Wireless Pers Commun (2016) 90:381–398
DOI 10.1007/s11277-016-3374-x

Product Authentication Using QR Codes: A Mobile Application to Combat Counterfeiting

M. Bala Krishna[1] · **Arpit Dugar**[1]

Published online: 1 August 2016

Abstract Counterfeiting is one of the biggest challenges for the authenticity of genuine products. An estimated average of 8–9 % trade consists of counterfeit goods that create a loss of revenue. To combat this situation, the product manufacturers use hologram and barcodes. The issue of genuine product remains the primary challenge in the market. With emerging trends in mobile and wireless technology, Quick Response (QR) codes provide a robust technique to fight the practice of counterfeiting the products. Apart from being used extensively in marketing and information transfer applications, the QR codes and encrypted QR codes are primarily used in security and privacy applications. Many web applications use QR codes for secure login where the user need not remember his/her login ID and password. The encrypted unique user ID is verified at the server using QR codes. Our proposed approach uses QR codes based on 2-dimensional codes (such as 19 Aztec, Data Matrix, etc.) to authenticate the product. This approach simplifies the size of QR code, and minimizes the complexity of encoding and decoding in QR code.

Keywords QR codes · Security · DES · Mobile application · Cryptography · Web-authenticity · QR-database · QR version · Error correction level

1 Introduction

Quick Response (QR) codes [1, 2] are the advanced version of traditional barcodes. 1-Dimensional barcode comprises of 10–20 characters of information. For more information, a 2-Dimensional code known as QR code is used. The main difference is barcode represents the

✉ M. Bala Krishna
mbalakrishna@ipu.ac.in

Arpit Dugar
arpit.dgr@gmail.com

[1] University School of Information and Communication Technology, Guru Gobind Singh Indraprastha University, New Delhi, India

Fig. 1 QR code

data in horizontal axis, and QR code represents the data in horizontal axis and vertical axis. Figure 1 gives an illustration of QR code. The salient features of QR code are as follows:

- *High Capacity Data Encoding*: More capacity as compared to 1-D barcodes, and the data is stored in two dimensions
- *Small Printing Size*: More data in compact form is displayed in small square print area
- *Dirt and Damage Resistant*: Error correction technique such as Reed–Solomon codes ensures the reliability of QR codes
- *Easy Decoding*: QR code is read from a view of 360° and decoded using mobile phone application. This ease of decoding has led to widespread applications of QR codes in businesses and secure applications

The augmented reality application of QR codes [3] extracts the information in QR codes and represents it in 3D format. The capacity of QR code is the amount of information stored and is dependent on the following factors:

- Version of QR code
- Encoded data type
- Error correction level

There are 40 versions of QR codes, and the capacity of QR codes increases with the version number. There are four data types supported by QR codes: numeric, alphanumeric, kanji and byte. Reed–Solomon algorithm concatenates zeroes at the end of polynomial and the data words and equated to the highest power present in the generator polynomial. The decimal representation of the remainder is used to define the error correcting code words. The error correction code using Reed–Solomon [4, 5] recovers the partially lost data in QR code.

1.1 Encryption: Data Encryption Standard

Data Encryption Standard known as DES is a symmetric encryption algorithm that uses the same key for encoding and decoding. DES as a block cipher divides the data into blocks of 64 bits, and the encryption algorithm is applied to each block simultaneously. The proposed approach uses DES to encode the product ID. For more secure applications, AES is used in QR codes. Figure 2 illustrates the working of DES algorithm.

1.1.1 DES block size: 64 bits

Key size: Though the key length is 64 bits, last bit of every byte is a parity bit and is not used for encryption. So, the key size is 56 bits.

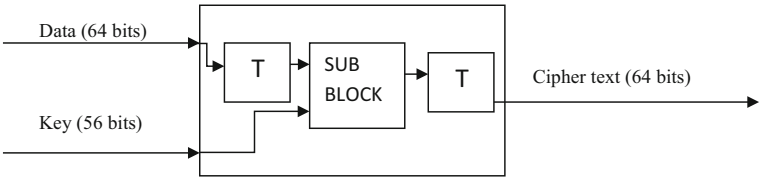

Fig. 2 Working of DES Algorithm

Input: Data bits 64-bit block, key 56 bits
Output: Cipher text 64 bits

Here Block T is the transposition block where the bit order is varied. The sub-block consists of a series of 16 rounds with each round comprising of substitution and permutation. In each of 16 rounds, shifting, permutation and exclusive-OR operations are performed for the key and data bits. Substitutions and permutations are always done with reference to the look-up-table.

