VPN Remote Access Authentication:

Enabling Remote Access Without Compromising Security

Joel M. Snyder
Opus One

Introduction

In two short years since the final version the IPSec standards were finalized, enterprise network managers have overwhelmingly voted confidence in the security, reliability, and manageability of IPSec Virtual Private Networks (VPN). With corporate installations of VPN gateways in the tens of thousands, IPSec is used all day, every day, in business critical networks around the world.

Designing and writing standards for security protocols is a long and arduous task. When the IPSec standards were published as Internet RFCs[1] (see Table 1 for a list of the key IPSec standards), the Internet Engineering Task Force (IETF) Security Working Group knew its job was far from finished. While IPSec was aimed squarely at LAN-to-LAN VPN services, the working group knew corporate network managers had another pressing issue: providing secure remote access for a growing number of off site users. Additional work in standardization would be required to offer traveling road warriors and telecommuters secure network services.

Remote Access is characterized by roaming users with single user workstations, such as PCs or laptops, who have unpredictable IP addresses and will connect and disconnect frequently. Telecommuters with stable IP addresses, such as are commonly used in cable modems and DSL services, can benefit from these changes but don't necessarily need them.

As network managers began to deploy IPSec-based VPNs, it became evident that two major areas in the IPSec standards needed attention in order to properly support remote access in these environments. First, some mechanism for controlling the IP addresses for remote access users is required to solve access control and routing problems. The second area the IPSec standard requires work in is establishing a proper authentication method.

[1] RFCs are "Requests for Comments," the consensus-driven standards process for the Internet community. Just as the UN's International Telecommunications Union publishes "Recommendations" for telecommunications protocols, the Internet Engineering Task Force writes RFCs for Internet protocols. TCP/IP, the most widely used network protocol in the world, was published as RFC 791 and RFC 793.

The much needed IP addressing feature, already a given in older remote access tunneling protocols such as Point to Point Tunneling Protocol (PPTP)[2], was not included in the original IPSec standards set. This addition is needed to simplify deployment and give network managers more flexibility in the placement of VPN tunnel servers on their networks.

IP Security (IPSec)	*Requests for Comment (RFCs)*
RFC2401	IP Security Architecture (Nov. 1998)
RFC2402	AH Protocol (Nov. 1998)
RFC2403	MD5 HMAC (Nov. 1998)
RFC2404	SHA-1 HMAC (Nov. 1998)
RFC2405	DES/3DES Encryption in ESP (Nov.1998)
RFC2406	ESP Protocol (Nov.1998)
RFC2407	IPSEC DOI (Nov. 1998)
RFC2408	ISAKMP (Nov.1998)
RFC2409	IKE (Nov. 1998)
RFC2411	IPSEC Standards Roadmap (Nov.1998)
RFC2451	CBC Encryption in ESP (Nov. 1998)

IPSec extensions to add centralized assignment of addresses have been written and stand before the IETF. Preliminary products that support these extensions are currently being tested by both VPN vendors and users. The first is called Dynamic Host Configuration Protocol (DHCP) Configuration and it uses existing DHCP servers sitting on the network protected by the VPN gateway to handle address assignment and management. The second is a simpler, faster, extension called Configuration Mode which doesn't have all the rich attributes of DHCP, but does meet the need for address assignment.

While address management is a crucial issue to solve for effective deployment of remote access, this white paper is focused primarily on the issue of adding authentication to support remote access users.

The second area of the IPSec standards requiring additional work is authentication. Of the three types of authentication supported under the IPSec Internet Key Exchange (IKE)[3] specification, only digital certificate-based authentication provides secure access for remote users. The other two IPSec methods, Pre-Shared Secrets and Encrypted Nonces, are generally not suitable for enterprise-sized deployment of remote access users. Digital certificates require additional services on the corporate network that are collectively known as a Public Key Infrastructure (PKI)[4].

---

[2] PPTP, Point-to-Point Tunneling Protocol, is a Microsoft-proprietary VPN protocol for remote access which was first introduced in Windows 95 and Windows NT. Although PPTP does have some limited security features, the design and implementation of PPTP does not provide a high level of security. PPTP's tunneling features live on in the IETF standard protocol L2TP (Layer 2 Tunneling Protocol), while the security features have been pushed into IPSec.

