An evaluation of hypothetical attacks against the PassWindow authentication method

Matthew Slyman, University of Cambridge
Sean O'Neil, VEST Corporation
Gadescu Horatiu Nicolae, University of Birmingham
Ben van der Merwe, University of Stellenbosch

Abstract: This paper comprises an evaluation of the viability of various online attack methods against the PassWindow second-factor online authentication system. PassWindow claims to significantly improve user authentication security in the online environment by offering security features not available in comparable online authentication methods. The first sections of this paper explain how PassWindow works and how PassWindow compares to other existing online authentication methods. We also explore the viability of various online attack scenarios commonly used against existing online authentication methods, including analytical attacks by an advanced and determined adversary.

We conclude that within the scope of our analysis PassWindow credibly protects against online, transaction-specific authentication attacks and is less vulnerable overall in comparison to the existing mobile, software or hardware OTP (*one-time password*) authentication methods.

Introduction

PassWindow is a method for providing second-factor authentication in the online environment. It involves two segment matrices — a physical *key pattern* printed on a portable plastic substrate and a digital *challenge pattern* displayed as an image on an ordinary electronic screen, such as the display on a laptop or mobile device. These, when superimposed, reveal to the user a unique single-use passcode and a set of transaction-specific digits. This passcode is then used for online authentication and transaction verification.

The specific transaction information included with these digits; for example, a representation of the intended destination account or transaction amount, enables the user to visually confirm the purpose of the received authentication challenge. These features make PassWindow one of the very few authentication mechanisms presently available that offers robust and credible protection against the latest online Man-In-The-Middle (MITM) security threats.

Scope

Physical (in person) attacks on hardware, such as physically stealing the PassWindow authentication server, gaining physical access to the user's card token, or line-of-sight attacks whereby an attacker directly *shoulder surfs* a password in the physical presence of the user, are deemed to be outside the scope of this analysis, which focuses solely on addressing the online authentication problem for which PassWindow was designed.

We have defined what constitutes a credible attack with regard to all the authentication methods discussed in this paper as one that does not noticeably force the user outside the bounds of a normal authentication procedure.

Social-engineering attacks, which lead a user to perform actions outside the bounds of the normal authentication process, may not be fully mitigated by any authentication technology. These attacks are therefore not a major area of focus for this document.

In addition, we assume the computer system employing the authentication server technologies discussed here has not had its security compromised prior to installation of the authentication server system. An already-subverted authentication management system cannot be effectively secured post-hoc by any authentication security technology.

The PassWindow method

PassWindow is a novel authentication technology designed for use as a second-factor in online user authentication and transaction verification processes.

PassWindow utilizes segment matrices to transmit information decipherable only by the intended recipient upon superimposing a *physical key pattern* (something the user *has*) over a *challenge pattern* displayed on the user's communications device or computer screen.

The combination of the key and challenge patterns reveals the encoded information to the user alone, as a direct line of sight is necessary in practice for a human observer to view the complete pattern.

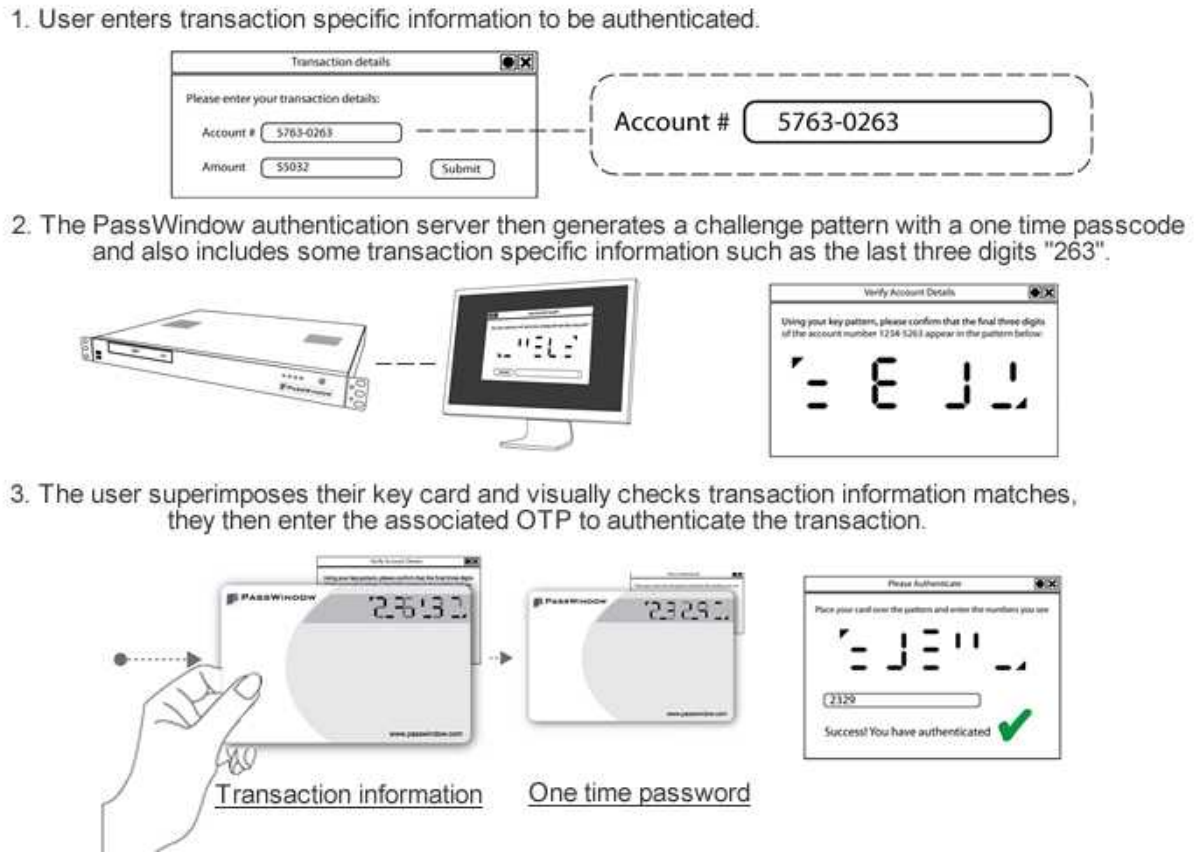
Any interception of the transmitted challenge does not leak sufficient information for an attacker to deduce the user's secret key pattern over the life of the card.