Example:

Sample data: "YOU12345"
Key used: 0E329232EA6D0D73
Encrypted data: Z+7XwGEP508=

2 Related Work

Pre-processing low-quality QR codes [6] is based on blurred and wrapped images that use intrinsic approach and stored database to restore the quality of QR code. QR codes in high-resolution images [7] based on matrix code detection and Hough's recognition method uses pre-detection approach and reduces the complexity of QR codes. Identity Document Authentication with Visual Secret Sharing (VSS) and QR codes [8] considers a secret image that is encoded in two different images and further encoded into two distinct QR codes. The first part is printed on the ID card, and the second part is stored in the database. To check a genuine ID card, the printed QR code is scanned and verified with the existing database file. The main limitation of this approach is that the QR code printed on ID card can be copied and misused. QR codes provide enhanced digital services [9] used in various departments of government services, QR codes used in National Park Service store the web links and interactive pages of location coordinates that help to find the valid path in the park. This approach can be extrapolated and scaled to provide enhanced digital services. The main limitation of this approach is the phishing attack. SafeQR (based on two level security related to API's—Google safe browsing API) and Phishtank API [10] ultimately enhance the malicious URL and malware detection in the system. Both these API's enable the user to consistently check the URL in updated database of malicious URLs. This method first verifies the URL in Google Safe browsing list and then in the Phistank list to validate the genuine URL. QR codes are used to keep track of beverage consumption [11], the QR labels comprising of QR codes are allocated to each employee with unique ID. For buying any beverage like coffee, tea, etc. two QR codes are scanned: (i) employee QR code and (ii) product QR code. With the confirmation of matched database, coffee is dispensed from the machine, amount is deducted from the user account, and the corresponding e-mail is sent to the employee. Embedding barcode to perform phishing attacks [12] uses 2D barcodes such as

Aztec, Data Matrix. Multi-level barcode creates ambiguity and is used in non-secure applications. Geo-location based QR code [13] authenticates the system using session ID, domain name and web server URL. The scanned QR code generates the location of end user, and if there is ambiguity in location authentication, then the system is vulnerable to active real-time man in the middle (ART MITM) phishing attack. A visual QR code known as halftone QR code [14] performs minimal binding with QR code and adapts to halftone images. Challenges and risks associated with QR code security [15] are as follows: (i) QR codes are replaced by malicious code that directs the user to phishing site (ii) intruders modify sections of QR code to generate the fake product. Secure QR code in ecosystem includes digital signature, anti-phishing tools and malicious URL detection. QR codes and visual cryptography based E-voting authentication system [16] logs the voting system by using a unique voter password encoded with QR code. The QR image is encrypted using VSS scheme that creates two parts of the code, one part is given to the user and the other part is stored in the database. The encrypted image is mapped using unique RSA key pairs that are assigned to each voter ID. The public key is given to the voter, and the private key is stored in the database. During the voting procedure, the voter scans the QR code with the assigned public key. A successful match indicates that the authenticated voter can cast the vote. Two major attacks associated with QR codes are phishing (where the URL in QR code directs the user to a fake website binding the user with login ID and password window) and spreading malware. QR code for automated assessment [17] generates a valid task and prevents the user from invalid task. This approach uses randomized input, feedback, and transfer functions. Analysis of QR codes used in the wild environment [18] gives the frequency of user interactions and recognizes the misuse of QR codes.

3 Proposed Product Authentication using QR Codes: A Mobile Application to Combat Counterfeiting

To counterfeit a product, the original QR product code generated by the manufacturer is considered. If the counterfeiting practitioner tries to copy the QR information of an existing product, or if the same QR code is scanned again, then the product authentication system using the QR code displays the following message: "QR code was scanned earlier" or "The product is not genuine". Hence, to solve the problem of counterfeiting of genuine products, the product IDs are encrypted which further adds an additional layer of security in QR coding method. Product IDs are passed as query string parameters to the link authentication page, and the encryption will hide the product IDs from the customer. This method supports shorter product IDs and reuses them after a considerable amount of time.

3.1 Working Mechanism

- The proposed application needs login access to facilitate authentic products
- After login to the company, authorized personnel enters the product details such as product ID, product name, category description, batch number, price, manufactured date and is_verified. The product ID is unique and stored in the database. The attribute 'is_verified' is initially set to NULL for all the products registered in the database.
- The product ID is encrypted using symmetric key DES algorithm
- The product URL directs the user to product web page with encrypted product ID (as query string parameter). This URL is encoded into the QR code and printed on the product package.

- The QR code is covered with a plastic seal. When a customer buys a product, the plastic seal is removed and the QR code is scanned for product authenticity. This process directs the customer to URL web page of the product.
- The encrypted product ID passed as query string parameter is decrypted using the same key and further authenticated with the database. If the product ID is found, it is marked as 'verified' in the database (the product is genuine), else the message "Not Found" is displayed.
- The product is identified as a counterfeit product for the following conditions: (i) if the scanned product ID is not found in the database and (ii) the scanned product ID is already verified (that is the QR code is a copy of original product ID). In the proposed secure approach, the QR codes can be scanned only once, and this method ensures genuine products in the market.

QR code comprises of four levels of error correction schemes [19] such as: L (LOW), M (MEDIUM), Q (QUARTILE) and H (HIGH). The number of error correction codewords increases from Level L to Level H. Data restoration rate for QR code error correction [20, 21] is given as follows: L [LOW level] : 7 %, M [MEDIUM level] : 15 %, Q [QUARTILE level] : 25 %, H [HIGH level] : 30 %.

3.2 Error Correction Level

The number of data words accommodated in QR code version decreases from level L to level H, that is more space will be needed to accommodate the information when encoded in level H as compared to level L.

