Authentication and Transaction Verification Using QR Codes with a Mobile Device

Yang-Wai Chow[1](✉), Willy Susilo[1], Guomin Yang[1],
Man Ho Au[2], and Cong Wang[3]

[1] School of Computing and Information Technology,
Centre for Computer and Information Security Research,
University of Wollongong, Wollongong, Australia
{caseyc,wsusilo,gyang}@uow.edu.au

[2] Department of Computing, The Hong Kong Polytechnic University,
Kowloon, Hong Kong
csallen@comp.polyu.edu.hk

[3] Department of Computer Science, City University Hong Kong,
Kowloon Tong, Hong Kong
congwang@cityu.edu.hk

Abstract. User authentication and the verification of online transactions that are performed on an untrusted computer or device is an important and challenging problem. This paper presents an approach to authentication and transaction verification using a trusted mobile device, equipped with a camera, in conjunction with QR codes. The mobile device does not require an active connection (e.g., Internet or cellular network), as the required information is obtained by the mobile device through its camera, i.e. solely via the visual channel. The proposed approach consists of an initial user authentication phase, which is followed by a transaction verification phase. The transaction verification phase provides a mechanism whereby important transactions have to be verified by both the user and the server. We describe the adversarial model to capture the possible attacks to the system. In addition, this paper analyzes the security of the propose scheme, and discusses the practical issues and mechanisms by which the scheme is able to circumvent a variety of security threats including password stealing, man-in-the-middle and man-in-the-browser attacks. We note that our technique is applicable to many practical applications ranging from standard user authentication implementations to protecting online banking transactions.

Keywords: Authentication · Mobile device · One-Time-Password (OTP) · QR code · Transaction-Authentication-Number (TAN) · Transaction integrity · Transaction verification

1 Introduction

User authentication and the verification of online transactions in Internet based services is an important issue that has received much attention by researchers and

G. Wang et al. (Eds.): SpaCCS 2016, LNCS 10066, pp. 437–451, 2016.
DOI: 10.1007/978-3-319-49148-6_36

practitioners alike. Addressing the security concern surrounding user authentication and online transactions is essential considering the extensive use of computers and electronic devices in our everyday life. Moreover, with the increasing number and variety of malicious threats such as phishing, Trojans, key-loggers, etc. many transactions are conducted on untrustworthy computers or devices.

In addition, not only are the conventional approaches to authentication, like the traditional username and password login approach, susceptible to password stealing attacks, the increasing number of online services means that a person either has to remember a large number of different passwords or compromise on the security by using the same password for multiple services. As such, over the years a large variety of different authentication schemes have been proposed and studied [2,3]. For example, a number of schemes have proposed the use of One-Time-Passwords (OTPs) to prevent attacks like key-logging and phishing [14], Short-Message-Service (SMS) based OTP schemes [29], as well as others like two, or three, factor authentication [8,13].

However, while these schemes are useful, they are not necessarily secure. For instance, SMS-based OTP schemes rely on the security of the cellular network. Mulliner et al. [22] have contented that SMS OTP schemes cannot be considered to be secure, as researchers have shown several successful attacks against Global System for Mobile Communications (GSM) and 3G networks [1,12,19]. Furthermore, it has been argued that two, or three, factor authentication does not overcome man-in-the-middle and Trojan attacks [10,26,27].

This paper investigates the challenging problem of user authentication and transaction verification on an untrusted computer or device. We define transaction verification as encompassing both transaction authentication (i.e. the transaction was indeed performed by the user) and transaction integrity (i.e. the transaction has not been altered). In this paper, we present an approach that uses a personal trusted mobile device, with the requirement that the mobile device has a camera. This is a reasonable requirement that does not overburden the user, as nowadays personal mobile devices are common place and many individuals already own and use personal mobile devices like smartphones every day. Moreover, in our approach the user does not have to remember any passwords, except for the passcode used to login to the mobile device. In fact, some devices allow for other login methods like biometrics.

Unlike a number of other camera-based mobile phone approaches [11,20,23,29], our approach *does not* require the mobile device to have an active connection (e.g., connection to the computer, cellular network, or Internet), all required information is obtained by the mobile device's camera via the visual channel using QR codes. As such, our approach does not suffer from lost of connection problems (e.g., losing Internet connection within a building, no roaming services when bringing the phone to another country, etc.) and does not require the user to establish a connection (e.g., Bluetooth) with, or install any special software on, the computer.

In principle, our approach does not specifically require the use of QR codes per se; any method of transferring the required data to the mobile device will suffice. We adopt the QR code approach as it is a convenient and widespread method of communication via the visual channel. One should also note that in our approach the device does not have to be a mobile phone; it can be any trusted mobile device with a camera, e.g., a tablet computer, or even a specialized security token.