PassWindow challenge patterns can comprise a static challenge image or a more extensible and analytically robust animated challenge as discussed in this document. Animated challenges consist of a sequence of static challenges which either reveal encoded characters or add obfuscating entropy to the overall challenge.

The authentication server generates challenge patterns that are only meaningful when combined with the intended key. Any interference or tampering with the challenge pattern is passively revealed to the user by the appearance of pattern combinations that do not conform to expectation; for example, randomly placed segments, a sequence of digit patterns that does not conform to the user's expectations, or the presence of transaction verification information that does not pertain to the active transaction.

Any alphanumeric code may be transmitted securely with the PassWindow method. The current implementation transmits a short string of random digits for use as a one-time-password with context-identifying digits specific to the transaction the user is authenticating.

Once the user confirms that the transaction-details encoded in the challenge conform to the desired transaction, they can finalize the transaction by entering the corresponding single-use password.



The format of the encoded information may be customized according to the administrator's desired policies. As an example, the start of the single-use passcode digits may be delimited with a **P** and the corresponding transaction or account information may be prefixed with a letter **A**.

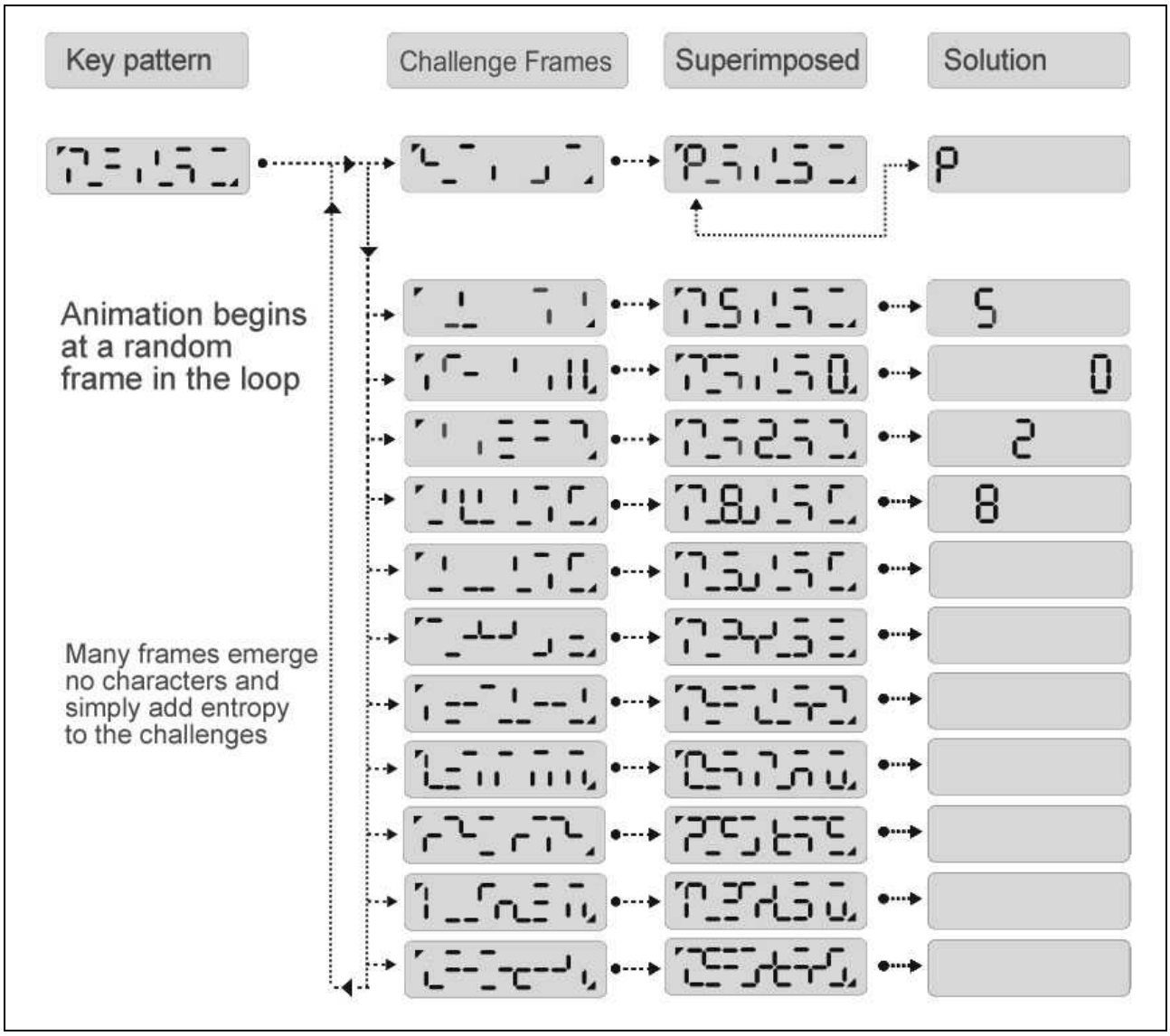


Figure 1: Example of a PassWindow key pattern and matching challenge consisting of twelve animated frames. The final pattern visible upon superposing the key/challenge pair is shown in the central column, and the four-digit solution is shown in the right column.

Each challenge frame contains approximately half the segments needed to visualize the intended character; in addition, extraneous segments act as background noise to obfuscate the true location of the single digit within each frame of the animation.

Additional frames in the sequence contain only extraneous segments and reveal no characters at all. These serve to reduce the information density of the challenge, thereby enhancing the entropy of the system and complicating any potential long-term deductive analysis of the user's key that might be attempted through interception and analysis of a user's challenges and responses.

Once the user enters the unique one-time password, the authentication server checks whether the response was consistent with the challenge; and, if so, reports to the supporting software infrastructure that the user or transaction has been successfully authenticated.

Transaction authentication

PassWindow can encode specific transaction verification information into the PassWindow challenge alongside the one-time password (OTP). This transaction-specific information is viewed and confirmed passively by the user. If correct, the user proceeds by entering the associated OTP.

