A Study of Key Management Systems in Storage Area Network

Kyawt Kyawt Khaing, Toh Liew Loong, Khin Mi Mi Aung

A*STAR, Data Storage Institute, DSI Building,5 Engineering Drive 1, (off Kent Ridge Crescent, NUS), Singapore 117608

Summary
As secure storage becomes more pervasive throughout the enterprise, the focus quickly moves from implementing encrypting storage devices to establishing effective and secure key management policies. Without the proper key generation, distribution, storage, and recovery, valuable data will be eventually compromised. How to manage keys becomes a challenging task. Adequate understanding of these new challenges is essential to effectively devise new key management policies and mechanisms to guard against them. In this paper we study key management systems and perform some scenarios used for deploying data-at-rest encryption solutions in storage area network (SAN) environment.

Key words:
Storage Security, Key Management System, Storage Area Network

1. Introduction

The recent advance in data storage technology has seen the capacity of data increased beyond the Petabytes range [5]. Nowadays, data may come from different variety of sources such as semi-structured or unstructured data. These data may contain confidential records ranging from social security numbers, credit card information to patients medical imaging and personal financial data. In order to store these data, mass storage spaces are required. Thus, confidential data stored on the mass storage devices is at risk to be disclosed to persons getting physical access to the device. To protect and secure these data, encryption methods are introduced and used. Encryption contains two fundamental components: the encryption algorithm and the key. There are standard implementations for encryption algorithms such as DES, 3DES, AES and P1619 AES-XTS [7] for encrypt data-at-rest and IPSec [8], Fibre Channel Secure Protocol (FCSP) to transport encrypted data-in-flight. Although these encryption algorithms are not new in the cryptography research, it provides a mean of reducing the risk to minimum.

Whenever we perform encryption, a key is generated or allocated by the key generator based on the specific security policies for either to encrypt or decrypt the data. To ensure that security is maintained for encryption operations, processes must be put into place that allow for complete control and security of the keys used to encrypt and decrypt the data. Thus, in storage, the focus quickly moves from implementing encrypting storage devices to establishing effective key management policies, as secure storage becomes more pervasive throughout the enterprise. Without the proper generation, distribution, storage, and recovery of keys, valuable data will be eventually compromised. Worse, without proper management of keys, data can be completely lost [6] on the disk.

To safeguard the keys, appliances based encryptions are widely adopted. For example, NeoScale System [14, 16, 11, 15, 13, 12] Key Vault has introduced a centralized policy-based management for tape media and NetApps Decru [17, 18, 19, 20, 21] has come up with Life Time Key Manager (LTKM). One of the benefits of using these appliances is that the generated keys never exposed or available to the users or to the hosts. When exporting keys out from the appliance for backup or recovery purposes, the keys are further encrypted with a master key. The master key is a quorum m of n also known as Shamir's secret sharing scheme [10, 24], since $m \leq n$ of the original n shares are needed to reconstruct the master key, where m is chosen when the master key generation is performed. Thus, extra security measures ensuring the administrator cannot recover encrypted keys by only him/herself.

Only until recently, there are a number of new methods proposed from industry vendors for example, by performing encryption at Fabric Switch, Cisco MDS 9000 family Storage Media Encryption, Brocade Encryption switch, and Hard disk drive known as Full Disk Encryption (FDE) [3, 22, 23]. Each method has its own merits and weaknesses.

While performing encryption at the appliance level and Fibre Channel Switch level, key lifecycle must be well maintained and policies must be enforced. We organize the paper as follows: In section 2, we discuss where we can do encryption in SAN, their weaknesses and strengths and a brief discussion on the encryption keys. In section 3, we discuss work on the key management as well as key management lifecycle in SAN. Emphasis will be placed on key management as keys are crucial in providing access control, authentication for the users to access the data. In

Manuscript received July 5, 2009
Manuscript revised July 20, 2009

Section 4, we discuss possible research direction and finally, we conclude in Section 5.

