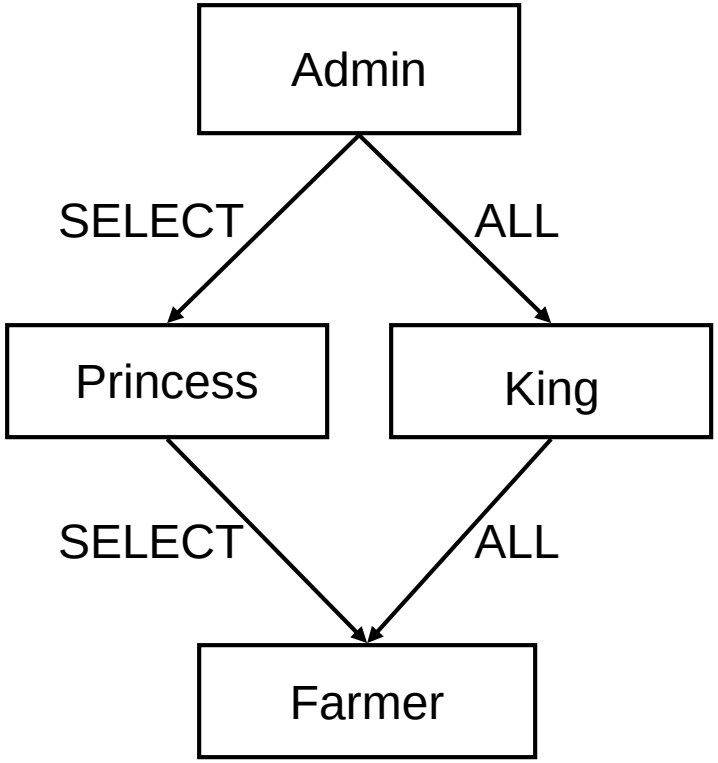
- If you want to remove a privilege you have granted you use

```
REVOKE <privileges>
  ON <object>
  FROM <users>
```

- If a user has the same privilege from other users then they keep it
- All privileges dependent on the revoked one are also revoked

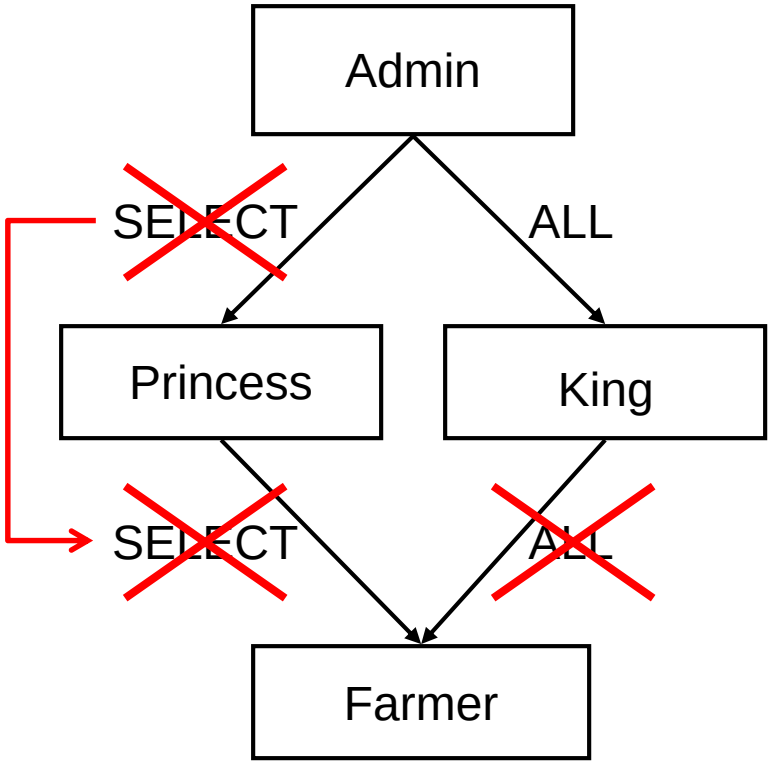
Granting Privileges

- Example
  - ‘Admin’ grants ALL privileges to ‘King’, and SELECT to ‘Princess’ with grant option
  - ‘King’ grants ALL to ‘Farmer’
  - ‘Princess’ grants SELECT to ‘Farmer’



Removing Privileges

- Example
  - ‘King’ revokes ALL from ‘Farmer’
  - ‘Farmer’ still has SELECT privileges from ‘Princess’
  - ‘Admin’ revokes SELECT from ‘Princess’
  - ‘Farmer’ loses SELECT also



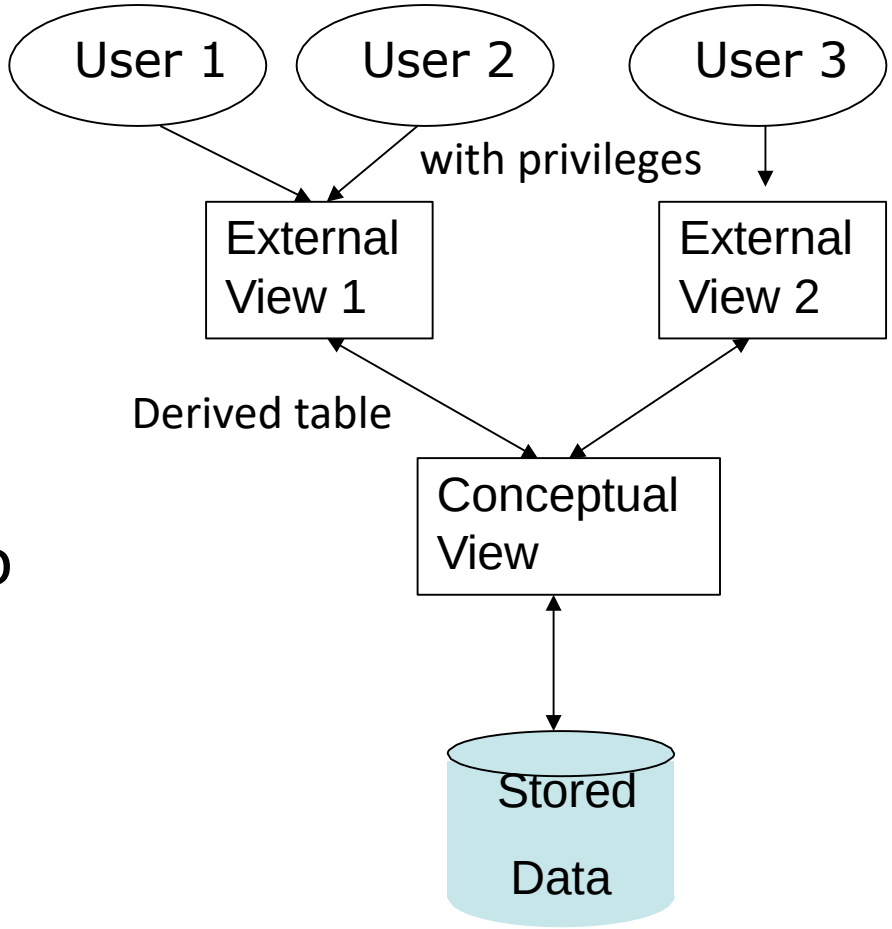
Views

- Privileges work at the level of tables
  - You can restrict access by column
  - You cannot restrict access by row
- Views, along with privileges, allow for customised access