For example, the following series of characters encoded in the challenge — **P**123456**A**007 — might be used to authenticate the creation of a new online transaction destination account entry within the user's online banking platform. Within this example:

P designates the beginning of a six-digit one-time password (OTP), in this case: **123456**

A designates a destination account identifier comprising, for example, the last three digits of the account number; in this case: **007**

The design and security profile of transaction authentication codes may be modified dynamically to suit a wide variety of specific online authentication problems.

Factors in online security

Threats to online security may be divided into *network-based attacks* (originating from a remote agent) and *locally-hosted attacks*, such as those facilitated by malware already installed on the client's system, such as *Trojans*, rootkits etc.

Authentication security evaluations often focus mainly on network-based attacks under the assumption that the user's computing terminal (i.e., their desktop, laptop, or mobile device) is a secure platform. However, it is common for the attacker to have gained full access to the victim's machine via hidden communications processes introduced by residual malware or by exploiting unaddressed security holes in legitimate software.

We assume the attacking agent in our authentication security evaluation is similar to that described in the Dolev-Yao model [18]. We endow this agent with the ability to monitor, manipulate and impersonate all networked software on both the user's online electronic terminal and freely access all authentication information passed between the terminal and the web server running the authenticated service.

The prevalence of malware has created the need for two-factor authentication. The first factor being something the user knows — their username and password — and the second factor is something the user physically possesses. To offer credible protection against the prevailing threats, this second factor should be effectively *isolated* from the potentially compromised network upon which the user is authenticating.

Common second factor authentication methods

SMS-based authentication

The designation of SMS or mobile-based authentication systems as a second-factor authentication method is incorrect — a more accurate term is '*out of band*' authentication. However, with the increasing adoption of GSM and network-enabled smartphones and tablets, even this security benefit may be lost if the user is authenticating a transaction being carried out on the mobile device itself.

Third parties may either gain access to the authentication codes sent via SMS via traditional malware-based interception [1] or by intercepting and decrypting data sent via the GSM telecommunications network [2].

Mobile authentication attacks have been successfully conducted without such technologies. Instead, the attacker has simply impersonated the user to their carrier and requested that all SMS messages be forwarded to a different phone number for the duration of the attack [3].