Our Contributions. In this paper, we present the design of an authentication and verification approach for online transactions on untrusted devices, using a trusted camera-based mobile device in conjunction with QR codes. Our approach is separated into two phases; the user authentication phase and the transaction verification phase. In the user authentication phase, an OTP is obtained by the mobile device. The OTP is only valid for a single session, thus circumvent password stealing or replay attacks. After user authentication, important transactions are verified using a Transaction-Authentication-Number (TAN). The user can verify that the transaction information is accurate and the server can verify that the transaction came from the user. This is to prevent session hijacking attacks after user authentication. This paper analyzes the security and discusses the practical issues as well as the drawbacks of the proposed approach. The scheme described in this paper is applicable to many practical applications ranging from standard user authentication to protecting online banking transactions.

2 Related Work

Over the years, there has been a lot of work in the area of authentication and transaction verification. Researchers have proposed a variety of different schemes, that rely on diverse mechanisms to secure transactions. A number of key research efforts that are related to the scheme proposed in this paper are described in this section. We roughly organize them here into a number of categories; namely, SMS-based, token-based, connection-based, camera-based and QR code-based approaches. However, it should be noted that many of the approaches overlap and are not confined to a single category.

2.1 Authentication Methods

SMS-Based. Sun et al. [29] describe oPass, an SMS-based authentication method of using a cellphone. During the registration phase of this approach, a user registers an account ID and a phone number. The user will also setup a long-term password for generating a chain of OTPs for subsequent logins. To access an online service, the user enters his/her ID into an untrusted web browser. The user then opens the oPass program on his/her phone and enters the long-term password. The program will generate an OTP that is sent via SMS directly to the server, which verifies the user's identity based on the SMS. Similar approaches have also been proposed in other work [15] and there are numerous approaches where OTPs are sent to the user's cellphone via SMS [2,3].

Token-Based. The use of security tokens are another approach to authentication. Tokens can be in the form of a physical device like a smart card or even a mobile phone. For example, RSA SecurID [25] is a approach where a security token is used to generate authentication codes at certain time intervals based on an initial seed value. However, it has been highlighted that this approach does not defend against session hijacking or online phishing [11]. Li et al. [17] propose a low-cost hardware token based PIN/TAN system for protecting e-banking systems. This hardware takes the form of a physical USB token that has to be inserted into an untrusted computer to perform user/server/transaction authentication.

Connection-Based. MP-Auth [20] is an approach that relies on a trusted personal device to perform cryptographic computations. It requires a pre-established long-term password to be shared between the user and the server. To protect the password, password information is entered into the personal device instead of the untrusted terminal. For this to happen, MP-Auth needs a connection between the personal device and the computer, because cryptographic computations that are performed on the personal device are sent to the computer, which in turn forwards it to the server [20].

The Phoolproof phishing prevention system [23] is cellphone based approach that uses public key cryptography in conjunction with a username/password combination and an SSL connection. For the approach to work, a connection must be established between the trusted cellphone and the untrusted browser. A user who wishes to access an online account must always initiate the connection using a secure bookmark stored in the cellphone, and the cellphone will direct the browser to the associated URL [23].

Camera-Based. Clarke et al. [6] propose a camera-based authentication approach that requires a specialized device to perform constant monitoring of the user's interaction with an untrusted computer, by monitoring the information displayed on the computer's screen. The aim of the approach is to detect whether information displayed on the computer's screen has been tampered with. The required monitoring in this method can be rather computationally intensive.

Chow et al. [4] describe a visual OTP challenge-response authentication approach that is based on visual cryptography. The secret authentication message is split into two visual cryptography shares. Using the mobile device's camera, the user visual obtains the secret by overlaying one share on the mobile device's screen with the other share that is displayed on the computer's screen. Another method of authentication via the visual channel was demonstrated by McCune et al. [21]. Their approach is focused on authentication between two devices rather than an online service.

Cronto [7] is a commercial transaction authentication system for online banking transactions. The system uses a patented visual transaction signing approach in which a graphical cryptogram, a CrontoSign image, is displayed on a user's computer screen. The user uses a camera phone or a dedicated hardware device to capture the cryptogram. Transaction information can be securely decoded from the cryptogram if the image is untampered. The user is to check the transaction details and confirm that the transaction is genuine. An authentication code is then generated on the user's phone or device and the user has to pass this back to the bank's server to complete a transaction [7].