For example: Message to be encoded is www.ankitdugarart.com

Clearly, QR codes encoded using level Q and level H require more space (approx 39 % increase in area) as compared to QR codes encoded using level L and level M. Based on product size and secure level the corresponding QR code is selected. The error correction level is selected based on the operational environment and the space available for printing QR code. QR code used in harsh conditions such as the factory or industrial environments use either level Q or level H error correction since the impairment probability of QR code is high.

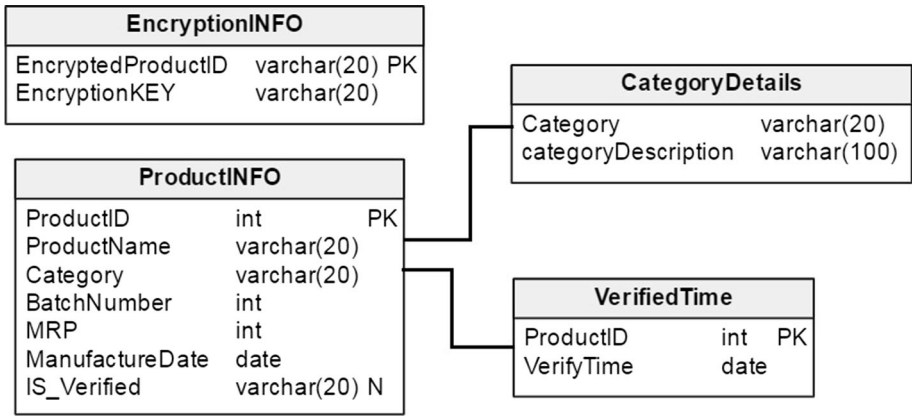

Fig. 3 Database Schema

3.3 Database Scheme

Database comprises of four tables as illustrated in Fig. 3. The attributes of each table are given as follows:

(i) **EncryptionINFO**

Attributes: EncryptedProductID, EncryptionKEY
Encrypted ProductsID with EncryptionKey is used to encrypt the ProductID and the same key is used to decrypt the ProductID

(ii) **ProductINFO**

Attributes: ProductID, ProductName, Category, BatchNumber, MRP, ManufactureDate, is_verified
ProductINFO attributes identify the product, and is_verified attribute indicates scanned status of the product

(iii) **Categorydetails**

Attributes: Category, CategoryDescription
Description for each category of ProductINFO

(iv) **VerifiedTime**

Attributes: ProductID, VerifyTime
Date and time of ProductID that was scanned for the first time

3.4 Algorithms for Product Authentication Using QR Codes

Algorithm 1. QR Code Generation for a Product

Input : Alphanumeric: *Username, Password;*
Integer: *Product ID;*
Alphanumeric *Product Name*, *Category*, *Batch Number*, *MRP;*
DateType *Date;* Char: *Error Correction Level;*
Output : JPEG: *QR Code Image*
Start
Step 1. Enter Login details
Step 2. If login Credentials = = True Then
Continue
Else
Go to step 1
End If
Step 3. Enter Unique Product ID of the Item
Step 4. Enter other details such as Product Name, Category, MRP, Batch Number
Step 5. Select the Error Correction Level (L, M, Q or H)
Step 6. Encrypt the Product ID using DES algorithm
Step 7. Add the Encrypted product ID as a query string parameter to the company's official product authentication webpage
Step 8. Encode the resultant link into the QR code image
End

Algorithm 2. QR Product Verification and Validation

Input: JPEG: *QR Code Image*
Output: Text: *Product Authentication Information*
Start
Step 1. Scan the QR CODE using a Mobile Phone App
Step 2. Open the link contained in QR code
Step 3. Input the encrypted Product ID passed as a query string
Step 4. Decrypt the Product ID with the same key (as encryption) and check product ID in the database
Step 5. If ProductINFO.ProductID in Database = = True Then
Step 6. If ProductINFO.is_verified = = NULL Then
Step 7. Populate the 'is_verified' of ProductID as 'Verified'
Step 8. Print the message "THE PRODUCT IS GENUINE"
Else
Step 9. Print the message "THE PRODUCT IS NOT GENUINE"
End If
Else
Step 10. Print the message "THE PRODUCT IS NOT GENUINE"
End If
End

4 Simulation Results

Simulation is performed using .NET and C# with Visual Studio Software and MS SQL Server as the database. We have provided an illustration for the Art Company known as *ankitdugarart.com*. Every painting is provided with a unique ID, and the authenticity is checked by scanning the QR code.

Simulation parameters

Software used:	Visual Studio 2013 Ultimate
Platform:	Microsoft.Net
Programming language:	C#
Library used:	Open Source QRCode Library
DES key size:	56 bits (64 bits including parity)
Error correction levels:	L, M, Q, H
Mobile platform used:	Android 4.4.2
Android app used:	QR code reader
Database primary key:	Product ID
Webpage URL:	http://www.ankitdugarart.com/Authenticity_Chk.html (The web page prompts an alert saying "Invalid or Null Product ID", since, the Product ID is not passed as string parameter (to check the authenticity))

(i) **Text encoded in QR code**

http://www.ankitdugarart.com/Authenticity_Chk.html?id=luivH%20jsuek

Whenever a user scans that QR code, the webpage display the message for authenticity check as: "THIS PRODUCT IS GENUINE" or "THIS PRODUCT IS NOT GENUINE". Here, since the QR code is scanned more than once, the web page displays the message: "THIS PRODUCT IS NOT GENUINE".