- Views provide 'derived' tables
  - A view is the result of a SELECT statement which is treated like a table
  - You can SELECT from (and sometimes UPDATE etc) views just like tables

Using Views and Privileges

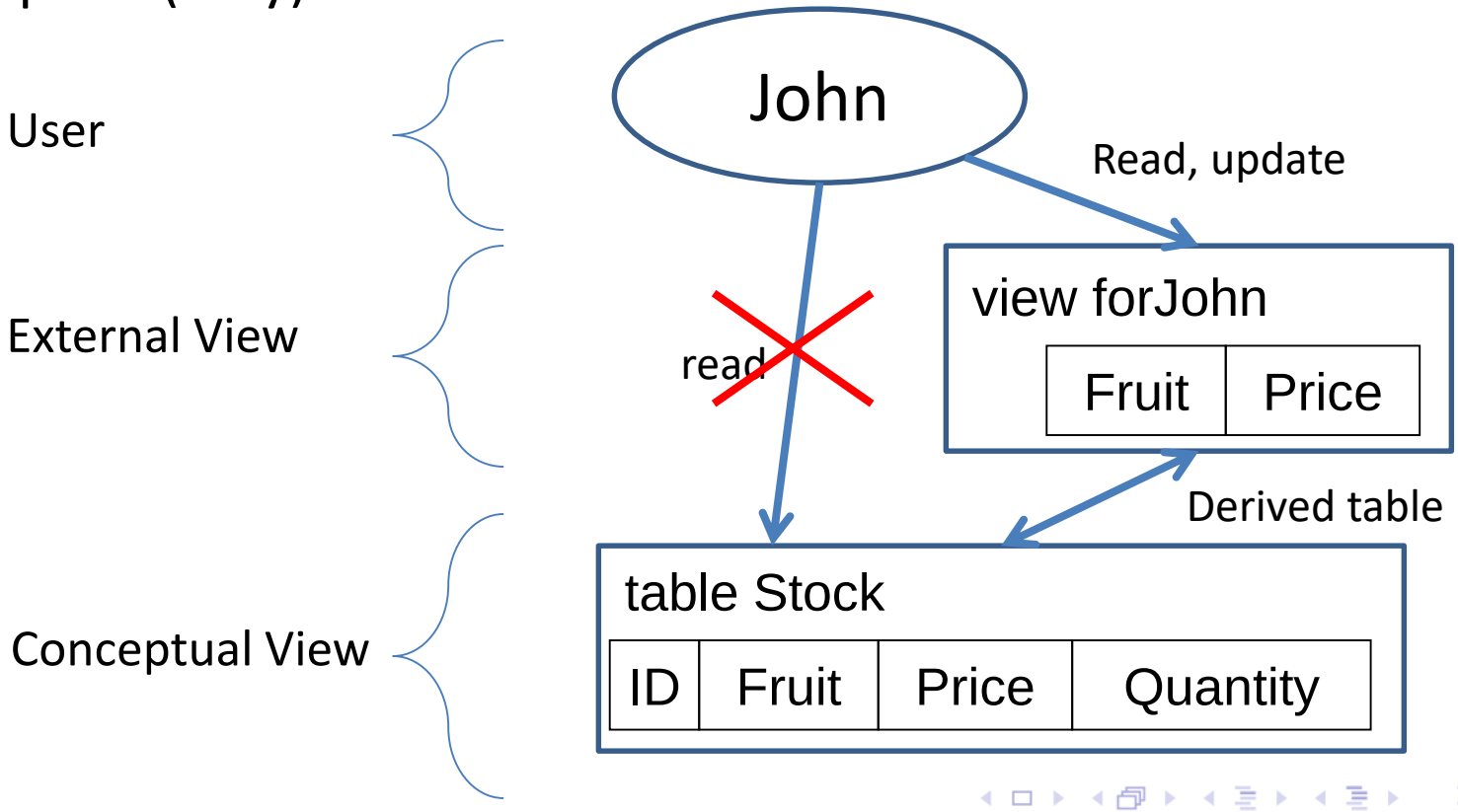
- Views and privileges are used together to control access
  - A view is made which contains the necessary information only
  - Privileges are granted to / revoke from view, rather than the underlying tables



Using Views and Privileges

Example

- User 'John' originally allowed to read from Stock
- We want to revoke 'John' access to Stock table, but let him read the fruit and price, and be able to update the price (only)



Using Views and Privileges

Example

1. Create a view

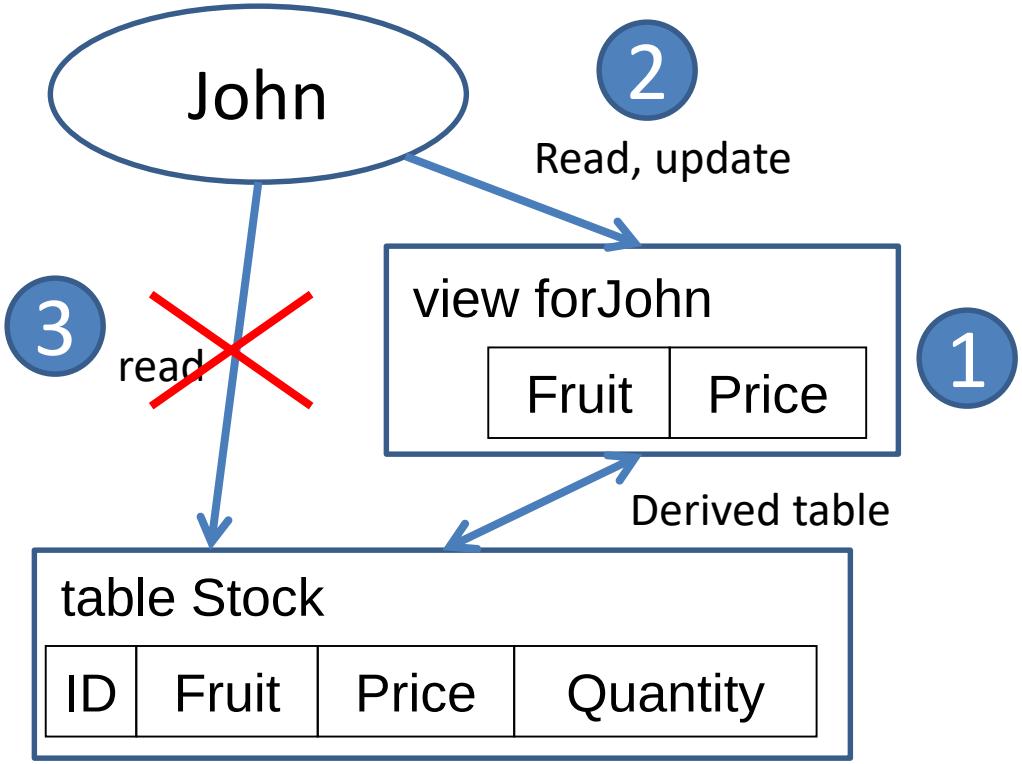
```
CREATE VIEW forJohn
AS SELECT Fruit, Price
  FROM Stock
```