Another authentication method uses the mobile device camera to read a challenge image displayed on the user's workstation which is encoded with OTP and transaction information. This method makes the mistake of assuming that the operating system on the user's mobile device does not suffer similar malware vulnerabilities as all other forms of networked electronic software [4].

Biometric online authentication

Biometric authentication as a second-factor authentication method — in this case *something you are* — is frequently proposed for online authentication. While biometrics can simplify the identification of end-users via a secure terminal; due to the lack of transactional context-awareness in challenge responses, vulnerability to replay or spoofing attacks and the inability to reissue compromised biological data; biometrics offer no advantage as a second factor for online transaction authentication, especially via potentially maliciously compromised terminals.

Biometric authentication provides the user with a convenient method of generating an online username; however, in the context of a hostile network and a compromised device, the overall security performance of such methods is no better than an ordinary username and password.

Electronic hardware tokens

Hardware tokens come in a few forms and include a variety of authentication security features. Most common hardware tokens generate one-time passwords (OTPs) through a cryptographic process using an internal secret key and a series of cryptographic events; or, more commonly, a secret key based on a shared, synchronized clock value.

The user reads the digits displayed by the device and manually enters them into their terminal for cross-reference with the authenticating server.

This simple method of electronic OTP generation remains vulnerable to man-in-the-middle attacks, as users are required to divulge the OTP without any means of validating the authentication context.

In response, many token manufacturers have added a small numeric keypad; markedly increasing the token's size but allowing the user to type-in transaction-specific information that is encrypted with the secret key before the user enters the resulting OTP into their terminal. This allows a type of transaction verification or transaction signing, and this does indeed provide some security against man-in-the-middle attacks.

However, this method is still vulnerable to attacks exploiting the laborious nature of the manual transaction signing process. As this involves a relatively lengthy task utilizing a cryptographic tool that is inherently hard for the end-user to understand, an attacker can more easily coach the user through a modified transaction authentication procedure under various false pretences, such as under the guise of device clock resynchronization or a security upgrade [6]. The time and concentration required to perform the manual operation has been successfully exploited to distract the user from the context of the transaction information they are entering, and consequently attacks have been successfully perpetrated on a large scale [7][8].

Printed OTP lists / number grids

An older method of providing single-use passwords is to print a list of randomly-generated passcodes or transaction authorisation code numbers on a sheet of paper or scratch-card. Each passcode is then requested in sequence and used to authenticate a single transaction.

Alternatively, a grid of characters would be printed, and the authentication server would issue a challenge querying the characters located at specific coordinates.

Both these methods use keys and challenges that are verbally communicable. This allows an attacker to interrogate the user about the next valid code through malware, social engineering or phishing attacks. In addition, the relatively low entropy of the lists or grids necessitates frequent key reissue to prevent repetitious use of codes.

These methods remain vulnerable to the full range of man-in-the-middle attacks which are the common Achilles heel of all context-agnostic authentication methods.

Hypothetical attacks against PassWindow authentication

Man in the middle / network phishing

Man-In-The-Middle (MITM) attacks occur when an attacker positions themselves between client and server, impersonating both to each other and intercepting, recording or altering communications between them [9].

Phishing is an example of a MITM attack whereby the user is presented with a fake authentication screen that instead reports their authentication details to the attacker while the user remains unaware that these details have been compromised to be used maliciously [10]. This attack method is one of the most effective and most difficult to defend against. The standard one-time password (OTP) methods fail to provide protection as the OTP itself is simply passed to the attacker along with any other required information, such as a username and password.

Evaluation: Encoding transaction-specific information into the challenge at the authentication server level is a critical defence against MITM attacks. According to [19] a key element in ensuring two-factor authentication systems are effective is their ability to authenticate the *transaction*, not just the *person* carrying it out.

PassWindow addresses this issue by providing *passive* transaction-level verification (as explained previously in the "Transaction authentication" section) to ensure user awareness of the transaction they are authorizing before entering the OTP to finalize that specific transaction. PassWindow protects against fraudulent transaction-specific MITM attacks by providing authentication in both directions — from the user to the server and server to user (where *user* refers to the human end user, not merely their workstation or device).

When logging into an account or performing other generic authentication procedures where no specific transaction data are typically exchanged, PassWindow can encode other event or user-specific information; such as the originating IP address of the login request, an incremented authentication counter, the date of the last successful login or other data that might alert the user to any unauthorized access or activity.

Man in the browser / locally-installed rootkit malware

Malware that is installed locally on the user's computer or communication device provides a more accessible vector for bypassing authentication measures. Not only can these threats be used as a platform for enhanced man-in-the-middle attacks, but they also facilitate more comprehensive manipulation of the authentication experience.

The assumption must therefore be made that local malware has not only the ability to intercept all the communications between the user and online server, but to manipulate the user's screen and all the associated contextual information provided by the remote web server, thereby maintaining the impression of a normal authentication experience.