2. Encryptions in Storage Area Network

In Data Storage, Data-at-Rest encryption can be applied in different layer in the data centre. For example, the encryption can be performed at Host, Fibre Channel Switch, Appliance, Disk Array Controller, and Hard Disk Drive (HDD) such as Real Time Disk Encryption (RTDE) or the Full Disk Encryption drive (FDE).

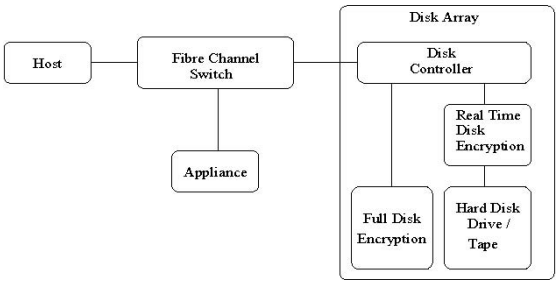


Fig.1 Encryption in SAN

Figure 1 shows a typical SAN where encryption can be applied. The following lists the advantages and disadvantages of each approach for data-at-rest in the data center:

Host: At host, encryption [1] is performed by adding the encryption cards to the host or by the encryption HBA before en-route to the disk arrays. This seems to be the benefit factor because the data is encrypted before it leaves the host. Thus, the channel between the host and the disk arrays can be fairly safe. Since the HBA encryption is based on hardware encryption on the HBA chip itself; it will not impact the CPU performances. However, there are some downsides of this approach:

- By using the encryption cards, it will depend on the operating system at host. For example, some cards can only run on Windows 2000/2003 or limited Linux build kernels. Because encryption cards reside on host, the operating system support is a limited factor.
- Data tape cartridges are normally shipped with compression capabilities or data de-duplication can save storage space in the storage arrays. If the data arrives to tape in an encrypted format, or encryption is performed before de-duplication, then the compression capabilities or the de-duplication will simply not work.
- Single point of failure. Host can be attacked or compromised. If this happens, then data could not be retrieved.
- Potential bottleneck. The host may connect by many PCs and many users may simultaneously access data through to the same host at the same time. Thus, heavy loads could result in bottleneck.

Fibre channel Switch: In the fabric, encryption is applied and then en-route to the disk arrays. The main advantage is that the cost savings over the appliance based encryption. Thus, by using this it can also eliminate the need to have extra appliances performing the encryption. The disadvantages are:

- Limit to bandwidth of the fabric. Thus, if the fabric can only support 4Gbps then the encryption engine can handle 4Gbps traffic at any time.
- Product proprietary to vendors. E.g. tied to vendors for specific fabric switch.
- Data path between host and fabric switch are not protected. Thus, open up possible data attack between host and fabric switch.

Some fabric switch vendors for example are the Cisco MDS9000 series and Nexus 7000 Series Brocade Silk-Worm series, DCX and CipherMax CM180D, CM250 and CM500.

Appliance: Encryption and Decryption are embedded inside the hardware specifically design to intercept plain data traffic from fabric switch and re-route it to the disk arrays. The advantages include operating system independent regardless from the host and load balancing distribution function for multiple appliances management. Thus, allow more data traffics from multiple fabrics. However, the disadvantages include the purchase the multiple appliances could be costly and data path between host to fabric switch and into the appliance are not protected. Thus, open up possible of data attacks. Some appliance vendors include NetApp's Decru DataFort and Ncipher's Neoscale FCDisk.

Disk Controller: The encryption and decryption hardware reside on the disk controller [21] itself. The main advantage of this is easy to implement and provide a good fit for different or mixed environment with a variety of operating systems. It also validates and eliminates the performance penalty in the server. The disadvantages include product proprietary to vendors and the data is transmitted unencrypted until it reaches the storage devices. Some Disk Controller vendors for example are the Fujitsu Eternus series and Hitachi Universal Storage Platform V (USP V) & USP VM.