2. Set the privileges

```
GRANT SELECT, UPDATE
(Price)
ON forJohn
TO John
```

3. Revoke existing privilege

```
REVOKE ALL ON Stock FROM
John
```



Topics

- DBMS security
  - User accounts, privileges and views
  - SQL Injections
  - Network and Encryption

Example

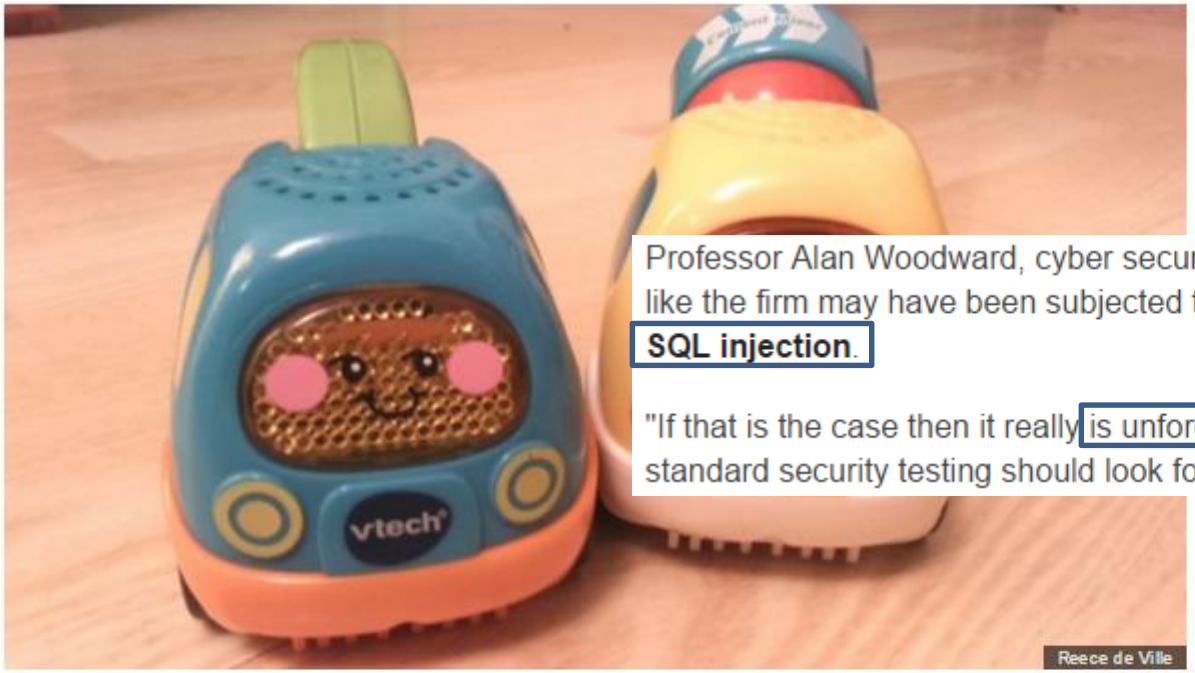
Children's electronic toy maker Vtech hacked

By Zoe Kleinman
Technology reporter, BBC News

27 November 2015 | Technology

App store database hacked

http://www.bbc.co.uk/news/technology-34944140



Professor Alan Woodward, cyber security expert at Surrey University, said it looks like the firm may have been subjected to a simple hacking technique known as an **SQL injection**.

"If that is the case then it really is unforgivable - it is such an old attack that any standard security testing should look for it," he said.

Vtech, a company which specialises in electronic toys and educational material for children, has had its app store database, Learning Lodge, hacked.

SQL and PHP

- Use SQL in PHP

```
<?php
// create connection $conn to a database at localhost
// using database user account "WebUser"
$conn = mysql_connect("localhost","WebUser","abc123");

// select database to use
mysql_select_db("spj_db", $conn);

// SQL query and store result in $result
$result = mysql_query("SELECT * FROM SPJ;");

// close connection
mysql_close($conn);

?>
```

SQL as a string

SQL and PHP

- Use SQL in PHP, with application user account

```
<?php
$con = mysql_connect("localhost", "WebUser", "abc123");
if (!$con)
{ die('Could not connect: ' . mysql_error()); }

// authenticate application user
$sql = "SELECT * FROM users
WHERE user='$_POST['user']'
AND password='$_POST['pwd']'";

// ... Query the database with SQL string ...
$result = mysql_query($sql);
$num_results = mysql_num_rows($result);
if ($num_results > 0){
        // ... return record > 0, user authenticated ...
        // ... some important code ...
}
mysql_close($con);
?>
```

Form data submitted from User

Build a SQL string

User Account

- Example:

```
$_POST['user']: Gary
$_POST['pwd'] : abc
$sql= " SELECT * FROM users
        WHERE user='Gary'
        AND password='abc' "
```

- users table stores username and password of application user

Hacker: SQL Injection

- A hacker may do this trick:
- $_POST['user'] : john;
  $_POST['pwd'] : ' OR ''=';
  $sql="SELECT * FROM users
  WHERE user='john' AND
  password= '' OR ''='' "

' or ''='

- ''='' always true. (It reads: empty string ' ' equals empty string)
- This means that anyone could log in without a valid password!

SQL Injection

- ***Some*** protections:
  - **PHP: mysql_real_escape_string()**

    The function adds an escape character, the backslash, \, before certain potentially dangerous characters (e.g. ') in a string passed in to the function.

```
$sql="SELECT * FROM users
        WHERE user='john' AND
        password= '\' OR \'\'=\''  "
```

  - **PHP: PDO (prepared statement)**

```
$oDB=new PDO('... your connection details... ');
$hStmt=$oDB->prepare( "select * from users
                        where user=:userid and password=:pwd");
$hStmt->execute(array(':userid'=>$nUserID,':pwd'=>$nPwd));
```

PDO object *talks directly to ODBC driver*; avoids unsafe string interpretation.

SQL Injection

- Old version of PHP PDO calls mysql_real_escape_string()
- Always update to latest version

- Note:
- mysql_real_escape_string and PDO are not perfect.
- PDO, in general, safer.