QR Code-Based. Snap2pass, and its extension Snap2pay, is a QR code-based approach which requires a camera cellphone to have an active connection [11]. To login to a website, the server sends a QR code, which encodes a cryptographic challenge, to the browser. The user is to take a picture of the QR code with his/her camera cellphone. After the cellphone performs the necessary cryptographic computations, it sends a cryptographic response directly back to the server. Upon receiving a valid response, the server then logs the user in through the browser.

Starnberger et al. [27] describe QR-TAN, a transaction authentication method based on QR codes and a trusted mobile device. In QR-TAN, a shared secret key must be pre-arranged between the mobile device and the server. In addition, it uses public key cryptography where the private key is stored in the mobile device and the untrusted computer has access to the public key. To perform a transaction, a nonce is requested from the server. The untrusted computer then encrypts transaction information and the nonce using the mobile device's public key and displayed the result as a QR code. The mobile device scans the QR code to obtain the encrypted information, which it is required to decrypt using its private key. It then computes a hash based on the transaction information, the nonce, and whether the user approves or rejects the transaction. Part of this hash is sent as a TAN to the server, which computes its own approve and reject hash values, and tries to match these with the value computed on the mobile device.

In addition to the approaches described above, there are various other proposed QR code authentication schemes [18,24,28,30].

2.2 The QR Code

The QR code is a two-dimensional code that was invented by the company Denso Wave [9]. Its widespread adoption in many different applications is due to its convenience and ease of use. Any device equipped with a camera and QR code reader can retrieve the information encoded within a QR code. Other than for authentication, QR codes have been used for a variety of security applications including secret sharing [5] and digital watermarking [16]. While our approach does not specifically require the use of QR codes, as any method that is able to pass data to the mobile device will suffice, we chose to adopt the QR code because of its intuitiveness.

3 System and Adversarial Models

In this section, we describe our system and adversarial models that will be used to analyze the security of our proposed scheme.

3.1 System Model

The system consists of the following entities: end users who are equipped with a mobile device that holds the user's long-term secret information such as passwords or cryptographic keys; a transaction server that the users will connect to for online transactions; and a public computer that will be used by the user to interact with the server via the Internet.

We assume that the mobile device is equipped with a camera but does not have any network connection (i.e., it is a stand alone device). The mobile device can only communicate with the computer via the visual channel using QR codes. The computer can connect to the transaction server through the Internet. In this paper, we assume that the computer is public such that any user, including malicious attackers, can access it and install any (possibly malicious) software on it. The transaction server will process any connection request from the Internet, including those initiated by malicious attackers.

Without losing generality, we also assume a Public Key Infrastructure (PKI) is in place where users and the transaction server can obtain Digital Certificates from a trusted Certification Authority (CA).

3.2 Adversarial Model

Based on the system model described above, we present our adversarial model. We assume that the transaction server and the mobile device of an honest user are trusted, which means attackers cannot access any secret information maintained by the mobile device or the transaction server. However, attackers can corrupt the public computer and install any software (e.g., key-logger) on it. In addition, an attacker can use the compromised computer to communicate with the mobile device via the visual channel and the transaction server through the Internet.

We also assume that an attacker, via the compromised computer, can access and record any user input from peripherals (such as keyboard, mouse, etc.) as well as any traffic generated in an online transaction between the user and the transaction server. Furthermore, during an online transaction, an attacker can modify the data exchanged between the user and the server through the compromised computer.

Security Goals. It is obvious that given an untrusted computer described above, we are unable to achieve security against certain attacks, such as the Denial of Service attack or the eavesdropping attack since we assume the attacker can directly control the network communication and monitor any transaction performed by the user. Therefore, in this paper we mainly focus on the security goals related to the integrity and authenticity of the online transactions. Specifically, we require the following security properties to be preserved with overwhelming probabilities.

- *User authentication.* We require that without the cooperation of the user (and the mobile device), an attacker who controls the compromised computer cannot successfully impersonate the user against the transaction server, given that the attacker can access all the previous user communication transcripts.
- *Transaction authentication.* Without the involvement of the user (and the mobile device), the attacker cannot successfully perform an online transaction with the server, given that the attacker can access all the previous user transactions.
- *Transaction integrity.* The attacker cannot modify any transaction data exchanged between the user and the transaction server without being detected.

4 The Proposed Scheme

The proposed approach addresses the problem of authenticating the user and verifying online transactions using two phases; namely, an initial user authentication phase and the transaction verification phase. During the user authentication phase the untrusted computer establishes an SSL connection with the server; information is exchanged between the server and the mobile device via the untrusted computer, and if certain conditions are met, the server will be able to authenticate that it is indeed communicating with the correct user.