Number of characters: 66 in decimal; 01000010 in binary
Encoding mode: 0010 Alphanumeric

(a) **Alphanumeric equivalent**

794 1330 2023 1967 1472 1900 1055 839 615 730 1225 1244 1902 1102 1945 1379 779 1064 822 839 1566 557 942 794 1011 945 630 975 841 801 883 1364 900

(b) **Alphanumeric binary equivalent with added encoding mode and character count**

0010 001000010 1100011010 10100110010 11111100111 11110101111 10111000000 11101101100 10000011111 1101000111 1001100111 1011011010 10011001001 10011011100 011101101110 10001001110 11110011001 10101100011 1100001011 10000101000 1100110110 1101000111 11000011110 1000101101 1110101110 1100011010 1111110011 1010001010 1001110110 1111001111 1101001001 1100100001 1101110011 10101010100 1110000100

(c) **Resultant string divided into block of 8 bits**

0010 0010 00010110 0011010 1 01001100 10 111111 00111 111 10101111 10111000 00011101 10110010 00001111 1 1101000 11110011 00111101 10110101 00110010 01100110 11100011 10110111 01000100 11101111 00110011 01011000 11110000 10111000 0101000 1 10011011 01101000 11111000 01111010 00101101 11101011 10110001 10101111 11001110 10001010 10011101 10111100 11111101 00100111 00100001 11011100 11101010 10100111 00001000

(d) **Decimal equivalent**

34 22 53 76 191 63 175 184 29 178 15 232 243 61 181 50 102 227 183 68 239 51 88 240 184 81 155 104 248 122 45 235 177 175 206 138 157 188 253 39 33 220 234 167 8

The error correction code words are then appended to the final string based on error correction level (L, M, Q, H). The results of four possible error correction levels are shown as follows:

(ii) Error correction levels

Result 1:

Error correction	L
Version	4
Number of data words	66
Number of data blocks	1
Number of error correction words	20
QR code size	1.93 sq inch

Result 2:

Error correction:	M
Version	5
Number of data words	66
Number of data blocks	2
Number of error correction words	24 per block
QR code size	2.4 sq inch

Result 3:

Error correction	Q
Version	6
Number of data words	66
Number of data blocks	4
Number of error correction words	24 per block
QR code size	2.95 sq inch

Result 4:

Error correction	H
Version	7
Number of data words	66
Number of data blocks	4
Number of error correction words	24 per block
QR code size	4.2 sq inch

(iii) Screen outputs

Figure 4 illustrates the creation of unique QR code for the selected product and attributes such as product ID, product name, category, batch number, MRP, date of manufacture and error correction level. This image shows an example of creating a unique QR code for a painting being sold by ART Company. Figure 4 also show the resultant QR code generated.

Figure 5 shows the information contained in the QR code. This process succeeds the previous step where the administrator successively enters the product attributes. QR code as shown in Fig. 5 directs the user to product website with encrypted product ID passed as the query string operator.

