SECURITY

Discryption: Internal Hard-Disk Encryption for Secure Storage

Laszlo Hars, *Seagate Research*

Internal encryption to protect the confidentiality of stored data in disk drives has many advantages.

There have been many recent cases of information getting into unauthorized hands from lost or stolen laptops or insiders accessing unattended enterprise computers or storage devices. The Privacy Rights Clearinghouse maintains a long list of such reported cases at http://privacyrights.org/ar/ChronDataBreaches.htm.

Providing physical protection and using remote locations are two means of keeping stored data confidential. The least expensive secure-storage systems use local data encryption with optional data authentication, together with access control and physical tamper detection.Such devices, encrypting disk drives, are now being mass-produced after a period of sampling. Manufacturers are deploying them by large numbers in laptops, desktop PCs, data-center applications, portable media players, and TV broadcast video recorders.

The IEEE P1619 Security in Storage Working Group (http://siswg.org) is developing standard architectures for external encryption modules and tape drives. However, there's no standard yet for hard disks, where developers can adapt the data layout to security needs and provide access control to the encrypted data.

That means an attacker can only see the ciphertext after disassembling the drive and examining the magnetic platters with multimillion-dollar equipment. And because of the attacks' destructive nature, if the disk drive is returned, the owner will notice the disk was tampered with and won't trust the stored information. This effectively renders all kinds of data-modification attacks harmless.

ADOPTING A DISCRYPTION STANDARD

Adoption and utilization of a secure-disk architecture standard would offer a number of advantages, including

- freeing an implementer from custom-designing a security architecture;
- reducing development costs and time to market by avoiding the expensive and time-consuming security analysis necessary for a proprietary solution;
- providing a secure architecture that has already met public scrutiny;
- increasing trust levels, since nonprofits are viewed as more open than for-profit companies; and
- giving OEMs a second source of drives with similar security attributes.

The proposed *IEEE P1619 Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices* deals with the security of information in general storage devices that randomly read or write data in fixed-sized blocks. Its basic assumption is that attackers can access the stored data.

Discryption can mitigate these inherent risks. The proposed P1619 standard describes a transparent encryption module that developers can insert into the data path without modifying the data layout. This restriction doesn't apply to discryption, however, because developers can easily and transparently employ hidden disk areas or longer physical records. Hard disks can also accept new security-related interface commands.

EXTERNAL ENCRYPTION MODULES

The main drawback of encryption outside the storage device is the easy accessibility of the ciphertext. Connecting the hard disk to an unencrypting controller allows free access to the stored data. Inherent weaknesses result.

The most obvious weaknesses are evident with traffic-analysis attacks, which reveal the locality of changes when multiple snapshots were made on the disk. The attacker can copy back a disk sector's old content (unless a system performs expensive large-set data authentication), allowing malicious manipulation of data, such as undoing banking transactions and online orders and unspending electronic cash.

Even just randomly changing certain disk blocks can have catastrophic consequences for the drive's unsuspecting owner. One example takes advantage of knowing the location of system files. An attacker who locates a jump address in the beginning of an encryption block can randomize it by changing the corresponding ciphertext block. There's a nonnegligible chance that this will alter the OS's behavior, such as opening some security backdoors for later attacks.

Furthermore, attackers can send maliciously crafted documents or programs to unsuspecting users. If users save those documents, attackers might be able to find their location (where disk blocks changed). Randomizing a certain block will then change the file in a predictable way, altering documents or program behavior.

HOST SOFTWARE ENCRYPTION

Encryption performed in host software has all the drawbacks of the external encryption modules and all the risks associated with an open environment. User errors, SW bugs, and sloppy security policies could lead to the loss of secret keys or confidential data, and malware—rootkits, Trojans, viruses, and worms—could get into the system, compromising its security.

These weaknesses aren't present when developers do encryption in a closed environment with restricted I/O and unchangeable firmware.

DISCRYPTION ARCHITECTURE POSSIBILITIES

There are several options for discryption architectures.

Threat model, attack scenarios

Even if spying hardware (such as a key logger or cable snooper) is attached to the host, secure authentication should be possible using challenge-response protocols with random nonces. Since malicious host software can steal small amounts of data, users shouldn't enter any keys when the OS is running.