After the user authentication phase, any important transactions made by the user will have to be verified by both the user and the server. This is to prevent any tampering by an attacker who manages to hijack the session after user authentication has already occurred, e.g., via a Trojan on the untrusted computer. Details of these two phases are described in the respective sections to follow.

4.1 User Authentication

During the user authentication phase, it is assumed that the user has already logged in to the mobile device and has opened the specific mobile application (app), which implements the proposed scheme. The mobile app will have access to the user's secret private key, k_{priv}, which is kept in a secure location on the mobile device. It is also assumed that given a user's identity, the server can obtain the user's public key from a PKI. Figure 1 presents an overview depicting the flow of required events during the user authentication phase.

The steps required for user authentication are described as follows. Let $\mathsf{PKE.Enc}(pk, \cdot)$ denote public key encryption[1] of the parameter with public key

[1] In practice, we need to employ a CCA-secure public key encryption as part of the protocol.

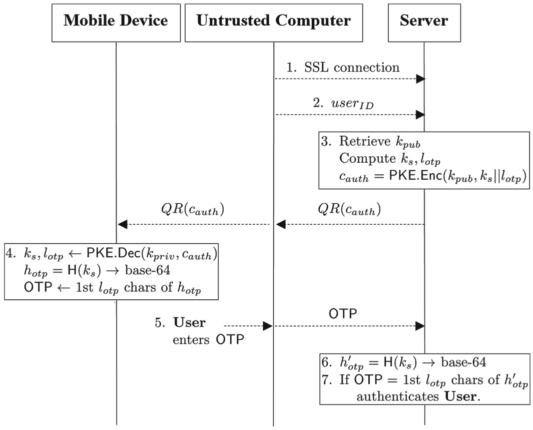
Mobile Device
Untrusted Computer
Server
1. SSL connection
2. $user_{ID}$
3. Retrieve k_{pub}
Compute k_s, l_{otp}
$c_{auth} = \mathsf{PKE.Enc}(k_{pub}, k_s||l_{otp})$
$QR(c_{auth})$
$QR(c_{auth})$
4. $k_s, l_{otp} \leftarrow \mathsf{PKE.Dec}(k_{priv}, c_{auth})$
$h_{otp} = \mathsf{H}(k_s) \rightarrow$ base-64
$\mathsf{OTP} \leftarrow$ 1st l_{otp} chars of h_{otp}
5. **User**
enters OTP
OTP
6. $h'_{otp} = \mathsf{H}(k_s) \rightarrow$ base-64
7. If OTP = 1st l_{otp} chars of h'_{otp}
authenticates **User**.

Fig. 1. Overview of the user authentication phase.

pk, and $\mathsf{PKE.Dec}(sk, \cdot)$ represent public key decryption of the parameter with private key sk. In addition, let H be a collision free one-way hash function, and OTP denote a one-time-password.

1. User, U, enters URL into Browser, B, on an untrusted computer; B establishes an SSL connection with the Server, S.
2. U enters user identity, $user_{ID}$, into B; B sends $user_{ID}$ to S.
3. S:
   - uses $user_{ID}$ to retrieve U's public key, k_{pub};
   - computes two random numbers; namely, a random session key, k_s, and a random length, l_{otp}, which will be used as the length of OTP;
   - uses k_{pub} to compute an authentication ciphertext, c_{auth}, where $c_{auth} = \mathsf{PKE.Enc}(k_{pub}, k_s||l_{otp})$
   - encodes c_{auth} into a QR code, $QR(c_{auth})$;
   - sends $QR(c_{auth})$ to B.
4. B displays $QR(c_{auth})$; U uses Mobile Device, M; M:
   - scans $QR(c_{auth})$ and decodes it to obtain c_{auth};
   - obtains k_s and l_{otp} using U's private key, k_{priv}, via $\mathsf{PKE.Dec}(k_{priv}, c_{auth})$;
   - computes $h_{otp} = \mathsf{H}(k_s)$;
   - converts h_{otp} into base-64 representation;
   - displays the first l_{otp} base-64 characters of h_{otp} to U. These characters will be used as OTP.
5. U enters the OTP into B; B sends this to S.
6. S:
   - computes $h'_{otp} = \mathsf{H}(k_s)$;
   - converts the h'_{otp} into base-64 representation;
   - compares the first l_{otp} base-64 characters of h'_{otp} with OTP.
7. If the two character sequences match, then S authenticates U with $user_{ID}$.

4.2 Transaction Verification

After the user authentication phase, the user with $user_{ID}$ would have been authenticated. In addition, the mobile device and the server would have established a session key, k_s. Figure 2 illustrates the steps required for transaction verification.