This has given rise to the so called Man-In-The-Browser (MITB) attack, in which authentication context is manipulated to deceive the user into entering valid authentication values such as an OTP generated by their electronic hardware token. The malware then surreptitiously uses this OTP to authenticate a hidden transaction while the user is fooled by the fake user-interface into believing that everything is still normal [11].

Many variations on this tactic have been discovered 'in the wild', frequently derived from the widely-distributed Zeus malware [12], which injects what appears to be a genuine HTML login form directly into the user's browser requesting the user's authentication details while the browser (according to the URL field) appears to remain connected to the intended service [13].

Once the attacker has used this authentication information to connect to the service, if a second OTP is required to validate an outgoing transaction, a popular tactic involves simulating a browser "session expired" message followed by a new authentication OTP-request form. The attacker then uses the valid passcode supplied by the end-user to validate the hidden transaction.

This attack is effective because the information injected into the browser disguises the authentication context the user is actually facilitating. Because no context-specific information is available when following hardware-token-based authentication procedures, the user is unable to identify what they are actually authenticating. The attacker's real transaction can instead be presented to the user as a routine login authentication that requires the same type of OTP.

Alternatively, the attacker might wait until the user wishes to perform a similar transaction (such as transferring funds from one account to another) and then modify only the transaction information within the web page; for example, replacing the destination account with the attacker's account details when the user is authorizing an online payment [14].

Evaluation: With transaction-specific information encoded into the challenge by the PassWindow authentication server, the attacker is unable to remove this transaction information from the challenge without knowing which part of the challenge animation contains the encoded transaction authentication. The same fundamental difficulty exists for an attacker with root access to the user's device via locally installed malware.

The encoded transaction information within the associated OTP reveals the genuine context directly to the user regardless of the fake context presented by the malware user-interface in the browser, and this should alert an educated user. PassWindow's sequential and repeating visual challenges further reinforce user awareness of the true transaction information and authentication context as they identify the encoded OTP.

Social engineering attacks

In a *"Social engineering attack"*, the user is persuaded to divulge secret information or authorize a fraudulent transaction. To classify an attack as "social engineering", the deception must involve more than the attacker simply asking the

user to validate the attacker's transaction outright — credible deception must be employed in any social engineering attack [15].

Evaluation: With sufficient transaction information encoded into a PassWindow challenge, it is markedly more difficult for an attacker to manipulate an educated user into authorizing an unwanted transaction. An attacker may instead try to interrogate the user as to the configuration of the key pattern itself.

PassWindow's key patterns are not easily communicated either verbally or through typed characters, thereby eliminating the most convenient telephone-based social engineering attacks that are employed against electronic hardware tokens, a method that has been termed "vishing" [20]. These attacks involve someone telephoning the user and impersonating an authorized representative of the secured service. A verbal request is made for a valid authorization code to be read from the victim's authentication device to supposedly enable the caller to reveal, for example, "important confidential information". It is unlikely that an attacker would attempt to extract a PassWindow key pattern from the customer in this way, as it is difficult to verbally explain the visual characteristics of the PassWindow segment matrix.

An attempt might be made to convince the user to physically mail the authentication key card or any other authentication device to the attacker, but that is unlikely to convince an educated user and also puts the identity of the attacker at risk of detection. Likewise any demands for a user to physically copy their visual key at a scanner, photocopier or webcam would arouse their suspicions as this would directly contravene the only rule of handling static-pattern PassWindow key cards: to avoid visual surveillance or copying of the key. This rule will be familiar to any users acquainted with ATM machines, so user training requirements are minimal and end-users are well prepared to evade attempts to obtain their key details in this way.

Direct attack on the PassWindow authentication server

An attacker may try to attack the PassWindow authentication server directly in order to compromise the entire PassWindow authentication procedure.

Evaluation: If configured according to the server documentation, the PassWindow server should not have direct access to any external network. The sole communications link to external devices allows no administrative access whereby its internal data or settings might be revealed. Ultimately, the security of any authentication back-end system relies on the security of the Internet-facing server(s), which is not affected by PassWindow and therefore beyond the scope of this analysis.

The PassWindow authentication server utilizes a very simple and limited communications protocol, and all authentication processing is carried out within the server itself. Its functionality is limited to generating challenge image data, receiving short passcodes and user identity values, and ultimately issuing a binary (yes/no) response to an authentication request. In addition to this, various authentication policies manage the acceptable query rate and response time limits. This basic numeric communication to the authentication server provides little scope for an attacker to engage the server directly in any meaningful way that could lead to useful access.

So long as it is properly positioned on the network, the authentication server should not be directly accessible from the public Internet, instead being placed *behind* the public-facing server that contains the information or target an attacker would be seeking, so that as a more security-hardened and less accessible target containing less intrinsically valuable information, the PassWindow server is unlikely to be a target itself.