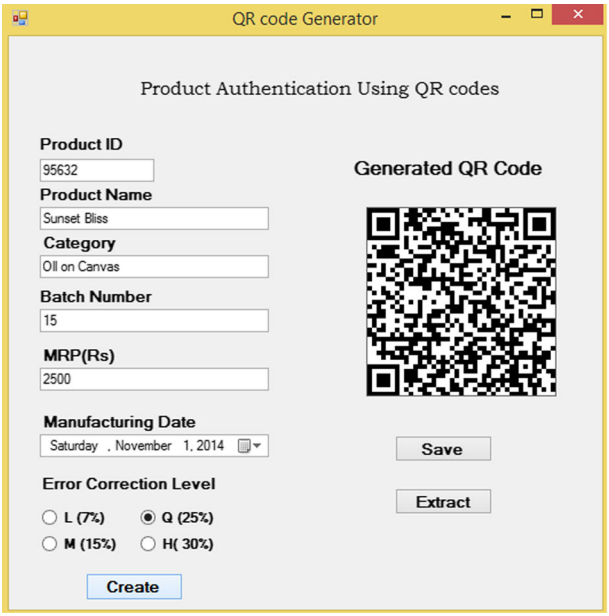

Fig. 4 Unique QR code generated for a product

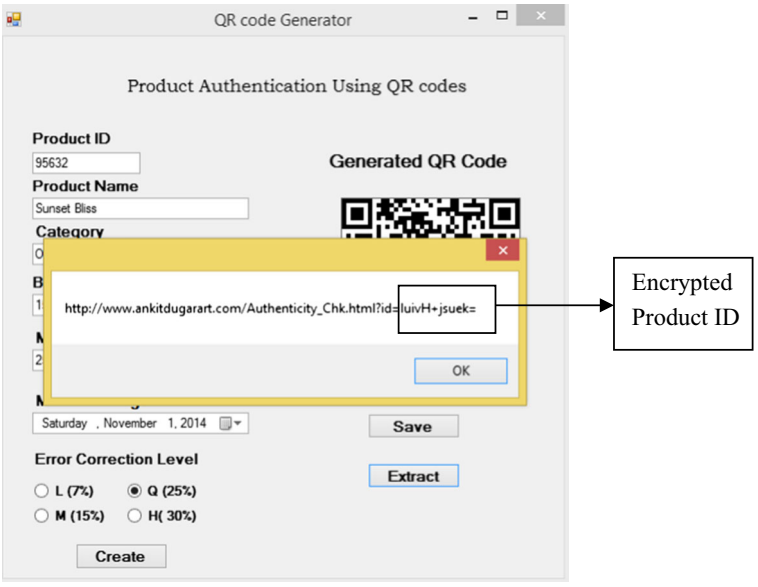

Fig. 5 Extraction of QR code information

The product ID is encrypted as follows:

Product ID used here:	95632
Encryption algorithm:	DES
DES encryption Key:	0E329232EA6D0D73
Encryption mode:	CBC
Encrypted text:	luivH+jsuek=

Figure 6a illustrates the scanning of QR code using an Android phone with QR Reader application, and Fig. 6b illustrates the URL displayed on the mobile phone and directs the user to authentic product website: http://www.ankitdugarart.com/Authenticity_Chk.html. The product information is retrieved from the database, and the status of the product (genuine or fake) is verified.

Figure 7 illustrates the URL opened by the mobile application. The product is identified as genuine when the QR code is scanned for the first time, and the corresponding product information is displayed. The authentic QR code used in our approach is only one-time scan, which indicates that a product cannot be scanned more than once. Hence, the product is scanned after buying the item, and verified with the manufacturer database for product authentication.

Figure 8 illustrates the scanning of QR code for the second time. In the proposed approach, the QR codes are one-time scan. Hence, this feature as shown in Fig. 8a illustrates that the QR code is copied from the genuine product or the product is not authentic. Figure 8a opens the URL page and displays the message: "This Product is Not Genuine".

(iv) **Database tables and SQL queries**

Table 1 illustrates the product database with corresponding attributes where QR code is generated for each product. The last field 'is_verified' is set to NULL. This feature

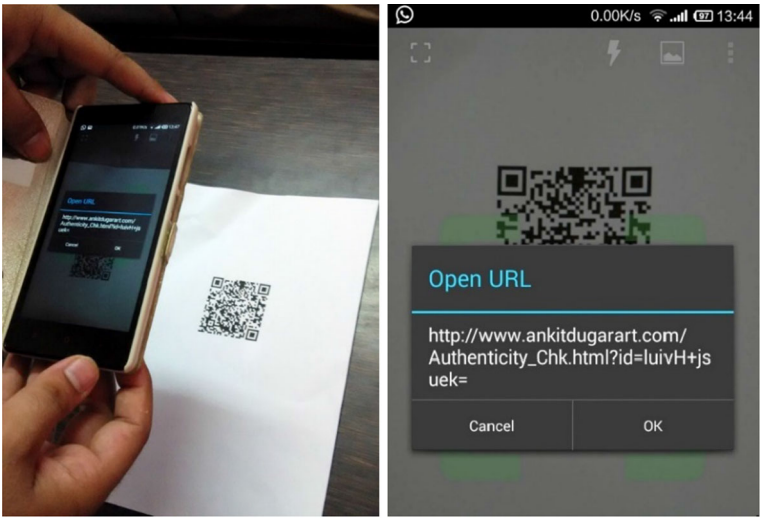

Fig. 6 Scanning of QR code using mobile application. **a** Mobile application scanning the QR code. **b** URL page of the corresponding QR code

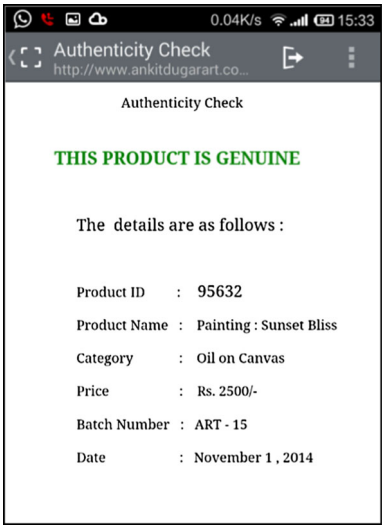

Fig. 7 QR code scanned for the first time

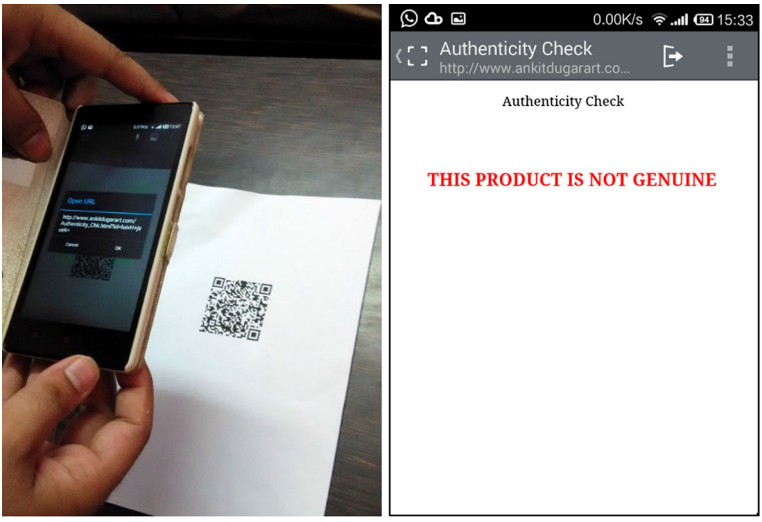

Fig. 8 Scanning of QR code for the second time. **a** Mobile application scanning the QR code (second time). **b** URL page opened with the corresponding QR code