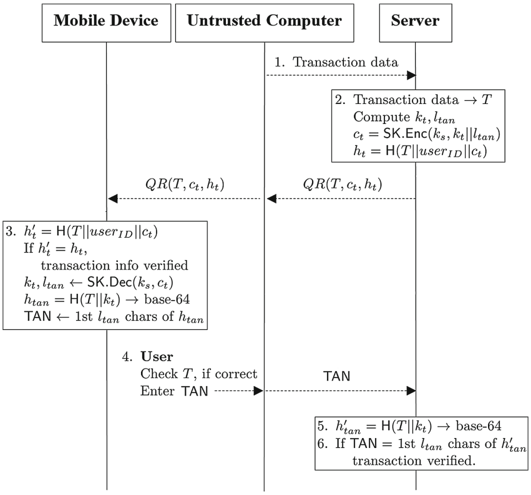
Mobile Device
Untrusted Computer
Server
1. Transaction data
2. Transaction data → T
Compute k_t, l_{tan}
$c_t = \mathsf{SK.Enc}(k_s, k_t||l_{tan})$
$h_t = \mathsf{H}(T||user_{ID}||c_t)$
$QR(T, c_t, h_t)$
$QR(T, c_t, h_t)$
3. $h'_t = \mathsf{H}(T||user_{ID}||c_t)$
If $h'_t = h_t$,
transaction info verified
$k_t, l_{tan} \leftarrow \mathsf{SK.Dec}(k_s, c_t)$
$h_{tan} = \mathsf{H}(T||k_t) \rightarrow$ base-64
TAN ← 1st l_{tan} chars of h_{tan}
4. **User**
Check T, if correct
Enter TAN
TAN
5. $h'_{tan} = \mathsf{H}(T||k_t) \rightarrow$ base-64
6. If TAN = 1st l_{tan} chars of h'_{tan}
transaction verified.

Fig. 2. Overview of the transaction verification steps.

The steps required for both the user and the server to verify the integrity of a transaction are described as follows. Let $\mathsf{SK.Enc}(k, \cdot)$ and $\mathsf{SK.Dec}(k, \cdot)$ denote symmetric key encryption and decryption, respectively, of the parameter with key k. Furthermore, let H represent a collision free one-way hash function, and TAN denote a transaction-authorization-number.

1. U confirms a transaction through B; B send the transaction data to S.
2. S:
   - converts the transaction data into a summarized form, T;
   - computes
     - two random numbers; namely, a random transaction key, k_t, and a random length, l_{tan}, which will be used as the length of TAN;
     - a transaction ciphertext, c_t, using k_s, where $c_t = \mathsf{SK.Enc}(k_s, k_t||l_{tan})$;
     - a hash of the transaction information, h_t, where $h_t = \mathsf{H}(T||user_{ID}||c_t)$;
   - encodes T, c_t and h_t into a QR code, $QR(T, c_t, h_t)$;
   - sends $QR(T, c_t, h_t)$ to B.
3. B displays $QR(T, c_t, h_t)$; U uses M; M:
   - scans $QR(T, c_t, h_t)$ and decodes it to obtain T, c_t and h_t;

  - checks whether the QR code or the transaction information has been tamper with by
    - computing $h'_t = \mathsf{H}(T||user_{ID}||c_t)$;
    - If $h'_t = h_t$, this verifies that the transaction information was sent from S, T has not been altered and the transaction was initiated by U with $user_{ID}$
  - obtains k_t and l_{tan} via $\mathsf{SK.Dec}(k_s, c_t)$;
  - computes $h_{tan} = \mathsf{H}(T||k_t)$;
  - converts h_{tan} into base-64 representation;
  - displays the first l_{tan} base-64 characters of h_{tan} to U. These characters will be used as TAN.
4. M presents T to U; U checks that the information in T is correct and authorizes the transaction by entering the TAN into B; B sends this to S.
5. S:
   - computes $h'_{tan} = \mathsf{H}(T||k_t)$;
   - converts the h'_{tan} into base-64 representation;
   - compares the first l_{tan} base-64 characters of h'_{tan} with TAN.
6. If the two character sequences match, the server verifies the integrity of T and that U authorized T.

Note that the inclusion of $user_{ID}$ in $h_t = \mathsf{H}(T||user_{ID}||c_t)$ is to provide additional assurance that the user initiated the transaction. It is not absolutely necessary to include this in the hash. The same applies to the inclusion of T in $h_{tan} = \mathsf{H}(T||k_t)$, which provides added assurance that the user authorizes transaction T. The scheme can function without the inclusion of either parameter.

