

## How 2factor Software Defeats Common Security Threats

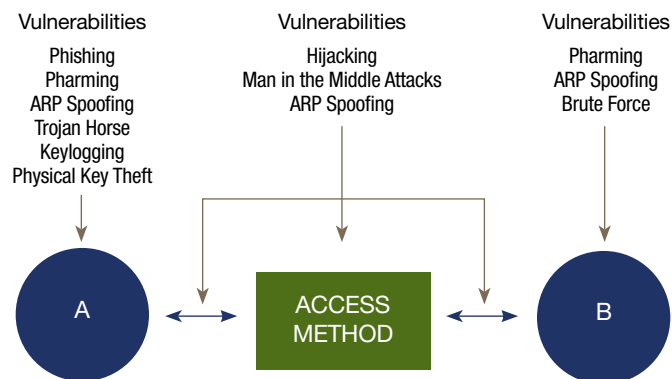
The increase in network devices and information has created an irresistible environment for hackers and network thieves. Potential threats to network security are expanding, and enterprises and service providers continue to search for better solutions.

In the world of electronic communication, authentication and encryption technologies are combined to secure any exchange of information. An array of authentication and encryption technologies are currently available in proprietary hardware or software products. Common breaches include man-in-the-middle, phishing, session hijacking, keylogging, and other attacks. Effective authentication is at the core of most security breaches. Most authentication solutions combat only specific threats; RPM can defeat any threat.

Authentication Method	Hijacking	Phishing	Pharming	MITM	ARP Spoofing	Trojan Horse	Brute Force	Keylogging	Physical Key Theft
Password	No	No	No	No	No	No	Maybe	No	No
Cookies	No	Yes	Maybe	No	Maybe	No	Yes	Maybe	No
Tokens	No	No	No	No	Yes	No	No	No	No
Software PKI	No	Yes	Maybe	No	Maybe	Maybe	No	No	No
Smart Card & PKI	No	Yes	Maybe	No	Maybe	Maybe	No	No	No
SSL Certificates	No	Yes	No	No	No	No	No	No	No
RPM	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Traditional authentication methods typically authenticate only the first electronic “handshake” between two parties. Every subsequent exchange of information is vulnerable to attack. An exchange of information can take place between any combination of mobile devices, personal computers, servers, proxy servers, data storage devices, point of sale devices, set-top boxes or other devices (shown as A and B below). Access can occur through Internet, voice, mobile, wireless or other connections. With one-time authentication solutions, information is vulnerable to attack at any point in an exchange.

### sample vulnerabilities in exchanging information



Patented 2factor technology authenticates and encrypts every exchange of information, and operates at up to 100 times the speed of current solutions. Security attacks are no longer threats. Common examples are outlined below:

### hijacking

**What it is:** Attacker issues commands and transactions through a user's Web browser or takes control of a session between a Web browser and a server.

**How RPM defeats it:** RPM operates in a secure browser using continuous and mutual verification of the browser and server through a constantly changing set of authentication and encryption keys. These keys are private, held in secure keystores in the computer of the web browser and on the web server. Without the verified 256-bit authentication key, the transaction will not be recognized. Since the keys are continuously changing and both parties must know what the next expected secret key is, a hijacker cannot determine or produce the next expected key, thus their attempt to enter a malicious transaction will fail.

### man-in-the-middle (MITM) attack

**What it is:** A specialized attack, where an attacker interrupts the communication between two parties and controls communication between the parties by secretly reading and modifying the communications stream.

**How RPM defeats it:** 1. By providing continuous authentication of both parties. 2. Exclusive use of private keys. 3. All communications (including initial authentication and handshake) are conducted through a secure, non-public, encrypted web browser.

### phishing

**What it is:** An attempt to fraudulently acquire sensitive information such as passwords or account information by masquerading as a trusted entity in an electronic communication, typically through e-mail, instant messaging or spoofing (see ARP spoofing below). Malicious messages often contain links to fraudulent web sites that mimic a trusted web site. These fake sites often gather personal, sensitive information.

**How RPM defeats it:** Once a trusted institution installs an RPM-enabled server, the only method of securely contacting them is through the secured RPM-enabled Web browser launched by the authorized end user, which opens the only page for user entry of passwords or other sensitive information. Even if criminals could acquire the logon info and the customer's

RPM application, the authentication keys would be out of sync with the expected sequence because the key sequence would be unknown to the criminal and the connection would fail.

### pharming

**What it is:** Redirects a web site's traffic to a bogus web site, either by changing the hosts file on the victim's computer or by exploiting vulnerabilities in DNS server software. Victims can enter a legitimate URL and be redirected to a fake site, then be subject to ID theft.