Real Disk Encryption: This is a hardware based encryption chip board that sits between the disk controller and the Hard Disk Drive. The vendor that supplies this product is the X-Wall MX series from nNova Technology.

Full Disk Encryption (FDE): In Jan 2009, Trusted Computing Group (TCG) Storage Work Group released the final FDE specification which gives vendors a blueprint to develop self-encrypting storage devices that lock data, can be immediately and completely erased and can be optionally combined with the Trusted Platform

Module (TPM) for safekeeping of security credentials. Nowadays, most of the vendors, e.g. Seagate, Hitachi, Fujitsu, Western digital, carry FDE disk drives as well as participate in TCG group too.

The FDE drive security provides a range of superior benefits for protecting an enterprise systems data-at-rest when compared to current software and hardware encryption tools. Such as the performance of the encryption is not affected by the FDE drive since the encryption engine is in the disk drives and matches the drives maximum port speed, thus encryption won't slow a system down. In terms of scalability, its performance automatically scales every time storage is added in the data center.

2.1 Symmetric Key

Symmetric-key algorithms are a class of cryptography algorithms uses a simple single or one cryptography key for both encryption and decryption. The key lengths vary accordingly and it depends on the cryptography algorithms used. Some of the earlier algorithms such as the Data Encryption Standard (DES) proposed under the Federal Information Processing Standards (FIPS) in the US in 1976 had a block cipher with 64 blocks and a key length of 56 bits. Due to the key size limitation of this algorithm, DES is consider being insecure for many applications and was broken in 22 hours by the Electronic Frontier Foundation.

2.1.1 3DES

Since the vulnerability of DES algorithm due to small key size of 2^{55} is not enough to protect the brute force attacks, 3DES has been introduced to extend the key space to 168 bits (3×56 bits) without changing to another algorithm. The algorithm operates as follow:

$$DES(k_3 ;DES(k_2 ;DES(k_1 ;M)))$$

where M is the plain block to be encrypted and k_3 , k_2 , k_1 are DES keys. Although the DES has increased its key size to 168bits, the best know attack (*meet-in-the-middle*) requires 2^{32} known pairs of plain text and cipher text, 2^{88} units of storage space, 2^{113} steps, 2^{90} single DES encryption [9]. This is not currently practical and suffers from slow performance in software.

2.1.2 Serpent

Serpent, having the block size of 128 bits, can support key sizes up to 256 bits. The encryption is perform by a 32-round substitution-permutation network operating on a block of four 32 bit words and each round is applies one of eight 4 bit to 4 bit S-boxes 32 times in parallel. The Serpent consists of key-mixing XOR, 32 parallel applications of the same 4x4 S-box and a linear transformation. It is a symmetric key block cipher which was a finalist in the Advanced Encryption Standard (AES) and has not been patented. Therefore, it is can be freely used by anyone.

2.2.3 Twofish

Twofish has 128 bits block size. It can support key sizes up to 256 bits. Twofish is the variation of the block cipher Blowfish. Twofish is use of pre-computed key dependent S-boxes and a relatively complex key schedule. One half of the n-bit key is used to modify the encryption algorithm (Key-dependent S-boxes) and the other half of an n-bit key is used as the actual encryption key. On most of the software platform, Twofish perform slightly slower than AES for 128-bit keys but faster for 256-bit keys. Twofish is also one of the finalists in the AES standards and it can be freely used by anyone without any restrictions.

2.2.4 Advanced Encryption Standard (AES)

AES algorithm is a replacement of the DES algorithm in 2001 by the NIST [4] after a 5 year standardization process. AES standard key length can be 128, 192 and 256. However, it is possible to go beyond 256 bit key length. In storage security, AES 128 bit has been commonly used while the AES 256 bit becomes a norm in today's industry standards. AES is a series of linked mathematical operations using block cipher known as the substitution-permutation network and it is fast both in software and hardware implementation compared to DES.