- Extra reading:

http://www.w3schools.com/php/func_mysql_real_escape_string.asp
http://stackoverflow.com/questions/6327679/what-does-mysql-real-escape-string-really-do
http://www.iodigitalsec.com/mysql_real_escape_string-wont-magically-solve-your-sql-injection-problems/
http://code.tutsplus.com/tutorials/why-you-should-be-using-phps-pdo-for-database-access--net-12059
http://software-security.sans.org/developer-how-to/fix-sql-injection-in-php-using-prepared-statements
http://stackoverflow.com/questions/134099/are-pdo-prepared-statements-sufficient-to-prevent-sql-injection

SQL Injection

- Don’t laugh

5 Oct, 2016

TalkTalk fined £400,000 for theft of customer details

5 October 2016 | Business

Share

TalkTalk has been fined a record £400,000 for poor website security which led to the theft of the personal data of nearly 157,000 customers.

TalkTalk hit with record fine for data breach

Company ordered to pay £400,000 for security failings that led to theft of 150,000 customers' details

Database software, which held details of customers inherited from the 2009 takeover of a rival firm, Tiscali, was out of date.

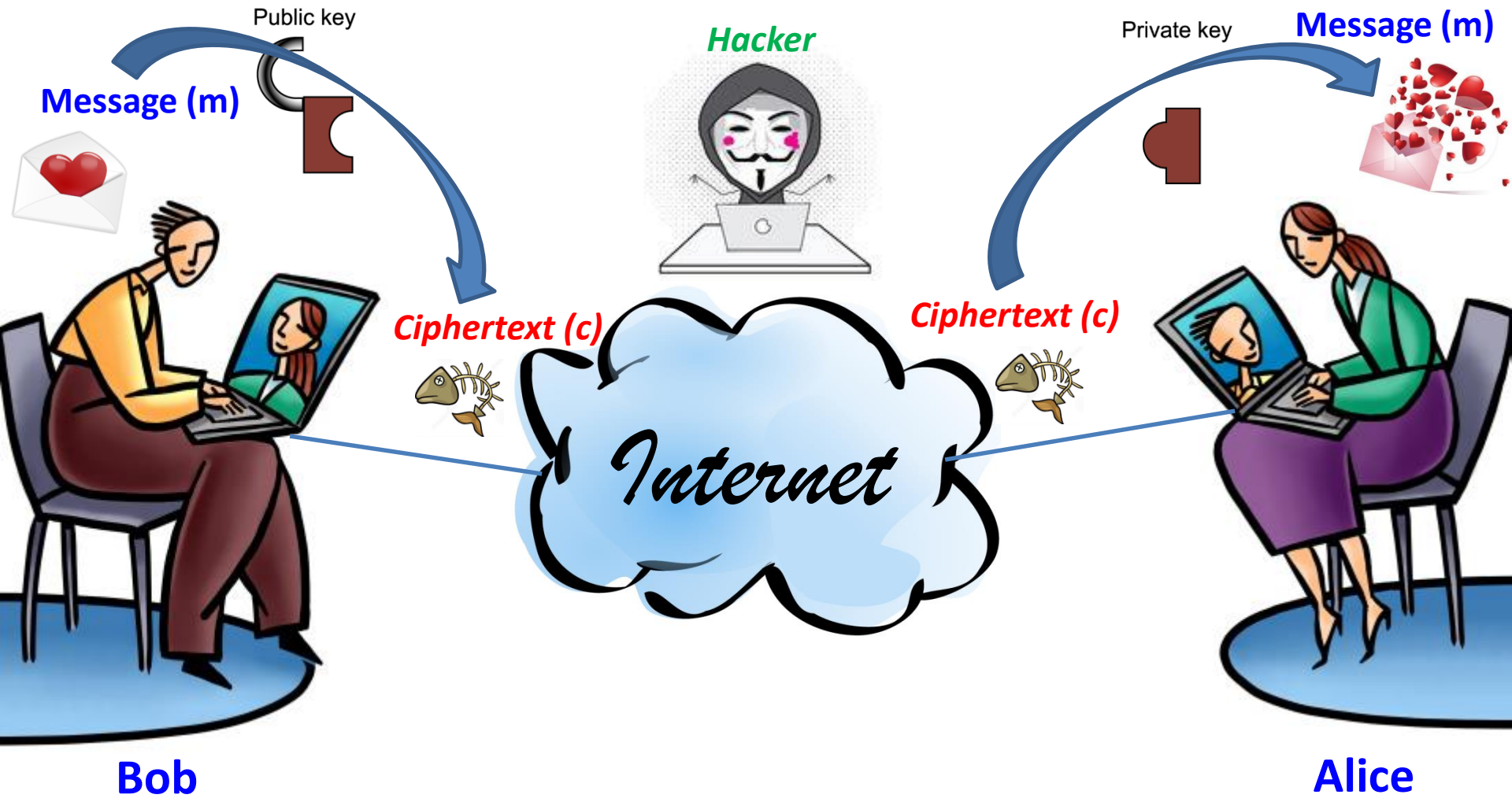
As a result, the attacker got hold of the customers' details by attacking three vulnerable web pages, using a well known hacking technique called SQL injection.

A bug, which could have been fixed, allowed the attacker to by-pass restrictions, but the company was simply unaware of the problem or that it could be solved easily.

Topics

- DBMS security
  - User accounts, privileges and views
  - SQL Injections
  - Network and Encryption

Network Security and Encryption



RSA Cryptosystem

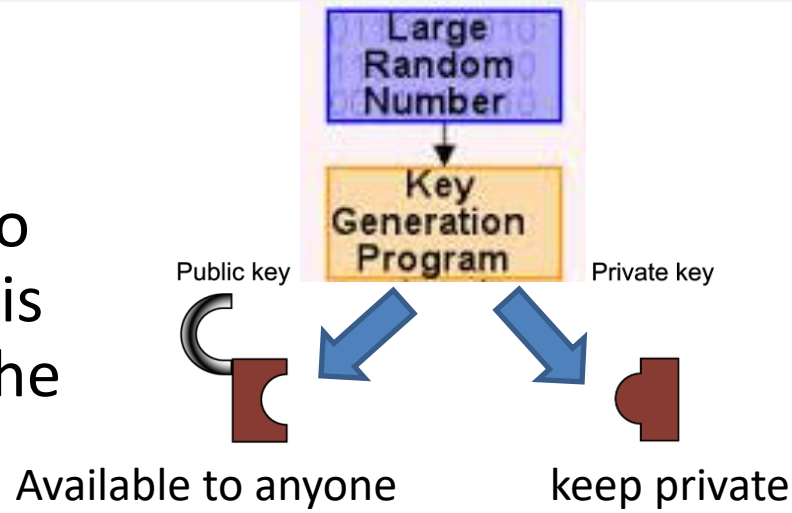
- Consider that Bob wants to send Alice a message **m** over the Internet.

- Let us regard **m** in its binary form, namely, **m** is a sequence of 0's and 1's, and therefore, can be regarded as a (perhaps very big) integer. As **m** needs to be delivered over a public network, it may be intercepted by a hacker. Our goal is to encrypt **m** into another integer **C** so that
  - It is very difficult for the hacker to infer **m** from **C**.
  - It is very easy for Alice to restore m from **C**.