[3] Internet Key Exchange (IKE) protocol, is the piece of the IPSec standards that defines where and how authentication occurs. The data transport protocols ESP (Encapsulating Security Payload) and AH (Authentication Header) are brought into play once authentication with IKE is completed.

[4] Elements of PKI include certification authorities that sign certificates; certificate servers (typically using LDAP) that store certificates and certificate revocation lists (CRLs); and registration authorities that assist users in obtaining certificates.

Digital certificates are also called X.509 certificates, after the ITU Recommendation that describes their format. Digital certificates issued by security product vendors like Verisign, Thawte, and Cybertrust are used extensively across the Internet in Web servers where they play an integral part of the Secure Sockets Layer (SSL) communication used to protect electronic commerce transactions and private information. Although SSL and IPSec are not directly related, the widespread experience of IT managers in using X.509 certificates has helped to demonstrate the suitability of certificates as an authentication method.

Unfortunately, while many enterprises are on their way to some type of PKI implementation, few have fully integrated PKI and digital certificates into their corporate IT infrastructure.

Even if PKI components were commonly available as part of corporate network infrastructure, management of large numbers of digital certificates remains an unsolved problem. For helpdesks accustomed to the constant low-level problem of users with lost or forgotten passwords, the idea of distributing and maintaining public/private key pairs and digital certificates across a wide spectrum of applications is truly frightening. In fact, the scalability of PKI for end user authentication across multiple applications has not been successfully demonstrated. While PKI vendors are doing their part to make the management of digital certificates easier, the pieces are not yet in place to make certificates as easy to distribute and replace as passwords.

Other problems with PKI and digital certificates include the cost of the software and lack of expertise in PKI management. Some network managers also are wary of authentication solutions which are susceptible to lax user behavior. For example, if the digital certificate and public/private key pair are stored on a smart card, the user may not be careful in removing the smart card when transporting the laptop.

PKI-based authentication	
Strengths: High security unique user identification; multi-application/single sign-on capability	Weaknesses: Deployment and scaling of individual certificates expensive and complex; most certificate support "point" solutions rather than integrated into operating systems; managing expired and lost certificates very difficult

However, corporate networks do have an enormous installed base of user authentication systems based on other technologies, such as RSA's SecurID card. The SecurID card is a time-based authentication token which authenticates a user by requiring them to prove they hold a particular physical device (the card or token) at a particular moment in time. With over five million SecurID cards deployed in corporate networks today, support of SecurID as an authentication method is a critical requirement for any VPN solution.

Corporate network managers have also embraced Internet standards such as Remote Access Dial-In User Service (RADIUS), which provides a simple, yet effective, way to authenticate users over a network against a central database. RADIUS was originally written by Livingston Enterprises (now part of Lucent) for use with their Portmaster series of remote access servers (RAS), but was quickly adopted by all RAS manufacturers as the method of choice for dial-in user authentication. RADIUS was

subsequently published as an Internet RFC, and is now used heavily for user authentication in a wide variety of environments. RADIUS is often the network-based interface linking applications to token-based authentication systems like SecurID, CRYPTOCard from CRYPTOCard, Inc., and S/Key, designed by Bell Labs, as well as simple username/password databases.

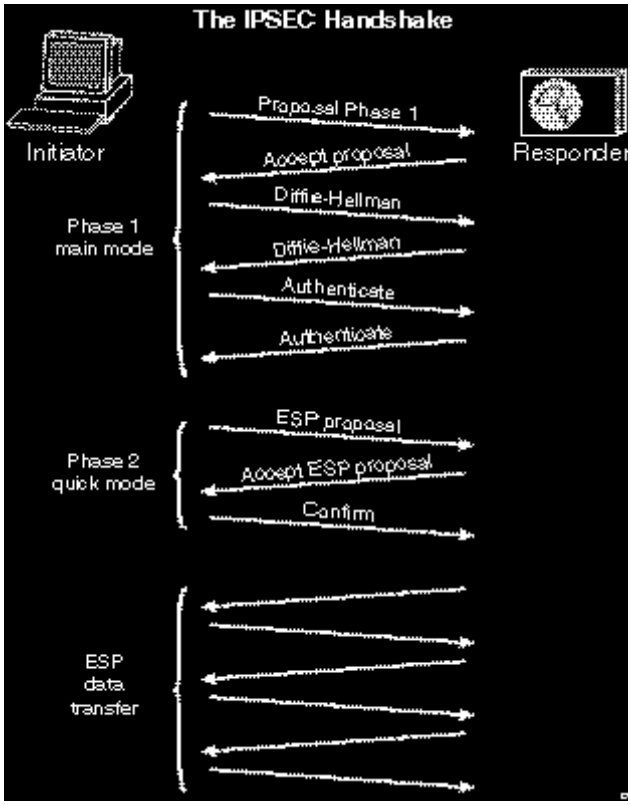
More importantly, network managers have already invested billions of dollars and substantial political capital in technologies like SecurID cards. This investment can't be casually replaced; therefore IPSec -- and specifically the IKE protocol -- needs to be changed to support these legacy authentication methods.