2.2.5 Variation of AES

Since the introduction of AES algorithm which is a block of fixed length, e.g. 128 bits, several modes of operation have been proposed which allow block cipher to vary the messages of arbitrary length. The earlier modes such as the Electronic Code Book (ECB), Cipher-Block Chaining (CBC), Output Feed Back (OFB) and Cipher Feed Back (CFB) provide only either the message integrity or the confidentiality. Other modes now include the message integrity and the confidentiality such as the Integrity protection and Error Propagation (IPEP), Counter with CBC-MAC (CCM), EAX, Galois Counter Mode (GCM), and Offset Codebook Mode (OCB).

Only until recently, we have seen the modes of operation for tweakable narrow-block encryption e.g. LRW and wide-block encryption (CMC and EME) modes, designed for the disk encryption. The IEEE 1619.1[7] is an example of XEX-TCB-XTS (XTS) for Cryptographic

Protection of Data on Block-Oriented Storage Devices. Other variation such as the IEEE1619.1 (Authenticated Encryption) andIEEE1619.2 (Wide-Block Encryption) are also proposed and used in disk encryption.

2.3 Asymmetric Key

Asymmetric key is used mainly for exchange, transport or wrapping symmetric keys since asymmetric public key algorithms are relatively computationally costly in comparison with many symmetric algorithms such as AES. Thus, in storage security, we tend to use both cryptosystems for reasons of efficiency. In such a cryptosystem, symmetric keys are used to encrypt and decrypt data while in asymmetric keys, we have pair of (public and private) key. Public key is used to encrypt symmetric key while the private key is used to decrypt encrypted symmetric key. This often applies to key backup, key archiving or key restore or retrieval situations.

3. Key Management in Storage Area Network

Current Research works have been focused on key management architectures. Key management is classified into three groups:

- Centralized key management
- Distributed key management
- Hybrid key management

In centralized key management system, all key creation, re-key, modification, deletion, backup, logs & events are performed centrally. The administrator has centralized control over where each part of the key management process occurs and limits the points at which the keys and data can be accessed by users or devices that perform encryption.

The advantage of this approach is that the control over key management is easier. However, there are a number of disadvantages. Firstly, to secure the key exchange between users and centralized key manager, there are a number of methods and techniques proposed in the literature such as SSH [24], TLS [2], IPSec [8]. These methods suffer from an increase in the latency because extra algorithm must be applied to secure the communication channel. Secondly, if the centralized site is compromised by attackers, then the entire system will fail. A backup site for the centralized manager will thus be necessary. This increase the complexity of the system by introducing a backup recovery plan, which add cost to the system design. Substantial time will be required to recover from the backup in cases of failure. Lastly, the key recovery process may be slower in a centralized system because more time is required to re-establish the keys at remote site.

NeoScale System [13, 15, 10, 14] is based on a centralized policy management. It implements block-level encryption and authentication and forwards the encrypted payload to the secondary storage subsystem. In their approach, the payload must go through the CryptoStor for encryption. During the process, keys are generated either by users manually or by the random number generator. The keys are stored in CryptoStor KeyVault and never leave the appliance. KeyVault is usually deployed between bridge, router and switch. This approach suffers from two drawbacks. Firstly, all data must go through the CryptoStor for encryption. This will create a performance bottleneck. Secondly, the data are plain unsecured text before it reaches to the CryptoStor. As a result, data can be compromised prior to reaching the appliance. Similar appliance is reported for the NetApp Decru DataFort.

In distributed key management, users manage their own keys locally. This makes key recovery relatively easier compared to a centralized system. Distributed systems provide better security mechanisms in place such that if one site is compromised or down, the rest of the site are still operational. However, there is no key management policy between a central site and remote sites since each site generates its keys independently. Thus, the transfer of data from remote site to other sites is infeasible. Secondly, a secure data communication protocol must be established between sites to transfer of keys, thus, incurring overheads for using secure communication protocol.