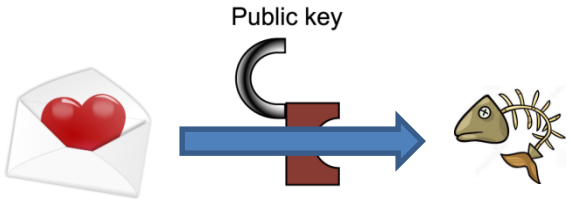
- **C** is therefore called a ***ciphertext***.

RSA Cryptosystem

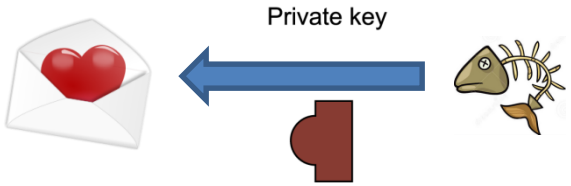
- Three steps:
  - **Preparation**: Alice prepares two keys: public and private key. This step is carried out only once. The keys will be used forever.



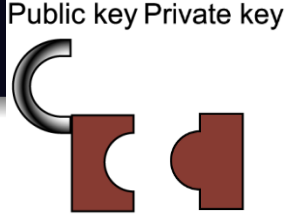
  - **Encryption**: Bob encrypts his message **m** for Alice into a ciphertext **C** using Alice's public key.



  - **Decryption**: Alice converts **C** back to **m** using her private key



Alice's Preparation



- Choose randomly 2 large prime numbers p & q.
  1. Let $n = p\,q$.
  2. Let $\varphi = (p - 1)(q - 1)$.
  3. Choose a number $e \in [1, \varphi - 1]$ that is co-prime to φ. ***co-prime*** means greatest common divisor is 1.
  4. Compute $d \in [1, \varphi - 1]$ such that $e \cdot d \pmod{\varphi} = 1$.

  5. Announce to the world the pair (e, n) - public key.
  6. Keep secret to herself the pair (d, n) - private key.

Example

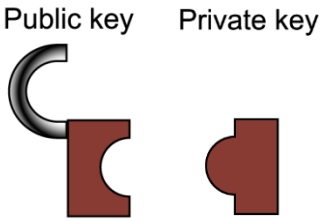
1. Choose $p = 3$ and $q = 11$.
2. $n = p\,q = 33$.
3. $\varphi = (p - 1)(q - 1) = 2 \times 10 = 20$.
4. Choose a number e that is co-prime to φ.

   Choice of e : {3, 7, 9, 11, 13, 17, 19} $\in [1, \varphi - 1]$
5. Compute d such that $e \cdot d \pmod{20} = 1$

   It means: $(7 \times d) / 20 = ?$ with reminder 1

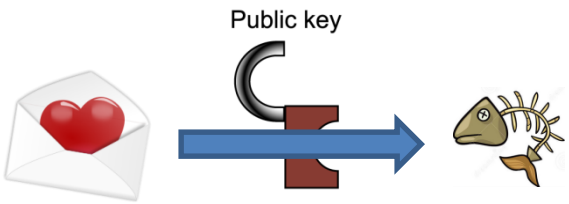
   Choice of d = 3 $\in [1, \varphi - 1]$ $(7 \times 3{=}21)\%20{=}1$

6. Announce the public key (7, 33).
7. Keep the private key (3, 33).



https://asecuritysite.com/encryption/rsa
http://jcla1.com/blog/rsa-public-private-key-encryption-explained/

Encryption



- Converts m to C as follows:

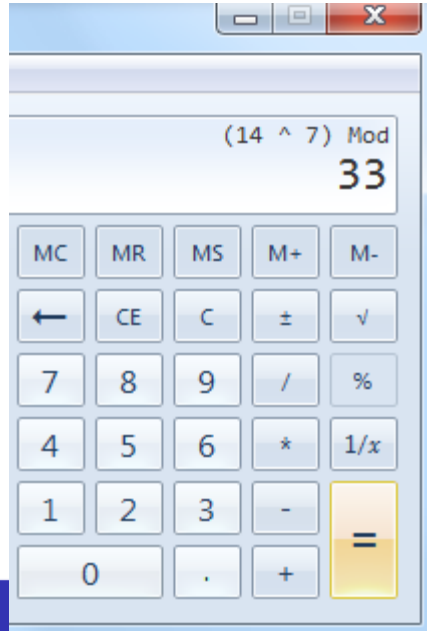
$$C = m^e \pmod{n}$$

$14_{10}=1110_2$

- Example
  - Assume that m = 14.
  - Alice's public key is (e, n) = (7, 33). Then:

$$C = 14^7 \pmod{33} = 20$$

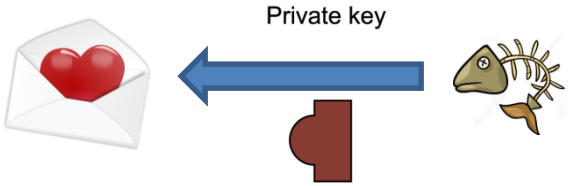
  - Bob sends C to Alice.



Decryption

- Recovers m from C as follows:

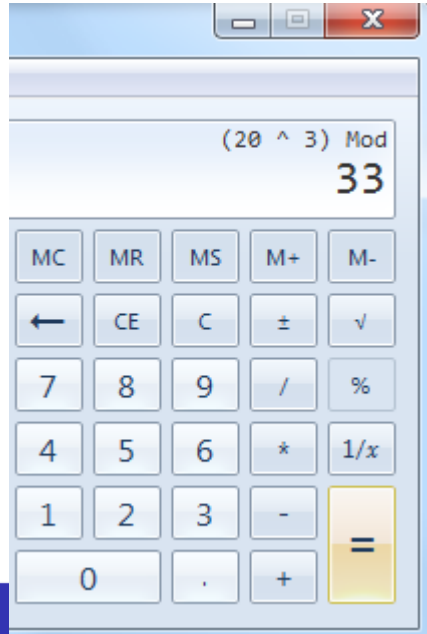
$$m = C^d \pmod{n}$$



- Example
  - Assume that C = 20.
  - Alice's ***secret*** private key (d, n)=(3, 33).