The question on the table for IPSec designers and implementers has been how to integrate existing user authentication systems into the standard while maintaining a high level of security, reliability, and manageability. Properly linking "legacy authentication methods" (LAMs)[5] to bring remote access users into corporate networks safely has been a difficult process. Due to market demand for secure VPNs, many vendors have proposed and implemented interim methods to link IPSec and LAMs. The challenge for the IETF has been to balance a vendor push for early adoption of existing systems as "*de facto* standards" with an IPSec designer push to consider all proposals carefully thereby not compromising the security standard set.

Why Must IKE Change?
• Typical user databases, such as Windows NT and RADIUS, not compatible with IKE
• PKI-based authentication not commonly available in corporate networks
• Large existing base of token-based authenticators
• Distribution and scaling of digital certificates still difficult for large user bases

Three systems geared toward solving the problem of linking remote access users and LAMs have been proposed to the IETF. These are called XAUTH (eXtended Authentication), Hybrid Authentication and CRACK (Challenge/Response Authentication of Cryptographic Keys). Most IPSec vendors have already delivered products to their customer base which use one or more of these proposed authentication systems. This early deployment has been good for the end user, but not so good for the standardization process. To protect their engineering investment and to minimize the impact on end users, each vendor is pushing for the technology they have chosen to be selected as the Internet standard. This creates some conflicting interests between the standardization process, which is designed to select the best technology available, and commercial interests of the participants. By understanding the issues and background, you can let your IPSec vendor know which system is right for you and your needs.

[5] One of the participants in the IPSec standardization process has a pet sheep, thus these are called LAMs in his honor. The pronunciation, then, is as "lamb," not "lame."



Understanding the IPSec Handshake

In all remote access systems based on IPSec, the encapsulating security payload (ESP) protocol is used to carry encrypted and authenticated traffic between a user and a security gateway, also called a VPN server. Before data are sent using the ESP protocol, the IKE protocol is used to set up ESP parameters and authenticate the two systems.

IKE operates in two phases, Phase 1 and Phase 2. In Phase 1, the remote user's machine and VPN server identify themselves to each other ("mutual authentication") and set up the keys that will be used for encryption with ESP. Phase 1 also sets up a channel between the user's machine and the VPN server–called a security association (SA) that will be used to encrypt and authenticate any further IKE traffic between the two systems. Figure 1 shows the IPSec handshake graphically. IKE uses the term "initiator" for the remote user and "responder" for the VPN server.

In Phase 2 of IKE, the two sides agree on security parameters for ESP, such as the choice of encryption algorithm, and on which traffic will be carried over ESP. A single Phase 1 SA can then be used to set up many ESP SAs. For example, when the lifetime of the keys used in the ESP SA expires, the Phase 1 SA is used to set up a new ESP SA to continue to carry traffic.