An attacker has at most onetime access to the ciphertext and can possibly read, modify, and copy other blocks, but not earlier content. Of course, attackers can carry out the usual general attacks, such as probes on data lines between electronic components and dictionary attacks on user passwords.

Encryption

Access control and encryption secure the data on disk drives. Available encryption modes include cipher-block chaining, various counter modes with location-dependent IV, location-tweaked electronic codebook (XTS of P1619), or wide-block encryption. Advanced Encryption Standard AES-128 or AES-256 are probably the best choices for the underlying cipher.

Key management

Developers can form data-encryption keys through secure combinations (cryptographic hash or encryption) of different entities: user key, platter signature, or a hidden root key in the electronics. The system randomly generates the user key, but it's not stored on the drive. For access control, the system also needs to store a secure hash of the user password. This way, the encryption key is of high entropy even if the user password is weak. After erasing the user-authentication information and key mixtures (for cryptographic disk erase or sanitation), a key search should deal with fullkeylength entropy.

Since the system might have several passwords, each with different associated rights, the encryption key can't be derived from any of them. The drive can restrict the number of login attempts with invalid passwords, mitigating the negative effects of weak user passwords.

The system might need key export or import, where the key is wrapped in a secure envelope. This carries security risks, but it helps with data recovery after the device electronics fail, or authorities in certain markets may mandate it.

User authentication

To authenticate users, the system can ask them to prove their knowledge of a secret or possession of some device with secret information (such as a token, smart card, or fingerprint). The authentication process can be as simple as providing a valid password or as complex as a challenge-response protocol with random or sequential nonces (useful against message-replay or man-in-the-middle type attacks).

The system can also support mutual authentication so users don't reveal their secrets to fake hard drives or the drive doesn't tell secrets to rogue hosts.

Administrators should limit the number of failed authentication attempts. For example, a drive could lock up after a user enters five wrong passwords, and only a higher-level authority could unlock it. This would hinder a search for weak user passwords.

The most basic authentication architecture features a user and master password. The latter can be used to reset drives that locked up after too many failed user-authentication attempts.

Access control

Without proper authentication, the disk drive shouldn't accept read/write commands, so an attacker won't see encrypted data or gain information about the locality of changes since a previous snapshot. Access control improves security, and export/import control authorities might require it. If the encrypted data was freely accessible, the user could use a secure disk as a stand-alone cryptographic coprocessor.

Disk drives can't provide absolute access control at a reasonable cost. Attackers can gain onetime access to the encrypted data when they remove the magnetic platters from a disk drive and place them on a multimillion-dollar spin stand. But when you employ tamper detection, legitimate users will detect tampering even when the disk drives are reassembled.

In this case, owners should no longer trust this data, so attackers can't make a second snapshot of new data, encrypted with the same keys, nor can they maliciously change the ciphertext and cause damage.

ADVANTAGES AND POSSIBLE FEATURES OF DISCRYPTION

Discryption has many advantages over other encryption methods. It costs less to implement than external encryption modules, and consumes less power than software encryption because of dedicated, optimized hardware.

With discryption, encryption is transparently performed in full interface speed without any host processor load. Security is better than with host software or controller-based encryption because of true random keys (no weak user keys) that are never stored in the drive. Disks drives are closed systems, making a malware infection impossible. In addition, developers can implement the security subsystem in a single chip, preventing debugger and bus-analyzer attacks.

The generation and storage of user keys in the drive provides further protection from malware. Hardware or hidden, protected ROM code performs security functions.

Discryption is easy to set up, use, and deploy. Disk drives are fully operational and secure after setup in the factory because of internally generated secret keys and default passwords provided in tamperproof envelopes.

Erasing keys provides fast and secure disk sanitation. Partitions can use different keys to separate user partitions and multiple operating systems. Unmounted partitions remain safe from malware, user errors, and lunchtime attackers.

Discryption can support multilevel and multifactor authentication as well as third-party services and encrypted communication. It also allows for hierarchical key management, internal re-encryption, and forensic logging.

Encryption for protecting the confidentiality of stored data is the most secure internally in disk drives, as opposed to host software or external encryption modules. There are also speed, cost, and power consumption advantages, flexibility in applications, extra services, and the simple and failsafe operation. A discussion about the "best" architecture would be valuable, leading the way to a discryption standard. ■