$$m = 20^3 \pmod{33} = 14$$

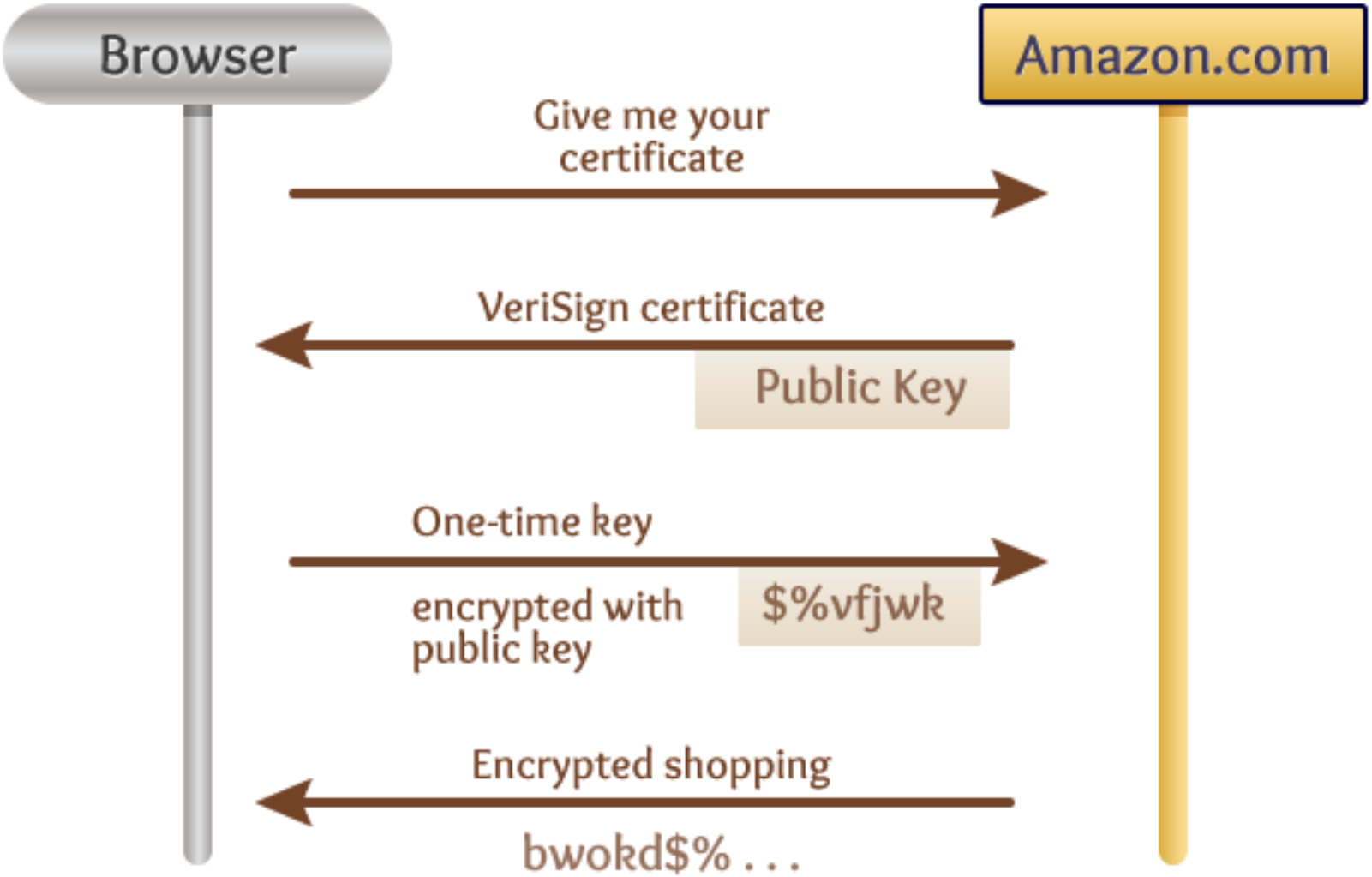
  - Alice recovers message (m= 14) from C.



RSA in http

- https://

s – secure
implemented as a Secure Socket Layer (SSL)



How to break RSA?

1. *Factor n back into p and q.*

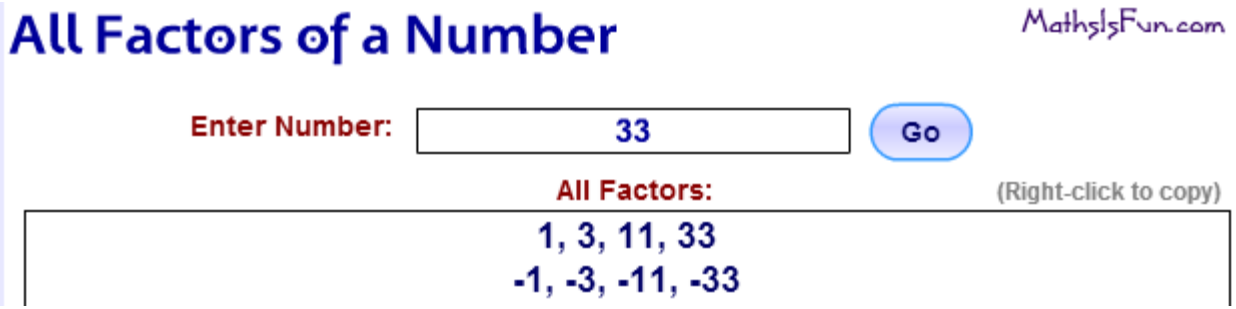
   n is available in public key!

2. Once this is done, essentially the entire preparation carried out by Alice has been revealed. So the following steps become trivial.
3. Obtain φ.
4. Compute d from e and φ.
5. Convert C using d and n back into m.

How to break RSA?

In our earlier example:
The encryption can be easily broken because it is trivial to factor n = 33 into p = 3 and q = 11.
This is because both p and q are small.

Factor n back into p and q?



http://www.mathsisfun.com/numbers/factors-all-tool.html

RSA

The presumed security of RSA is based on the following hypothesis:

Assumption

When primes p and q are big, it is computationally intractable to factor $n = pq$.