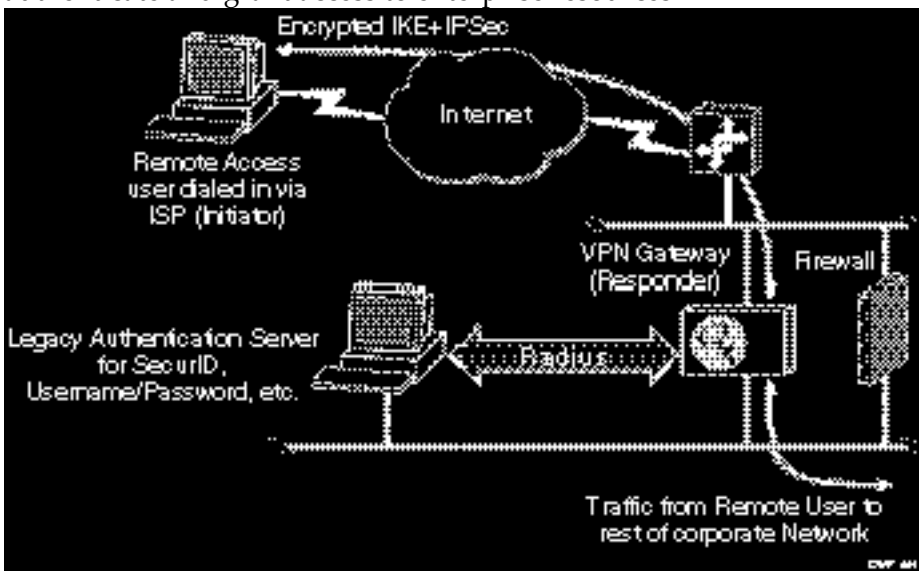
Because the ESP SA is established under protection of the Phase 1 IKE SA, it has well-defined security attributes which it inherits from Phase 1, such as strong mutual authentication of user and VPN server. The ESP SA itself continues this security by encrypting, sequencing, and authenticating every single packet carried in the SA. When the agreed-upon lifetime of the SA is over (typically an hour or so), Phase 2 is re-invoked

(again under protection of the already-authenticated Phase 1 IKE SA) to create a new SA with new keys.

What are the Alternatives?

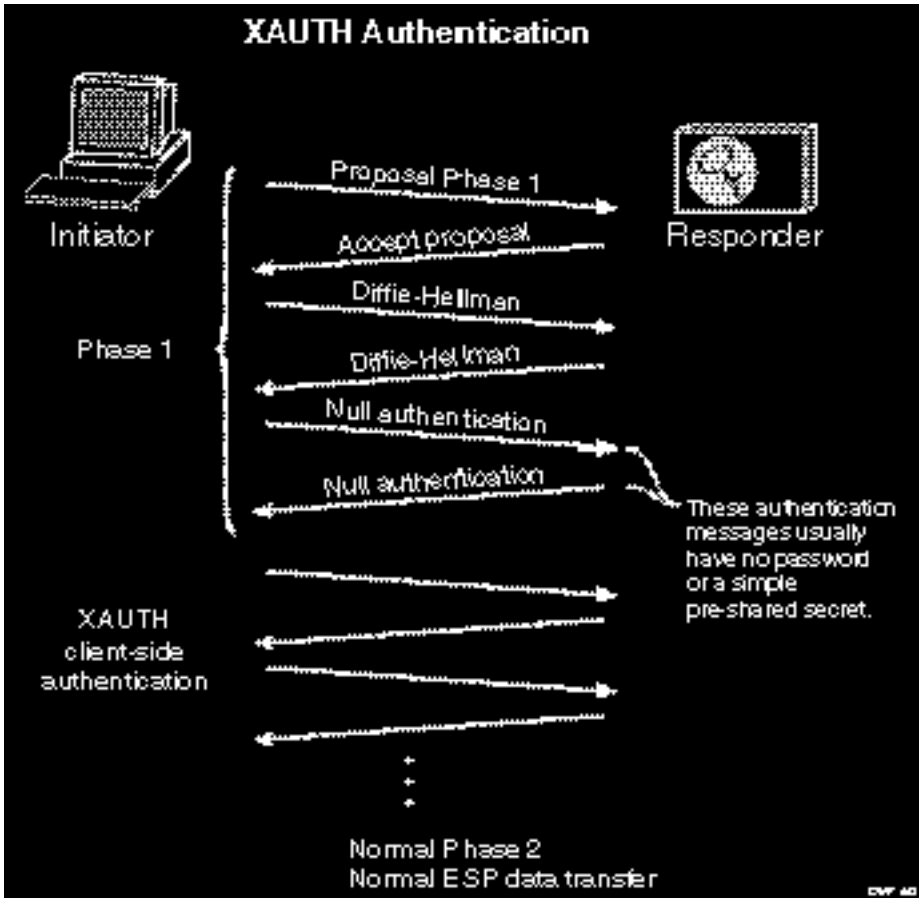
The IPSec community is considering three alternative authentication extensions to support remote access VPNs. All are designed to allow for asymmetric authentication: the VPN server authenticates itself to the end user differently than the end user to the VPN server. Legacy authentication methods, such as username/password and token-based systems, are fine for end users authenticating to VPN servers. But they don't work in reverse. The user can't authenticate the VPN server using the same technique–the VPN gateway can't look at a token card and read the number off of it, and if the gateway somehow could, the user couldn't then verify that it was correct by checking with the authentication server.