Denial-of-service attack

PassWindow does not significantly affect the vulnerability or invulnerability of a system to denial-of-service attacks, except insofar as such attacks may be foiled by preventing unauthorized access to privileged systems.

Malware + webcam attack

Photographic attacks against traditional keys have been demonstrated using high-resolution cameras and zoom lenses from a distance [22]. Theoretically, malware in the end-user's machine could use integrated camera hardware to capture images of a PassWindow key when shown for authentication in front of a natural backlight. Photographic duplication of PassWindow keys is impeded by dark tinting in the key window. Such attacks may therefore be difficult to perfect in private spaces where lighting and surveillance conditions are usually unfavorable for an attacker. While this attack has not yet been practically demonstrated, cautious static-pattern PassWindow key users might cover or disable any integrated webcam hardware during authentication to ensure the security of their key.

Analytical attack on secret key

A highly-motivated attacker may attempt to deduce the user's printed key pattern via an analytical (e.g. statistical or algebraic) attack. This could be carried out using a sophisticated man-in-the-middle or malware-based monitoring program installed locally that enables interception of both the PassWindow challenge patterns and the user's respective responses. Over time as the attacker accumulates these challenge/response pairs, the attacker may potentially gain some insight into the PassWindow key pattern through analysis of the intercepted data.

The salient strength of the PassWindow method that an analytical attacker must overcome is the wide range of possible candidate key segment patterns that could generate the (challenge/response) pairs observed via interception. A number of measures have been built into the supporting software to minimize this threat.

For any analytical attack to be successful, the attacker must still devise a way to overcome the following technical challenges when attempting to deduce key patterns:

Firstly, the PW authentication server runs a parallel statistical analysis on each user's authentication behaviour against a model presuming a worst-case analysis scenario, retiring and reissuing any user's key that may have leaked sufficient information to enable deduction of the key pattern.

Secondly, as the attacker makes analytical gains towards deducing the user's key, the PassWindow authentication server actively confounds analytical attacks by dynamically customizing the challenge to introduce greater entropy, which is accomplished with little adverse effect on the user experience. This is done via the inclusion of extraneous challenge segments and the addition of null (non-character emerging) "*noise*" frames. These inclusions may be configured on a per-implementation basis in accordance with an application-specific analysis of attack economics. These features substantially increase the difficulties involved in analytically exploiting any key card's authentication history. This increases the number of challenge and response combinations required to deduce the key; pushing the attacker's required interception target beyond the card's service lifetime. This capability differentiates PassWindow from standard electronic hardware devices, which require their cryptographic values and compatible challenge configurations to be set at the factory at the time of manufacture and are immutable for the lifetime of the device.

Thirdly, for an attacker to obtain any information about a key, a PW user must physically hold their card against the screen and manually enter the visualized digits. This limits the amount of information that can be gathered from an ordinary human user. For example, an ordinary user may conceivably authenticate once per day but certainly not 10,000 times in one day. This limitation means the attacker will only ever be able to work with a relatively small dataset.

Evaluation: After overcoming the inherent logistical and mathematical difficulties of deriving the key pattern in this way; even an attacker with pervasive access to communication (challenge/response) intercepts would be frustrated by the PassWindow server's active information monitoring and countermeasures.

The mathematical details of this statistical challenge control will be presented in a second paper titled "PassWindow challenge complexity analysis" [17].

Tampered (weakened) challenges

An attacker may attempt to subvert PassWindow's defences by removing animation frames from a genuine (intercepted) challenge before delivering a weakened (simplified) challenge to the user to solve. This method would reduce the entropy of a challenge to elicit details that might simplify analysis of (challenge/response) intercepts.

Evaluation: Without knowing which frames will or won't reveal the digits of the passcode, it is almost inevitable that frames containing genuine characters will be removed, destroying the challenge. The manifestly corrupted challenge passively alerts the user to the attempted attack, arousing the end-user's suspicions about the computing machinery and communication channels in use. Therefore, this attack method carries risks for the attacker and would furthermore be most unlikely to succeed. A more thorough treatment of these attacks will be published in the second white paper mentioned above [17].