**How RPM defeats it:** 1. By using the RPM with a secure browser, even if a victim's computer is redirected to a fake web site, that site will not have the RPM software and will not be able to generate the proper authentication from the "victim's" computer, thus the encrypted message from the user will not be seen by the hacker. 2. If the victim uses the RPM-enabled browser to open a secure connection, the software connects only to a legitimate web site that offers proper authentication keys in the sequence dictated by the RPM algorithm.

### arp spoofing

**What it is:** Sending fake ARP messages to confuse network switches as to the true identity of a computer or other network component. As a result, messages intended for one machine can be mistakenly sent to another machine allowing packets to be sniffed for passwords and other sensitive information. This technique can result in the insertion of a spoofed MAC address into an IP header, which could lead to Man-in-the-Middle attacks and Denial of Service attacks.

**How RPM defeats it:** When the customer tries to connect to the server, even if it is being spoofed, then an RPM-enabled browser will secure the message with the next expected shared secret key, which is only known by the original server. The transaction will not be authenticated and therefore the encrypted message will not be decrypted, so it will not be readable by the spoof.

### trojan horse

**What it is:** A destructive computer program that masquerades as a benign application. There are different varieties:

*Remote Access Trojans*—designed to give the attacker complete control of a victim's computer

*Data Sending Trojans*—designed to provide the hacker with specific sensitive data such as passwords, credit card information, log files, etc. This type of Trojan could also install a keylogger (see next page)

**How RPM defeats it:** The RPM secure browser that connects to a specific destination is controlled by the authentication and encryption keys being exchanged during each transaction for the particular destination. This makes it impractical for an attacker to use the application to send sensitive data to a malicious destination. Access to the RPM application may also require the correct username and password and can additionally be secured through use of a PIN. External control of a secure application would require that all of these elements be compromised.

### brute force (dictionary) attack

**What it is:** An attack that tries many permutations of a secret in an attempt to gain access to restricted material. Numeric-only passwords of limited length (i.e., 6-digit passwords found on tokens and in new Vista hard drive security) have a strength of  $2^{20}$ , or roughly 1 million combinations, which can be evaluated in a matter of seconds.

**How RPM defeats it:** RPM typically uses 256-bit hexadecimal keys with new keys generated for every transmission of data, selectable down to the transaction level. Thus, the key strength is  $2^{256}$ . New keys are generated, used, and discarded before even a small percentage of the possible permutations for one key can be calculated.

### keylogging

**What it is:** A malware program designed to record every keystroke on a victim's computer and remotely report that log of keystrokes to the hacker.

**How RPM defeats it:** Within the RPM application, everything generated from the applet is captured and encrypted, and is captured inside the application layer—a keylogger must do a lot more than just grabbing the input from a keyboard.

Because of the variability of the keys changing so fast within the RPM application, keylogging is not an effective tool against RPM, but 2factor could provide customized auxiliary layers if desired, as described below.

As additional protection, if keylogging has grabbed the UN/PW, a key synchronization problem will occur if a user is spoofed during the session. The server would detect different sequencing of the authentication keys, and is designed to not complete the authentication and connection process since there will be synchronization error for the next expected key. Even if the legitimate user attempts a logon after a theft of his UN/PW, his keys would not be the next expected secret, thus he would not be allowed to connect, alerting him to a recent theft of his

personal information.

In practical terms, the probability of a specific user being targeted with a keystroke logger then to be the subsequent target of the intentional theft of a laptop is quite remote, but the following remedy is suggested:

If it is required, a customized version of a UI can be developed to create a virtual keyboard or keypad to be used to insert a 6-digit hexadecimal key offset. A CAPTCHA™ can also be used as a key offset to address the problem of keys sitting in an unchanging state for extended periods.

An alternative mechanism would be to use a 6 character alphanumeric pin, which is served as a graphic CAPTCHA then typed in as a one-time pin. The CAPTCHA graphic is only generated after the login is completed and the latest authentication keys have been exchanged. Each CAPTCHA would be generated based on the properties of the next expected key. The Server decrypts the initial login verifying the UN/PW, then generate a 6 digit alphanumeric Verifier based on the next expected key and presents the Verifier to the user via a CAPTCHA.

In summary, this enhanced security process is in three steps:

1. Generate on the server side a Verifier based on the next expected key, and serve it to the user as a CAPTCHA.
2. User enters this one-time Verifier.
3. Client application matches the Verifier against the next expected key present on the user's PC. If it does not match, then user is not who they say they are.

### physical theft of encryption keys

**What it is:** The loss of a set of user keys through the theft of a laptop, USB drive or hard drive. This is more serious than the theft of a Username/Password

**How RPM defeats it:** Any stolen keys would, in a practical sense, be useless since every key is only used once and changes with every transaction. If stolen keys are used in another downloaded RPM application, the RPM-protected server would reject that key because it would be out of sync with the next expected secret key used to authenticate the actual (legitimate) user.