Before these three authentication extensions were proposed, IKE had always required symmetric authentication: whichever method, such as digital certificates, was used by the VPN server to authenticate to the user was the same method as used by the end user to authenticate to the VPN server. The solutions proposed to the IETF all rely on asymmetric authentication, having the VPN server use one method and the end user a different method, such as passwords. Figure 4 shows how end users, VPN tunnel servers, and legacy authentication servers can be arranged in corporate networks to authenticate and grant access to enterprise resources.



XAUTH is the longest standing proposal, and the one which has the most installed base among equipment vendors. XAUTH was first proposed by Timestep (now Alcatel), one of the first VPN vendors. In XAUTH, the Phase 1 IKE protocol is not changed. Either an empty password, equivalent to hitting "carriage-return" when asked for your password, or a single shared password is used for all users in the entire network to bring up the IKE security association. The goal is not to authenticate, but to get IKE running. At that point, XAUTH kicks in. This has been characterized by the designers of XAUTH as "Phase 1.5," since XAUTH runs between the Phase 1 IKE security association and the real Phase 2 ESP security association which is then used to transfer data. At Phase 1.5,

additional messages are sent using the newly-created IKE security association to accomplish the real end-user authentication. XAUTH can work with all common legacy authentication methods, including simple username and password, challenge-response systems (such as CryptoCards), and token-based authentication (such as SecurID). Figure 2 [[THIS IS CWP_002]] shows how the IPSec handshake is modified when XAUTH is in use.

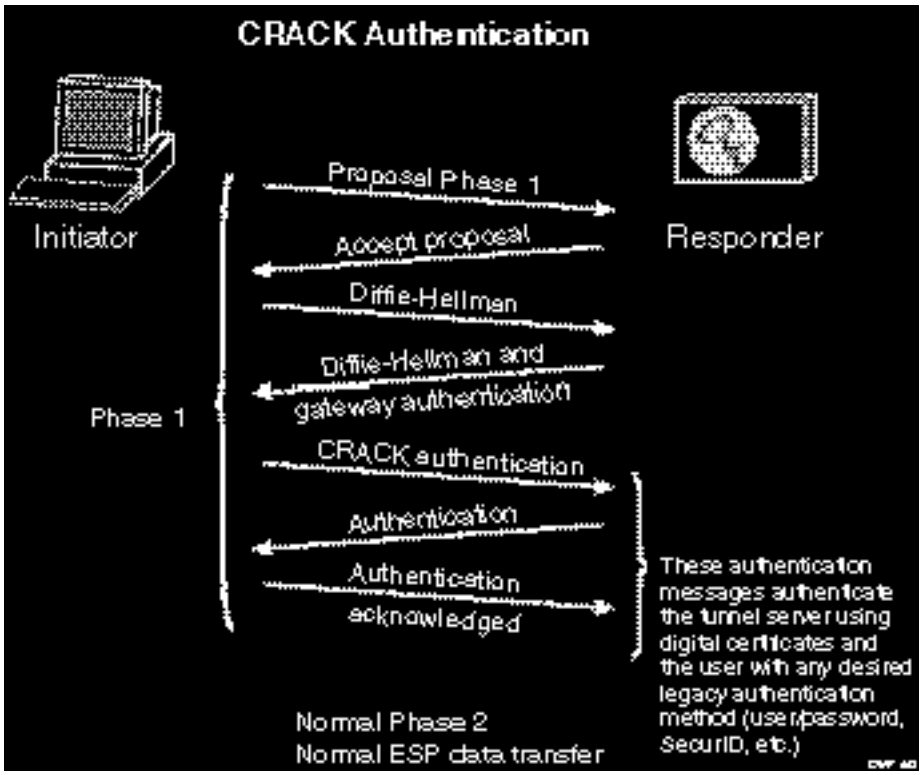


Hybrid Authentication, the second proposed authentication protocol, is a combination of existing Phase 1 IKE protocols and XAUTH. Hybrid Authentication was first proposed by Check Point Software, a firewall vendor. In Hybrid Authentication, a change is made to IKE that allows for asymmetric authentication. A new IKE authentication method is created where the security gateway authenticates itself to the remote access user (always using a digital certificate), but the remote access user is not authenticated to the security gateway. This allows the Phase 1 IKE security association to be created, although at this stage only the security gateway is authenticated. Once this happens, the XAUTH protocol is followed inside of the new IKE security association to authenticate the remote access user to the security gateway. Because Hybrid Authentication builds on XAUTH, all of the legacy authentication methods allowed by XAUTH are usable. Like XAUTH, the Phase 2 ESP security association completed inside a Hybrid Authentication scheme to protect data is created after authentication has been completed.