Comparative tamper-resistance of authentication technologies

Technology		Economics			Attacks						
Auth. key based on:	**Authentication technology:**	**Total cost of ownership**	**Technol. simplicity & opertnl. transparency**	**Convenience & procedural simplicity**	**Resistance to offline optical surveillance**	**Resistance to social engineering**	**Man in the Middle / eavesdropping**	**Malware (not on mobile device)**	**Mobile or trans-platform malware**	**Malware + high-res. webcam/ microphone**	**Overall evaluation (for online authentication)**
Something you know	**Password** & shared secrets	LOW	FAIR	POOR	POOR	POOR	FAIR	POOR	POOR	POOR	POOR
Something you are — digitized biological traits	**Biometrics** — fingerprint	MED.	POOR	FAIR	GOOD	FAIR	FAIR	POOR	POOR	POOR	POOR
	Biometrics — blood vessel patterns	HIGH	POOR	FINE	GOOD	GOOD	FAIR	POOR	POOR	POOR	POOR
	Biometrics — iris recognition	HIGH	POOR	FINE	GOOD	GOOD	FAIR	POOR	POOR	POOR	POOR
Something you have — a physical token, which might conceal from the transaction terminal a static or changeable key or set of keys	**Mobile SMS / voice-call OTPs** or "mTAN"	LOW	FAIR	GOOD	FAIR	POOR	POOR	FAIR	POOR	FAIR	FAIR
	List of OTPs — printed on scratch card	LOW	GOOD	GOOD	FAIR	POOR	FAIR	POOR	POOR	POOR	FAIR
	Hardware token — time-synchronized	MED.	FAIR	GOOD	FAIR	POOR	FAIR	POOR	POOR	POOR	FAIR
	Hardware token w. PIN pad, card reader & display	MED.	FAIR	POOR	GOOD	FAIR	GOOD	GOOD	GOOD	GOOD	FINE
	PassWindow with static-patterned keys	LOW	GOOD	GOOD	FAIR	FINE	GOOD	GOOD	GOOD	POOR	FINE
	PassWindow with LCD-displayed keys	MED.	FAIR	GOOD	GOOD	GOOD	GOOD	GOOD	GOOD	FINE	GOOD
	USB hardware token w. display, controls e.g. ZTIC	HIGH	POOR	GOOD	GOOD	GOOD	GOOD	GOOD	GOOD	GOOD	GOOD
	Estimated threat simplicity & frequency:				HIGH	HIGH	HIGH	HIGH	MED.	LOW	

These evaluations are based on the presumption of competently implemented technology and industry best-practices; and based on a hostile network environment with potentially malware-infected end-user terminals.

In addition to the evaluations given above; where PassWindow is provided as a service (e.g. "ShieldPass") or where provided by a separate internal team, this might help limit the damage that an insider/infiltrator can do, or improve disaster recovery options in some insider attack scenarios.

PassWindow is extremely simple and accessible to use and has been tested successfully by learning-disabled end-users. PassWindow has even been implemented in a paper-based system!

For online transaction authentication, very few technologies or products offer satisfactory accessibility, mass-market economy and resistance to online attack and social engineering.

Just as it is said in project management; "*Good, fast, cheap: pick any two!*" It might be said of *worthwhile* authentication technologies: "*Secure, convenient, cheap: pick any two!*" *For the best systems, there is a fundamental balance between security, accessibility and economy*; your priorities should be considered when selecting the best system for your particular application.

PassWindow offers an excellent combination of economy, simplicity, accessibility and effective security; with a unique technological and economic relevance to many applications.

Conclusion

Our analysis concludes that PassWindow's 'low-tech' approach to online authentication provides greater security compared to the conventional second-factor online authentication methods in use today, such as software-based OTP generation, mobile authentication schemes such SMS code delivery, and even dedicated hardware tokens.

Indeed, it is PassWindow's simple approach that minimizes its vulnerability to online attack. The ability to encode transaction-specific information alongside a purpose-specific passcode as a single authentication event is its salient security benefit.

At the time of writing, we can see no practical method of subverting PassWindow's transaction-specific authentication functionality. All known vulnerabilities can be effectively mitigated by implementing the system according to simple best-practice recommendations. All other potential attack vectors are common to all authentication technologies, and all involve marked deviation from the normal authentication process.

This is to say that while social-engineering methods can always be employed, this is an issue of user education. We contend that PassWindow's consistent and straight-forward authentication process facilitates user awareness more effectively than the other methods discussed.

Therefore, the likelihood of such attacks being widely successful, the critical aspect that motivates such undertakings, is extremely low.

In today's hostile network environment, with its profusion of highly-motivated attackers, readily-available hacking software and state-sponsored eavesdropping; every public-facing computer network should be considered unsafe. Therefore, for a second-factor authentication token to provide a useful security benefit, *it must be effectively isolated (electronically, cryptographically, optically etc.)* from the online network on which the authentication is taking place, and isolated from any network-attached or network-accessible devices which may be compromised in any feasible attack. The vast majority of authentication tokens being sold today do not provide meaningful isolation from the online network, and therefore cannot provide effective protection.

This statement challenges the existing security model of network-connected authentication methods, such as mobile SMS-based systems, particularly as mobile devices become increasingly connected to the Internet. Indeed, in evaluating any authentication method, it must be assumed that an attacker has already gained control of the software system hosting the authentication process [21]. Given this assumption, we conclude that PassWindow's non-electronic key provides a relatively safe haven from these threats.

About the authors

Matthew Slyman, M.A. in Computer Science, University of Cambridge. Matthew is developing novel commercially relevant data structures and algorithms, databases and ancillary tooling, ergonomically optimized graphical user interfaces, specialist engineering calculation tools and software security systems.