illustrates that the product (one-time scan) is not verified. Table 2 illustrates the database table of EncryptionINFO, and Table 3 illustrates the database table of CategoryDetails

Table 1 Database table of ProductINFO before QR code was scanned

	ProductID	ProductName	Category	BatchNumber	MRP	ManufactureDate	is_verified
1	95630	Fort	Acrylics	15	2000	2014-12-08	NULL
2	95631	Heisenberg	Charcoal	15	3000	2014-11-18	NULL
3	95632	Sunset Bliss	Oil on Canvas	15	2500	2014-11-01	NULL
4	95633	Humming Bird	Oil on Canvas	14	1500	2014-10-11	NULL
5	95634	Light Years apart	Oil on Canvas	14	4500	2014-09-17	NULL

Table 2 Database table of EncryptionINFO

	EncryptedProductID	EncryptionKEY
1	H7MJ0fvSa2c=	0E329232EA6D0D73
2	Kjm574o+/R8=	0E329232EA6D0D73
3	luivH+jsuek=	0E329232EA6D0D73
4	ZkPhnlZgLul=	0E329232EA6D0D73
5	kMPHvLwP0TU=	0E329232EA6D0D73

Table 3 Database table of CategoryDetails

	Category	categoryDescription
1	Acrylics	Bright Acrylics color on paper.
2	Charcoal	Smudged Soft Charcoal on Matte paper
3	Oil On Canvas	HIgh Quality Oil paints on Canvas cloth

(a) **SQL queries for product authentication using QR codes**

(i) When the product QR code is scanned, the Encrypted ProductID is passed as query string operator in the URL (here, '*x*'). This is further used to query the 'EncryptionINFO' table to retrieve the Encryption key. The SQL query is given as follows:

Select EncryptionKEY
From EncryptionINFO
Where EncryptedProductID = *x*

In the example shown, x = luivH+jsuek=

(ii) The Encrypted ProductID (here, *luivH+jsuek*=) can be decrypted using the encryption key fetched from the previous query to generate the original Product ID (PID), here PID is 95632.

(iii) The generated PID is used to query the ProductINFO table to find the value of 'is_verified' attribute. The SQL query to fetch 'is_verified' attribute is given as follows:

```
Select is_verified
From ProductINFO
Where ProductID = PID
```

(iv) 'is_verified' attribute can be either NULL or Verified. If the 'is_verified' attribute is NULL, then the product is genuine and a message is sent to the end user. After scanning the QR code, 'is_verified' attribute is set to Verified. Table 4 illustrates that the QR code is VERIFIED for the product in the third row. The SQL query to update the ProductINFO table is given as follows:

```
Update ProductINFO
set is_verified='Verified'
where ProductID=95632
```

(v) The VerifiedTime table given in Table 5 is updated with date and scan time. The SQL query of updated VerifiedTime table is given as follows:

```
Update VerifiedTime
set VerifyTime=getdate()
where ProductID=95632
```

(vi) If the 'is_verified' attribute is enabled with Verified (which means the product was scanned before), then the product will be identified as "Not Genuine" and the message will be sent to end user.

(vii) SQL queries used by administrator to display the product database before QR code was scanned (as shown in Table 6), fetch the verified products after QR code was scanned (as shown in Table 7), and fetch the unverified products (that is unscanned QR codes as shown in Table 8) is given as follows:

Table 4 Database table of ProductINFO table after the QR code was scanned

	ProductID	ProductName	Category	BatchNumber	MRP	ManufactureDate	is_verified
1	95630	Fort	Acrylics	15	2000	2014-12-08	NULL
2	95631	Heisenberg	Charcoal	15	3000	2014-11-18	NULL
3	95632	Sunset Bliss	Oil on Canvas	15	2500	2014-11-01	Verified
4	95633	Humming Bird	Oil on Canvas	14	1500	2014-10-11	NULL
5	95634	Light Years apart	Oil on Canvas	14	4500	2014-09-17	NULL

Table 5 Database table of VerifiedTime after QR code verification

	ProductID	VerifyTime
1	95632	2016-03-30 14:39:57.330

Table 6 Database of each product before QR code was scanned

	ProductID	ProductName	categoryDescription	BatchNumber	MRP	ManufacturedDate	is_verified
1	95630	Fort	Bright Acrylics color on paper.	15	2000	2014-12-08	NULL
2	95631	Heisenberg	Smudged Soft Charcoal on Matte paper	15	3000	2014-11-18	NULL
3	95632	Sunset Bliss	HIgh Quality Oil paints on Canvas cloth	15	2500	2014-11-01	NULL
4	95633	Humming Bird	HIgh Quality Oil paints on Canvas cloth	14	1500	2014-10-11	NULL
5	95634	Light Years Apart	HIgh Quality Oil paints on Canvas cloth	14	4500	2014-09-17	NULL