5 Analysis and Discussion

5.1 Practical Issues

In view of the fact that the proposed approach uses PKI, this means that this scheme can be used for multiple Internet services. Unlike private key approaches, which requires each Internet service to establish a shared secret between the user and the respective server, the PKI approach avoids practical issues concerning the difficulty of pre-arranging shared secret keys. In addition, it is obvious that the user authentication phase can easily be used in conjunction with a traditional username and password to produce a two factor authentication solution; something that the user knows (i.e. the password) and something that the user possesses (i.e. the mobile device).

In practice, a reasonable value for l_{otp} and l_{tan} should be between 6 to 8 characters. Ideally, for security purposes the full hash value should be transmitted to the server. However, there is a trade-off between security and usability, as it would be impractical to require the user to input more than 10 characters. Therefore, we adopt a method similar to the approach in Starnberger [27] of converting the hash value into an alphanumeric form and only requiring the user to enter the first few characters. In our approach, we convert the hash value into base-64 characters. The base-64 representation of a hash consists of upper and lower case alphabets, the numbers 0–9, and two additional printable characters that can be decided by the system. This will result in 64 possible values for a character and each of them represents 6 bits of the hash.

5.2 User and Transaction Authentication

We show that without the active involvement of a legitimate user and the device, an attacker who controls the untrusted computer cannot successfully impersonate the user or perform an online transaction on behalf of the user.

Brute-Force Attack. Let l be the OTP or TAN length (i.e. l can represent l_{otp} or l_{tan}). Furthermore, let the values of l range between l_{min} and l_{max}. Hence the total number of possible values for the OTP or TAN, denoted by N_{bf}, is

$$N_{bf} = \sum_{i=0}^{l_{max}-l_{min}} 2^{6(l_{min}+i)}$$

and the probability of success of a random guess will be $\frac{1}{N_{bf}}$. If we consider the simple (and less secure) setting where $l_{min} = l_{max} = 8$, then the probability of success of a random guess is at most $1/2^{48}$. As with most password/TAN mechanisms, there should be a limit to the number of incorrect password/TAN entry attempts, which can effectively defeat the brute-force attack.

Password Stealing and Replay Attacks. Unlike traditional username and password login approaches which are vulnerable to password stealing attacks like key-loggers, shoulder surfing, or replay attacks, our approach employs an OTP method. Hence, any attempt to reuse the OTP will fail. In addition, unlike other approaches like SMS approaches that require an active cellular or Internet connection between the server and the mobile phone to transmit an OTP, in our approach, the OTP is sent via the untrusted computer and communicated to the mobile device through the visual channel. This approach also prevents password phishing, because the user does not even know the password until he/she initiates the user authentication phase.

To measure the success probability of a replay attack, first of all we should note that both the session key, k_s, and transaction key, k_t, are single use keys randomly chosen by the server in each session/transaction, which guarantees the uniqueness of input to the hash function in each session/transaction. However, since we use only the first few characters of the hash output as an OTP or TAN, there is a chance of hash collision even when the hash inputs in two sessions are different. According to the Birthday attack, for an OTP or TAN of length l (i.e., l base-64 characters or $6l$ bits), the chance (denoted by $P(l, q)$) of a collision among q different sessions is bounded by

$$P(l, q) \leq \frac{q(q-1)}{2^{6l+1}}.$$

5.3 Transaction Integrity

Man-in-the-Middle and Man-in-the-Browser. Man-in-the-Middle (MitM) and Man-in-the-Browser (MitB) attacks can come in a variety of forms, for example, phishing websites or Trojans on an untrusted computer. In an MitM attack, an attacker may create a spoofed website and lure the user into using this website, while relaying and attempting to modify messages between the user and the actual transaction server. In MitB attacks, an attacker essentially hijacks a session, and it has to be assumed that the attacker has full control of the untrusted computer. These attacks are difficult to defend against because it can happen after the user has already logged in and been authenticated by the server. For the scheme proposed in this paper, MitM and MitB attacks are addressed in the transaction verification phase. In general, an attacker can perform two malicious activities; in particular, an attacker can attempt to perform an unauthorized transaction, or alter the transaction information sent to the server and/or to the user.

To combat against such attacks, our approach requires important transactions to be verified with a TAN. Our analysis in Sect. 5.2 has demonstrated that the probability of success for an attacker to launch an unauthorized transaction is negligible. If the attacker attempts to alter transaction information sent from the server to the user, the computation of $h'_t = \mathsf{H}(T||user_{ID}||c_t)$ on the mobile device will be able to detect if changes were made to the transaction information, T, as the resulting value will be different from h_t. Similarly, if the attacker attempts to alter transaction information sent from the user to the server, the value of h_{tan} computed by the mobile device and the value of h'_{tan} computed on the server will be different with an overwhelming probability. It is worth noting that here the chance of a collision is nearly negligible (close to $1/2^{6l_{tan}}$) since the target T is fixed. Hence, transaction integrity is ensured as any attempt to alter transaction information will be detected.