The third method of IPSec authentication proposed, Challenge/Response authentication of cryptographic keys (CRACK), is a new authentication method that allows for an

asymmetric form of authentication to take place directly within the IKE Phase 1 protocol. CRACK was first proposed by Network Alchemy (now Nokia), a high-availability VPN vendor. Figure 3 [[THIS IS CWP_003]] shows the IPSec handshake using CRACK authentication.

In the CRACK proposal, a fourth authentication method is added to IKE. This new method has the VPN server authenticate to the end user with a digital certificate, while the end user authenticates to the server with any legacy authentication method desired, such as username/password, challenge/response, or token-based systems. Only after the user fully authenticates is the Phase 1 IKE security association created. Then IKE Phase 2 proceeds as always.



How Do I Choose?

Your choice of remote access authentication system should be based on your business requirements. If your primary reason for deploying VPN technology across your enterprise is security, then you should choose the system that offers the best security. Unfortunately, for many security professionals, the process of choosing a remote access authentication method is unsupported by a complete understanding of the security risks involved. In this section, you'll learn the strong and weak points of each of the three proposals before the IETF. By understanding these strong and weak points, you can make the best–informed decision for your company.

XAUTH

XAUTH authentication	
Strengths: Simplicity, wide deployment	Weaknesses: Opens more security holes than it closes

security holes than it closes

XAUTH has the weakest security of all these proposals. With XAUTH, you open up a tunnel to someone you don't know and send them your password. XAUTH bypasses the strong mutual authentication which has long been a cornerstone of IKE. With XAUTH, the VPN server knows who the user is, but the user has no idea who they're talking to, other than that they appear to somewhere on the other end of the Internet.

In XAUTH implementations, Phase 1 authentication of IKE is handled by either an empty payload which has no authentication information in it or an empty password (pre-shared key). This allows the Phase 1 IKE security association to be established, in effect without authenticating either side of the conversation. Some implementations use a non-NULL, pre-shared key that is shared across all remote access users and is never changed. This offers somewhat better security, but has all of the security problems associated with stagnant passwords and introduces another one—if everyone has the same password it fails to be a password.

Once the Phase 1 SA is established, the end user authenticates to the VPN server using their choice of legacy authentication method. However, the VPN server **never** authenticates itself back to the user. In XAUTH, the VPN server knows who the user is, but the user doesn't know who the VPN server is. One of the most basic principles of IPSec, mutual authentication, is eliminated.

With XAUTH, you open up a tunnel to someone you don't know and send them your password.

The lack of strong mutual authentication in XAUTH is more serious than it seems at first glance. IKE depends on its authentication methods to prevent a man-in-the-middle (MITM) attack. With a MITM attack, someone who can intercept and modify the packets sent between a remote access user and VPN server can then eavesdrop on the conversation, insert messages, and delete messages–all completely undetected by either the end user or the VPN server. XAUTH's lack of real authentication within Phase 1 eliminates the protection against MITM attacks.

In XAUTH, it becomes a simple matter for a man-in-the-middle to impersonate the server (since the server is never authenticated to the user), intercept the now-unauthenticated cryptographic keys, and decrypt all traffic between the remote access user and the server. XAUTH breaks the IPSec model.

XAUTH makes other easy attacks on security possible. If the user's connection can be redirected to an attacker's server, the attacker can capture the user's authentication information (such as a time-sensitive SecurID password) and use them himself. The user thinks they typed in the wrong PIN and tries again, **never knowing** that someone else is currently masquerading on the network.

XAUTH's strength is in the long-standing history of its implementation. Although there is little cross-vendor interoperability with XAUTH, it is the most commonly implemented method of authentication for remote access users.