Table 7 Database table of verified products (scanned QR code)

	ProductID	ProductName	categoryDescription	BatchNumber	MRP	ManufacturedDate	is_verified
1	95632	Sunset Bliss	HIgh Quality Oil paints on Canvas cloth	15	2500	2014-11-01	verified

Table 8 Database table of unverified products (unscanned QR codes)

	ProductID	ProductName	categoryDescription	BatchNumber	MRP	ManufacturedDate	is_verified
1	95630	Fort	Bright Acrylics color on paper.	15	2000	2014-12-08	NULL
2	95631	Heisenberg	Smudged Soft Charcoal on Matte paper	15	3000	2014-11-18	NULL
3	95633	Humming Bird	HIgh Quality Oil paints on Canvas cloth	14	1500	2014-10-11	NULL
4	95634	Light Years Apart	HIgh Quality Oil paints on Canvas cloth	14	4500	2014-09-17	NULL

- SQL query to fetch all product details is given as follows:

  Select ProductID, ProductName, c.categoryDescription, BatchNumber, MRP, ManufacturedDate, is_verified
  from ProductINFO p join CategoryDetails c on p.Category=c.Category

  where p and c are aliases for ProductINFO table and CategoryDetails table respectively.

- SQL query to fetch all verified products details is given as follows:

  Select ProductID, ProductName, c.categoryDescription, BatchNumber, MRP, ManufacturedDate, is_verified
  from ProductINFO p join CategoryDetails c on p.Category=c.Category
  where is_verified ='Verified'

  where p and c are aliases for ProductINFO table and CategoryDetails table respectively.

- SQL query to fetch all unverified product details is given as follows:

  Select ProductID, ProductName, c.categoryDescription, BatchNumber, MRP, ManufacturedDate, is_verified
  from ProductINFO p join CategoryDetails c on p.Category=c.Category
  where is_verified = NULL

5 Conclusions

QR codes are extensively used to identify the product ID. The proposed product authentication for QR codes combats counterfeiting of products and identify product genuineness. The proposed QR authentication generates the QR code based on product attributes and directs the user to company web page indicating whether the product is genuine or fake. Comparative analysis of error corrections in QR code indicates the relationship between the size of QR codes, data capacity and levels of QR code. This significant approach combats the problem of counterfeit products and benefits the customer and product manufacturers. This work can be extended to include company specific QR code reader that provide two level authentication such as (i) offline authenticity (mobile application scans the QR code), online authenticity (the scanned QR code directs the user to product manufacturer's web page) and (ii) update the list of items purchased and scanned at the manufacturer web site. Further extension would be to include the time stamp approach in encrypted QR codes.

References

1. Shin, D.-H., Jung, J., & Chang, B.-H. (2012). The psychology behind QR codes: User experience perspective. *Elsevier Journal of Computers in Human Behavior, 28*, 1417–1426.
2. Schultz, M. K. (2013). A case study on the appropriateness of using quick response (QR) codes in libraries and museums. *Elsevier Journal of Library and Information Science Research, 35*, 207–215.
3. Kan, T.-W., Teng, C.-H., & Chou, W.-S. (2009). Applying QR code in augmented reality applications. In *Proceedings of ACM international conference virtual reality continuum and its applications in industry (VRCAI)*, Yokohama, Japan, December 14–15, 2009, pp. 253–257.
4. Moon, T. K. (2005). *Error correction coding—Mathematical methods and algorithms*. Hoboken, NJ: Wiley.
5. Morelos-Zaragoza, R. H. (2006). *The art of error correcting coding* (2nd ed.). Chichester, West Sussex: Wiley.
6. Muñoz-Mejías, D., González-Díaz, I., & Díaz-de-María, F. (2011). A low-complexity pre-processing system for restoring low-quality QR code images. *IEEE Transactions on Consumer Electronics, 57*(3), 1320–1328.
7. Szentandrási, I., Herout, A., & Dubská, M. (2012). Fast detection and recognition of QR codes in high-resolution images. In *Proceedings of ACM 28th international spring conference on computer graphics (SCCG)*, Budmerice, Slovakia, May 2–4, 2012, pp. 129–136.
8. Espejel-Trujillo, A., Castillo-Camacho, I., Nakano-Miyatake, M., & Perez-Meana, H. (2012). Identity document authentication based on VSS and QR codes. In *Proceedings of Elsevier Iberoamerican conference on electronics engineering and computer science (CIIECC), procedia technology 3*, Guadalajara, Mexico, May 16–18, 2012, pp. 241–250.
9. Lorenzi, D., Vaidya, J., Chun, S., Shafiq, B., et al. (2012). Using QR codes for enhancing the scope of digital government services. In *Proceedings of ACM thirteenth annual international conference on digital government research (DGO)*, College Park, MD, USA, June 4–7, 2012, pp. 21–29.
10. Yao, H., & Shin, D. (2013). Towards preventing QR code based attacks on android phone using security warnings. In *Proceedings of ACM eighth SIGSAC international symposium on information, computer and communications security (ASIACCS)*, Hangzhou, China, May 8–10, 2013, pp. 341–346.
11. Maurer, M.-E., Luca, A. D., Hang, A., Hausen, D., et al. (2013) Long-term experiences with an iterative design of a QR-code-based payment system for beverages. In *Proceedings of Springer human–computer interaction (INTERACT), LNCS vol. 8120, held with 14th IFIP international conference TC, part IV*, Cape Town, South Africa. September 2–6, 2013, pp. 587–594.
12. Dabrowski, A., Krombholz, K., Ullrich, J. & Weippl, E. R. (2014) QR inception: Barcode-in-barcode attacks. In *Proceedings of ACM fourht international workshop on security and privacy in smartphones & mobile devices (SPSM)*, Scottsdale, Arizona, USA, November 7, 2014, pp. 3–10.