5.4 Drawbacks

The security of the proposed approach relies heavily on the availability and integrity of a trusted mobile device. If the security of the mobile device is compromised and an attacker can steal the user's private key or hijack the mobile device, then the security of the proposed authentication and verification scheme will be compromised. This also applies if the user loses the mobile device or if it is stolen. Moreover, without the mobile device the user will not be able to use Internet services that are based solely on this scheme.

The proposed approach also assumes the trustworthiness, integrity and security of the PKI. It should be noted that if a PKI is by a user to secure multiple Internet services, so that a mobile device is only required to store one private key, the PKI will probably become the focus of attacks. Since once the security of the PKI is breached, an attacker will be able to gain access to the multiple Internet services employed by the user.

Another drawback of the scheme is that users might feel that it is tedious to have use the mobile device to scan a QR code and to enter a TAN for important transactions, despite having already been logged in and authenticated using the OTP. However, this may be a small price to pay to secure important online transactions, such as banking activities and financial transfers. Also, while our approach prevents MitM and MitB attacks from performing unauthorized transactions or altering transaction information, it only prevents attacks against transactions protected by the TAN. It does not defend against Denial-of-Service attacks or eavesdropping attacks.

6 Conclusion

This paper investigates the problem of authentication and the verification of online transactions performed on an untrusted computer or device. To address this problem, we proposed a user authentication and transaction verification approach using QR codes and a trusted mobile device, equipped with a camera. Our approach works via the visual channel and does not require an active connection. In this paper, we analyze the security of our scheme and discuss the mechanisms in the scheme for circumventing a variety of security threats including password stealing, man-in-the-middle and man-in-the-browser attacks.

References

1. Barkan, E., Biham, E.: Conditional estimators: an effective attack on A5/1. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 1–19. Springer, Heidelberg (2006). doi:10.1007/11693383_1
2. Bonneau, J., Herley, C., van Oorschot, P.C., Stajano, F.: The quest to replace passwords: a framework for comparative evaluation of web authentication schemes. In: IEEE Symposium on Security and Privacy, pp. 553–567. IEEE Computer Society (2012)
3. Bonneau, J., Herley, C., van Oorschot, P.C., Stajano, F.: The quest to replace passwords: a framework for comparative evaluation of web authentication schemes. Technical report 817, University of Cambridge Computer Laboratory (2012)
4. Chow, Y.-W., Susilo, W., Au, M.H., Barmawi, A.M.: A visual one-time password authentication scheme using mobile devices. In: Hui, L.C.K., Qing, S.H., Shi, E., Yiu, S.M. (eds.) ICICS 2014. LNCS, vol. 8958, pp. 243–257. Springer, Heidelberg (2015). doi:10.1007/978-3-319-21966-0_18
5. Chow, Y.-W., Susilo, W., Yang, G., Phillips, J.G., Pranata, I., Barmawi, A.M.: Exploiting the error correction mechanism in QR codes for secret sharing. In: Liu, J.K.K., Steinfeld, R. (eds.) ACISP 2016. LNCS, vol. 9722, pp. 409–425. Springer, Heidelberg (2016). doi:10.1007/978-3-319-40253-6_25
6. Clarke, D., Gassend, B., Kotwal, T., Burnside, M., Dijk, M., Devadas, S., Rivest, R.: The untrusted computer problem and camera-based authentication. In: Mattern, F., Naghshineh, M. (eds.) Pervasive 2002. LNCS, vol. 2414, pp. 114–124. Springer, Heidelberg (2002). doi:10.1007/3-540-45866-2_10
7. Cronto Limited, Cronto. http://www.cronto.com/