Hybrid Authentication	
Strengths: Solves XAUTH security problems	Weaknesses: Has its own subtle security problems; difficult to implement correctly

Hybrid Authentication is the most complex proposal before the IETF working group. Implementers must reference three different draft recommendations to properly implement Hybrid Authentication, which translates into increased complexity and cost, and greater likelihood of interoperability failure and bugs. Although the security of Hybrid Authentication is as strong as unadulterated IKE, the difficulty of implementation creates pitfalls for developers. Hybrid Authentication attempts to solve the security problems of XAUTH by combining the mutual authentication of IKE with the end-user authentication of XAUTH. The benefit is that the XAUTH model, which is well known and accepted, can be securely merged back into IKE for remote access users.

The problem with Hybrid Authentication is that once IKE Phase 1 is completed, the Phase 1 SA has been created, but not necessarily authenticated. Subtle coding errors in software running on products built according to this proposal become severe with the uncertainty as to whether or not the end user is authenticated. Simple problems in complex security protocols can cause massive security problems. For example, a gateway implementing Hybrid Authentication must take special care to ensure that an un-authenticated user is not allowed to proceed to Phase 2 and start sending traffic, while relaxing that requirement when the same user wants to re-key their ESP security association.

As we have learned through hard lessons, the construction of secure Internet applications is a difficult task, and very small errors can open a window of hacker opportunity. The number of security bugs reported daily on supposedly secure Internet applications is a sign that unnecessary complexity must be minimized. Whenever two equivalent solutions to a problem are presented, the simpler solution is the better solution.[6]

Unfortunately, trying to shoehorn Hybrid Authentication features into IKE can cause more problems than it solves. Part of the motivation for Hybrid Authentication is to change IKE as little as possible. The standardization of IKE in the IETF was a long process as it is already a fairly complex protocol. For that reason, one of the goals of the IETF has been to minimize changes to the IKE protocol. The difficulty of implementing IKE and the delicate interactions between parts of the protocol has led members of the IETF to declare: “no more changes to IKE!”

Keeping standard protocols stable is an admirable goal. But Hybrid Authentication–as are all the proposed remote access authentication methods–is indeed a change to IKE. Although the idea of Hybrid Authentication is to maintain the security of IKE with a minimum of changes to IKE, the reality is that the complexity of Hybrid Authentication makes for greater potential insecurity.

[6] This particular truth is often known as “Occam’s Razor.”

CRACK

CRACK authentication	
Strengths: Highest level of authentication and security	Weaknesses: Few commercial implementations available

CRACK is the most recent proposal, and has the fewest vendors offering CRACK-compliant products.

CRACK is the distillation of Hybrid Authentication, integrating all authentication back into IKE Phase 1. By taking the lessons learned in implementing XAUTH and Hybrid Authentication back to IKE, a new authentication method is added to IKE. CRACK allows remote access users to use legacy authentication methods, and it maintains the strong mutual authentication properties of IKE.

By adding a new authentication method to IKE via the CRACK protocol, the security properties which have been precisely crafted into IKE are retained. A critical goal of CRACK is that no sacrifice of security has been made for usability.

With CRACK, no sacrifice of security has been made for usability.

Because CRACK is another Phase 1 authentication method, it can be retrofitted into existing IKE implementations. The difficulties for developers posed by Hybrid Authentication are not present in CRACK. In CRACK, once Phase 1 is completed, the end user is authenticated–end of story. Adding CRACK into an existing IKE implementation is not a trivial matter, but it is straightforward and doesn't change the security model present in IKE. The risk of inadvertently adding defects into an existing implementation is minimized. CRACK offers a more straightforward design than hybrid authentication.

Because CRACK maintains the strong mutual authentication of IKE, it does not have the dramatic security weaknesses of XAUTH. The years of careful analysis which went into the development and approval of IKE still apply. The same confidence network managers have in the security of traditional IPSec VPNs is maintained in CRACK-based remote access VPNs.

Summary

Changes in the IPSec suite of security protocols are aimed at making IPSec-based products more accessible to remote access users. One of the stickiest areas of change is in authentication of end users. In IPSec, it is assumed that remote access users will use Digital Certificates for authentication, necessitating a full enterprise PKI implementation. IPSec's existing methods are not friendly to remote access users who are most comfortable with passwords and token-based authentication systems. As vendors have worked hard to develop extensions to IPSec for remote access, new authentication techniques have evolved. XAUTH, a generally insecure quick fix has grown into the more sophisticated, albeit very complex, Hybrid Authentication system. With CRACK, the lessons learned from implementation of XAUTH are properly folded back into IPSec. CRACK-based remote access authentication offers the proven and analyzed security of IPSec with the convenience of traditional authentication systems.