13. Kim, S.-H., Choi, D., Jin, S.-H., & Lee, S.-H. (2013). Geo-location based QR-code authentication scheme to defeat active real-time phishing attack. In *Proceedings of ACM international workshop on digital identity management (DIM)*, Berlin, Germany, November 8, 2013, pp. 51–62.
14. Chu, H.-K., Chang, C.-S., Lee, R.-R., & Mitra, N. J. (2013). Halftone QR codes. *ACM Transactions on Graphics, 32*(6), 217:1–217:8.
15. Krombholz, K., Fruhwirt, P., Kieseberg, P., Kapsalis, I., et al. (2014). QR code security: A survey of attacks and challenges for usable security. In *Proceedings of Springer second international conference on human aspects of information security, privacy, and trust, LNCS Vol. 8533, held with HCI international*, Heraklion, Crete, Greece, June 22–27, 2014, pp. 79–90.
16. Falkner, S., Kieseberg, P., Simos, D. E., Traxler, C., et al. (2014). E-voting authentication with QR-codes. In *Proceedings of Springer second international conference on human aspects of information security, privacy, and trust, LNCS Vol. 8533, held with HCI international*, Heraklion, Crete, Greece, June 22–27, 2014, pp. 149–159.
17. Hakulinen, L., & Malmi, L. (2014). QR code programming tasks with automated assessment. In *Proceedings of ACM 19th annual conference on innovation and technology in computer science education (ITiCSE)*, Uppsala, Sweden, June 23–25, 2014, pp. 177–182.
18. Lerner, A., Saxena, A., Ouimet, K., Turley, B., et al. (2015). Analyzing the use of quick response codes in the wild. In *Proceedings of ACM 13th annual international conference on mobile systems, applications, and services (MobiSys)*, Florence, Italy, May 18–22, 2015, pp. 359–374.
19. DataGenetics (2013). QR Codes—How do they work, and how much can you damage them and still have them work? Wounded QR codes. http://datagenetics.com/blog/november12013/index.html, November 2013.
20. Denso Wave Incorporated (2014). QR Code.com, error correction features: For advanced users. http://www.qrcode.com/en/about/error_correction.html.
21. Eby, C. (2015). QR code tutorial—character capacities by version, mode, and error correction. http://www.thonky.com/qr-code-tutorial/character-capacities. Thonky.com's QR code tutorial under a creative commons attribution, May 15, 2015.

M. Bala Krishna received Bachelor of Engineering (B.E.) Degree in Computer Engineering from Delhi Institute of Technology (presently Netaji Subhash Institute of Technology), University of Delhi, Delhi, India and Master of Technology (M.Tech.) Degree in Information Technology from University School of Information Technology (presently University School of Information and Communication Technology), GGS Indraprastha University, Delhi, India. He had received Doctor of Philosophy (Ph.D.) Degree in Computer Engineering from JMI Central University, New Delhi, India. He had earlier worked as Senior Research Associate and Project Associate in Indian Institute Technology (IIT), Delhi, India in the areas of Digital Systems and Embedded Systems. He had worked as Faculty Member and had handled projects related to Networking and Communication. He is presently working as Assistant Professor in University School of Information and Communication Technology, GGS Indraprastha University, Delhi, India. His areas of interest include Computer Networks, Wireless Networking and Communications, Mobile and Ubiquitous Computing and Embedded System Design. He has publications in International Journals, Conferences, and Book Chapters. His teaching areas include Wireless Networks, Mobile Computing, Data and Computer Communications, Embedded Systems, Programming Languages, etc. His current research work includes Wireless Ad hoc and Sensor Networks, Advances in Mobile Computing and Communications, Cognitive Radio Networks, Software Defined Networks and Internet of Things. He is a member for IEEE and ACM Technical Societies. He is TPC member for IEEE and ACM International Conferences.

Arpit Dugar received Bachelor of Technology (B.Tech.) Degree in Electronics and Communication Engineering from University School of Information and Communication Technology, GGS Indraprastha University, Delhi in 2015. His areas of interest include computer networks, wireless networks and Voice Over IP. He is presently working as Technical Assistance Centre (TAC) Engineer in Cisco Systems (India) Pvt. Ltd, Bengaluru, India.