8. DeFigueiredo, D.: The case for mobile two-factor authentication. IEEE Secur. Priv. **9**(5), 81–85 (2011)
9. Denso Wave Incorporated. http://www.QRcode.com, http://www.qrcode.com/en/
10. Dmitrienko, A., Liebchen, C., Rossow, C., Sadeghi, A.-R.: Security analysis of mobile two-factor authentication schemes. Intel Technol. J., ITJ66 Identity, Biometrics, Authentication Ed., **18**, 138–161 (2014)
11. Dodson, B., Sengupta, D., Boneh, D., Lam, M.S.: Secure, Consumer-friendly Web Authentication and Payments with a Phone, pp. 17–38. Springer, Heidelberg (2012)
12. Dunkelman, O., Keller, N., Shamir, A.: A practical-time related-key attack on the Kasumi cryptosystem used in GSM and 3G telephony. J. Cryptol. **27**(4), 824–849 (2014)
13. Grosse, E., Upadhyay, M.: Authentication at scale. IEEE Secur. Priv. **11**(1), 15–22 (2013)
14. Huang, C.-Y., Ma, S.-P., Chen, K.-T.: Using one-time passwords to prevent password phishing attacks. J. Netw. Comput. Appl. **34**(4), 1292–1301 (2011)
15. Jeun, I., Kim, M., Won, D.: Enhanced password-based user authentication using smart phone. In: Li, R., Cao, J., Bourgeois, J. (eds.) GPC 2012. LNCS, vol. 7296, pp. 350–360. Springer, Heidelberg (2012). doi:10.1007/978-3-642-30767-6_30
16. Lee, H.-C., Dong, C.-R., Lin, T.-M.: Digital watermarking based on JND model and QR code features. In: Pan, J.-S., Yang, C.-N., Lin, C.-C. (eds.) Advances in Intelligent Systems and Applications, vol. 2, pp. 141–148. Springer, Heidelberg (2013)
17. Li, S., Sadeghi, A.-R., Heisrath, S., Schmitz, R., Ahmad, J.J.: hPIN/hTAN: a lightweight and low-cost e-banking solution against untrusted computers. In: Danezis, G. (ed.) FC 2011. LNCS, vol. 7035, pp. 235–249. Springer, Heidelberg (2012). doi:10.1007/978-3-642-27576-0_19
18. Liao, K.-C., Lee, W.-H.: A novel user authentication scheme based on QR-code. JNW **5**(8), 937–941 (2010)
19. Lu, J., Li, Z., Henricksen, M.: Time-Memory Trade-Off Attack on the GSM A5/1 Stream Cipher Using Commodity GPGPU, pp. 350–369. Springer, Cham (2015)
20. Mannan, M., Oorschot, P.C.: Using a personal device to strengthen password authentication from an untrusted computer. In: Dietrich, S., Dhamija, R. (eds.) FC 2007. LNCS, vol. 4886, pp. 88–103. Springer, Heidelberg (2007). doi:10.1007/978-3-540-77366-5_11
21. McCune, J.M., Perrig, A., Reiter, M.K.: Seeing-is-believing: using camera phones for human-verifiable authentication. In: IEEE Symposium on Security and Privacy, pp. 110–124. IEEE Computer Society (2005)
22. Mulliner, C., Borgaonkar, R., Stewin, P., Seifert, J.-P.: SMS-based one-time passwords: attacks and defense. In: Rieck, K., Stewin, P., Seifert, J.-P. (eds.) DIMVA 2013. LNCS, vol. 7967, pp. 150–159. Springer, Heidelberg (2013). doi:10.1007/978-3-642-39235-1_9
23. Parno, B., Kuo, C., Perrig, A.: Phoolproof phishing prevention. In: Crescenzo, G., Rubin, A. (eds.) FC 2006. LNCS, vol. 4107, pp. 1–19. Springer, Heidelberg (2006). doi:10.1007/11889663_1
24. Pohlmann, N., Hertlein, M., Manaras, P.: Bring your own device for authentication (BYOD4A) - the Xign-system. In: Reimer, H., Pohlmann, N., Schneider, W. (eds.) ISSE 2015, pp. 240–250. Springer Fachmedien Wiesbaden, Wiesbaden (2015)
25. RSA Security, RSA SecurID. https://www.rsa.com/en-us/products-services/identity-access-management/securid

26. Schneier, B.: Two-factor authentication: too little, too late. Commun. ACM **48**(4), 136 (2005)
27. Starnberger, G., Froihofer, L., Goeschka, K.M.: QR-TAN: secure mobile transaction authentication. In: International Conference on Availability, Reliability and Security, ARES 2009, pp. 578–583, March 2009
28. Subpratatsavee, P., Kuacharoen, P.: Transaction authentication using HMAC-based one-time password and QR code. In: Park, J.J., Stojmenovic, I., Jeong, H.Y., Yi, G. (eds.) Computer Science and its Applications, pp. 93–98. Springer, Heidelberg (2015)
29. Sun, H.-M., Chen, Y.-H., Lin, Y.-H.: oPass: a user authentication protocol resistant to password stealing and password reuse attacks. IEEE Trans. Inf. Forensics Secur. **7**(2), 651–663 (2012)
30. Vapen, A., Byers, D., Shahmehri, N.: 2-clickAuth optical challenge-response authentication. In: International Conference on Availability, Reliability, and Security, ARES 2010, pp. 79–86, February 2010