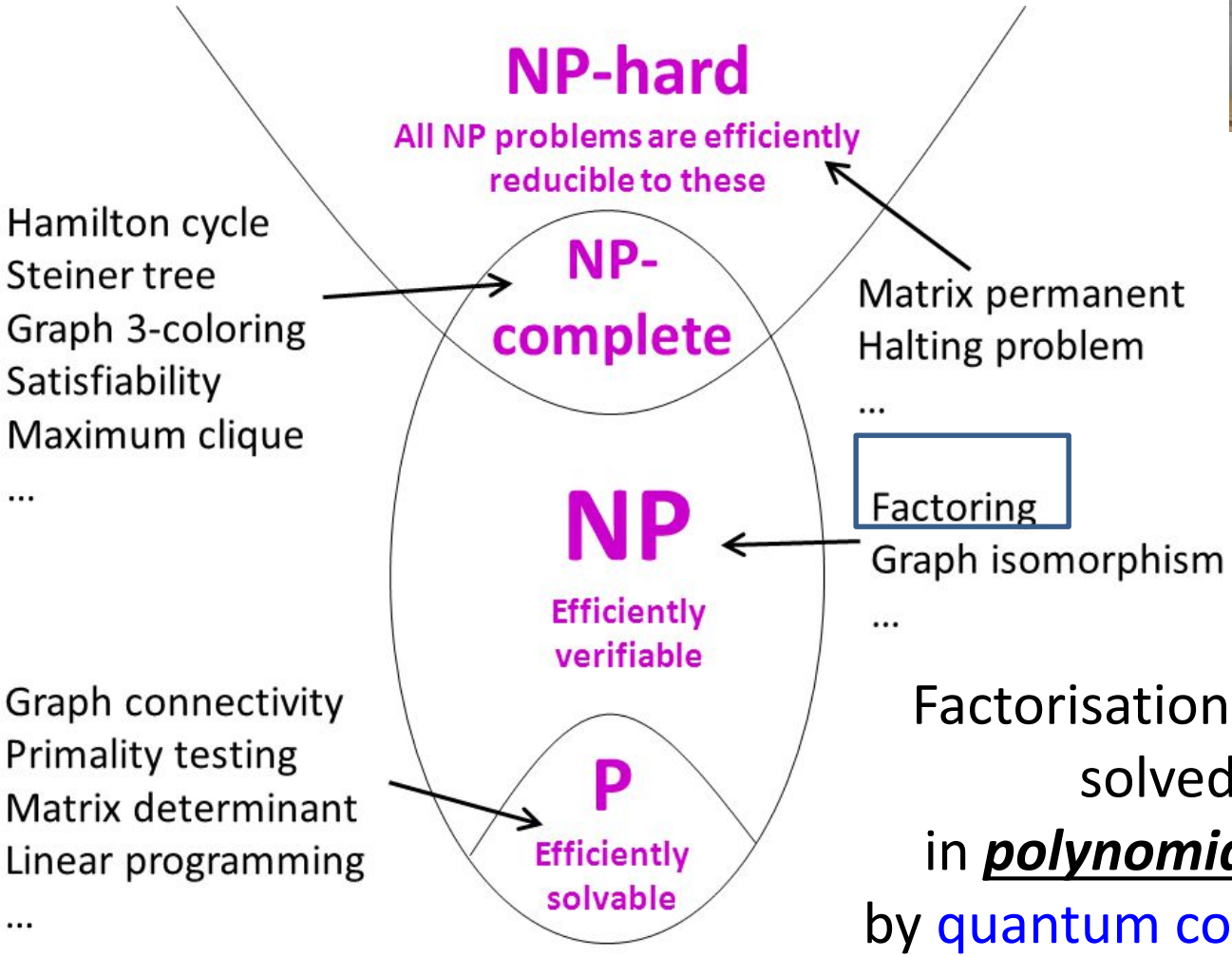
In practice, p and q should both be, for example, 1024 bits long.

WARNING

The best factorization algorithm known today requires excessively long time (e.g., a month) to factor a large n even on the fastest computer. However, nobody has ever proved that the hypothesis is correct. Even worse, nobody has ever proved that factoring n is the fastest way to break RSA. In other words, there may exist a clever algorithm (for either factoring n or breaking RSA in a different manner) that remains undiscovered yet. Once found, RSA algorithm will become insecure, and therefore, obsolete.

P vs NP, NPC, NP-hard (Out-Of-Syllabus)

- P vs NP?
  https://www.youtube.com/watch?v=YX40hbAHx3s



Factorisation can be solved in ***polynomial time*** by quantum computing.

How large a number in 128bit

- 32bit:
  - 2^{32} = 4,294,967,296.
- 64bit:
  - 2^{64} = 18,446,744,073,709,551,616.
- 128bit:
  - 2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456
  - 39 digits
- 1024bit?
  - ~300 digits

RSA


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

NSA, GCHQ

- RSA is difficult to break, but doesn't mean it is unbreakable…

NSA is aiming at processing 100,000 requests per hour by 2011, this means that they should be able to decrypt and reinject data of 100,000 VPN users…

http://www.theguardian.com/world/2013/sep/05/nsa-gchq-encryption-codes-security
http://www.hacker10.com/internet-anonymity/secret-documents-show-the-nsa-is-spying-on-vpn-users/comment-page-1/

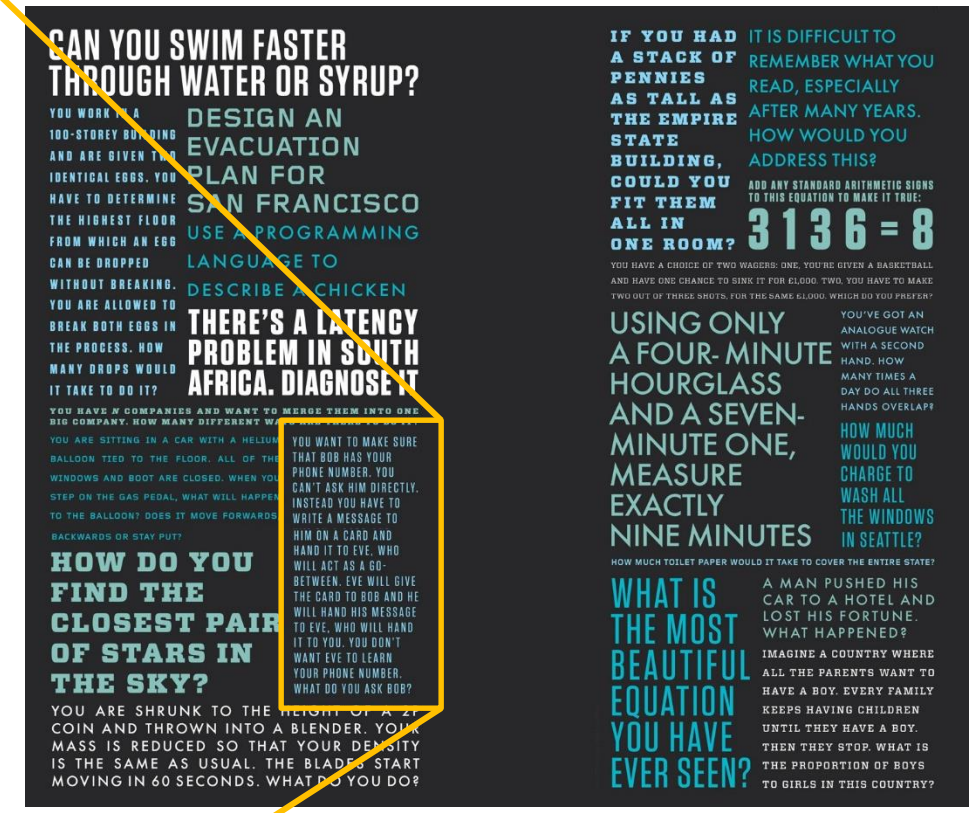
CSC318 Cryptography and IT-Security

- The aim of CSC318 is to examine theoretical and practical aspects of computer and network security.

Security threats and their causes.
Security criteria and models.
Cryptography: including basic encryption, DES, AES, hash functions.
Access Control.
Security tools and frameworks: including IPSec, TLS, SSL, SSH and related tools.
Vulnerabilities and attacks: including port scanning, packet sniffing, SQL injection.
Security issues in wireless networks.
Security on the cloud..
Block Chain Technology and Bitcoin
Penetration Testing.
Tor Network.