Sean O'Neil is a world-renowned cryptologist, reverse engineer, and code breaker with 19 years of experience in the industry as an IT security consultant, carrying with him a broad range of security expertise. He is responsible for the security of many products sold by a number of well known, large IT security and antivirus companies protecting corporations, academic, government and financial institutions around the world including the US Department of Defense, HSBC, and the Australian Government. Sean is also responsible for the design of VEST ciphers, EnRUPT, and the method of cryptanalysis known as monomial randomness testing also known as algebraic structure defectoscopy. Both EnRUPT and VEST ciphers have been recognized by the academic community as extremely hardware efficient, flexible and advanced cryptographic designs that differ dramatically only in their software performance and design complexity. EnRUPT, being the simplest block/stream cipher/hash ever made and VEST being the most complex stream cipher/hash ever made, are both used by many cryptology departments of universities as an educational tool for cryptology students.

Gadescu Horatiu Nicolae received his BSc. in Computer Science from University Politehnica of Bucharest, his Msc. of Science in Computer Security from the University of Birmingham. His thesis paper is written on the topic of verification of a digital envelope protocol using an automated tool (AVISPA). Since graduation he has been working in the industry developing security solutions for various online contractors. His areas of expertise include protocol analysis and formalization, penetration testing and computer forensics. He is currently Systems Engineer for PassWindow and has helped implement the authentication and administration servers.

Ben van der Merwe, Msc in Engineering Sciences, University of Stellenbosch. Ben is a systems architect and applied cryptographer. He has been the lead architect on several large-scale homeland security projects in Asia and the Middle East. Previously his academic studies were concerned with securing the space segment of LEO satellite systems where his work has been applied to multiple successful missions.

References

[1] Chickowski, Ericka (9 Oct. 2010)—
'Man In The Mobile' Attacks Highlight Weaknesses In Out-Of-Band Authentication

[2] Elad Barkan, Eli Biham, Nathan Keller—
Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication
Computer Science Department, Technion — Israel Institute of Technology

[3] Brett Winterford (6 Dec. 2011)—$45k stolen in phone porting scam

[4] Schwartz, Mathew J. (13 Jul. 2011)—Zeus Banking Trojan Hits Android Phones

[5] Christian Zeitz, Tobias Scheidat, Jana Dittmann, Claus Vielhauer, Elisardo González Agulla, Enrique Otero Muras, Carmen García Mateo, José L. Alba Castro
Dpt. of Computer Science, Univ. of Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany; Signal and Communications Processing Dpt., Univ. of Vigo, Campus Universitario, 36310 Vigo, Spain —
Security issues of Internet-based biometric authentication systems: risks of Man-in-the-Middle and BioPhishing on the example of BioWebAuth

[6] Spencer Kelly,(10 Feb. 2012)—Hackers outwit online banking identity security systems

[7] Dunn, John E (3 Jul. 2010)—Trojan Writers Target UK Banks With Botnets

[8] Het Belang Van Limburg, (24 Jul. 2010)—Belgian court found fraud in Internet banking

[9] NETRESEC Network Security. (27 Mar. 2011)—Network Forensic Analysis of SSL MITM Attacks

[10] Metropolitan Police Service. (3 Jun. 2005)—*Internet Banking Targeted Phishing Attack*

[11] Gühring, Philipp (24 Jan. 2007)—*Concepts against Man-in-the-Browser Attacks*

[12] F-secure. (5 Feb. 2012)—Threat Description Trojan-Spy:W32/Zbot

[13] Atif Mushaq, FireEye (19 Feb. 2010)—Man in the Browser: Inside the Zeus Trojan

[14] Jozsef Gegeny, Jose Esparza (25 Feb. 2011)—Tatanga: a new banking trojan with MitB functions

[15] Sarah Granger. (18 Dec. 2001)—Social Engineering Fundamentals, Hacker Tactics

[16] Simon Nettle, SecureLC Limited (6 Jul. 2011)—ShieldPass user statistics

[17] Gadescu Horatiu Nicolae, Sean O'Niel (9 Jun. 2012)—PassWindow challenge complexity analysis

[18] Dolev, D.; Yao, A. C. (1983)—On the security of public key protocols
IEEE trans. on Information Theory **IT-29**: 198–208

[19] Bruce Schneier (22 Sep. 2009)—Hacking Two-Factor Authentication

[20] Brian Krebs (20 Jun. 2010)—Spike in phone phishing attacks

[21] EU cyber security agency ENISA, (5 Jul. 2012)—Online bank robberies reveal security gaps

[22] UCSD Jacobs School of Engineering (30 Oct. 2008)—Keys can be copied from afar

[23] IBM: Thomas Weigold, Thorsten Kramp, Reto Hermann, Frank Höring, Peter Buhler, Michael Baentsch (2008)—ZTIC—http://www.zurich.ibm.com/ztic/—The Zurich Trusted Information Channel—An Efficient Defence against Man-in-the-Middle and Malicious Software Attacks