Google Interview Question

- 8. You want to make sure that Bob has your phone number. You can't ask him directly. Instead you have to write a message to him on a card and hand it to Eve, who will act as a go-between. Eve will give the card to Bob and he will hand his message to Eve, who will hand it to you. You don't want Eve to learn your phone number. What do you ask Bob?



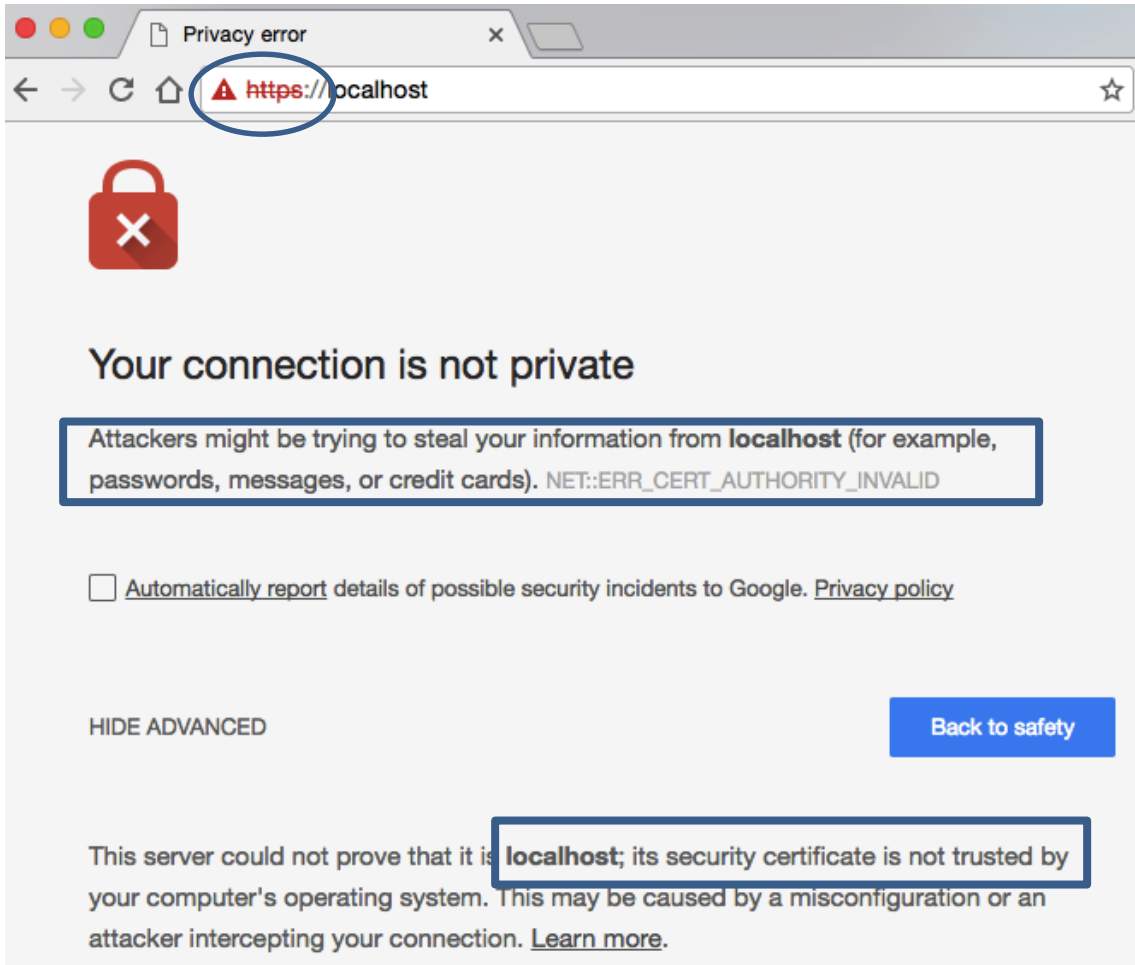
R S A

http://www.wired.co.uk/magazine/archive/2012/05/start/want-to-work-at-google

Personal Security - counterexample

- Secure?
  - https?
  - padlocked?
- Check Web address
  - Beware of phishing!
- Check certificate
  - Valid for the address?
  - Expired or not?
  - Trustable issuer?

XAMPP is for development purpose - https (443)



Personal Security

https://www.yamaha.com/en/

YAMAHA

Sources Network Timeline Security »

Security Overview

This page is secure (valid HTTPS).

Valid Certificate

The connection to this site is using a valid, trusted server certificate.

View certificate

Secure Connection

The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.2), a strong key exchange (ECDHE_RSA with P-256), and a strong cipher (AES_128_GCM).

Secure Resources

All resources on this page are served securely.

Certum Trusted Network CA
Certum Global Services CA SHA2
SpaceSSL CA
*.yamaha-europe.com

*.yamaha-europe.com
Issued by: SpaceSSL CA
Expires: Saturday, 18 February 2017 14:36:08 Greenwich Mean Time
This certificate is valid

Details

Subject Name	
Country	DE
Common Name	*.yamaha-europe.com
Email Address	peter@kaminski.biz
Issuer Name	
Country	PL
Organization	Unizeto Technologies S.A.
Organizational Unit	SpaceSSL Certification Authority
Common Name	SpaceSSL CA

OK

Personal Security

HSBC Holdings plc [GB] https://www.security.hsbc.co.uk/

HSBC

Sources Network Security » 1 6

Security Overview

This page is secure (valid HTTPS).

Valid Certificate

The connection to this site is using a valid, trusted server certificate.

View certificate

Secure Resources

All resources on this page are served securely.

Obsolete Connection Settings

The connection to this site uses a strong protocol (TLS 1.2), an obsolete key exchange (RSA), and an obsolete cipher (AES_256_CBC with HMAC-SHA1).

VeriSign Class 3 Public Primary Certification Authority - G5
↳ Symantec Class 3 EV SSL CA - G3
↳ www.security.hsbc.co.uk

www.security.hsbc.co.uk

Issued by: Symantec Class 3 EV SSL CA - G3
Expires: Thursday, 17 May 2018 00:59:59 British Summer Time
This certificate is valid

Signature Algorithm	SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters	none
Not Valid Before	Monday, 16 May 2016 01:00:00 British Summer Time
Not Valid After	Thursday, 17 May 2018 00:59:59 British Summer Time
Public Key Info	
Algorithm	RSA Encryption (1.2.840.113549.1.1.1)
Parameters	none
Public Key	256 bytes : C4 96 ED 4E 55 B3 31 8B ...
Exponent	65537
Key Size	2048 bits
Key Usage	Encrypt, Verify, Wrap, Derive
Signature	256 bytes : 52 1C D0 F8 F9 35 BE 55 ...