Hybrid key management is the combination of both centralized and distributed system. The centralized key manager has a communication channel with the entire remote key manager. Key generation is still performed at the centralized key manager but key recovery is performed on remote sites.

3.1 Key Management Lifecycle for Storage Area Network

Each and every key has a sequence of states throughout its lifetime is referred to as a key management lifecycle. Essential states of key management lifecycle are as follows:

Key Generation: Key Generation is the process of generating keys for encryption and decryption. Random number generators (RNGs) are required for key generation. There are deterministic Random bit Generators (DRBGs) or pseudo RNGs and Non-Deterministic random number generators or true RNGs.

Key Distribution: There is a need to distribute the key in a secure manner to the authorized entities. The suitable options are manual distribution (e.g. smartcards) and auto distribution (e.g. electronic key distribution).

Key Archiving: The best practice is to archive the key when a key is distributed. It needs to provide both integrity and access control.

Key storage: Storing the key is one of the important processes because if unauthorized person can access the key, the security will be broken. Key should be stored where it cannot easily recover by someone as well as can retrieve without degrading the performance significantly.
Key Recovering: When the original key is lost, key recovery is needed to decrypt the encrypted information. In common, a decryption key is split into one or several parts as recovery keys and distributed these keys to escrow agents or trustees. Key encapsulation method also can be used for recovery key.
Rekeying: Rekey is the process of decrypt the entire Logical Unit Number (LUN) then encrypts the entire LUN with a new key.
Key revocation: It is required to revoke the key if the device is lost or the users who use the device quit or misuse the service. So a key revocation certificate should be generated as soon as the key is created. This certificate also must be spread to all who potentially hold the key, and as rapidly as possible.

4. Performance Analysis

In this section, we perform some scenarios used for deploying data-at-rest encryption solutions in SAN environment. In storage networking, Zoning is use to partition a Fabric into smaller subnets to restrict interference, add security and to simplify management. In the environment, the zoning performed by adding the associated host port worldwide name (PWWN) with the associated storage (PWWN) as the member of the zone.

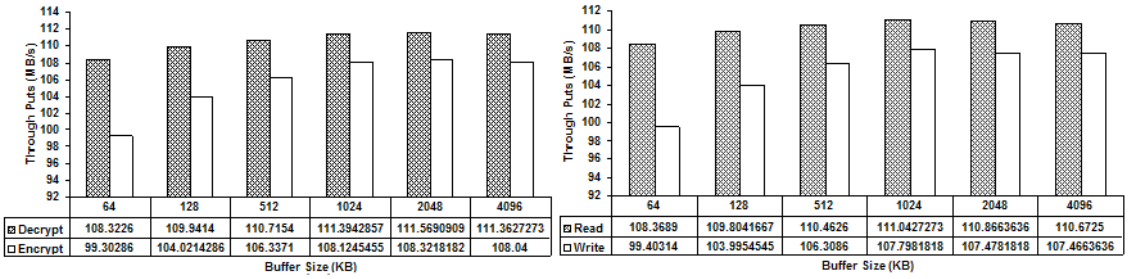
With an Encryption appliance (EA) adding into the fabric, the Zoning needs to be recreated. EA is built with two Host Bus Adapters (HBA). One of the HBA Card is act as the target while the another HBA card acts as the initiator. By plugging the EA in the fabric, two zoning sets need to be performed in the FC switch; first zoning was done with the Host (PWWN) and EA target HBA (PWWN) as the members and second zoning was done with the EA initiator HBA (PWWN) and Storage (PWWN) (Figure 2). Then completed zone sets are activated, accordingly.

One of the biggest challenges with encryption is the performance hit. Encryption is a difficult problem, given the performance issues and the management complexity. We perform a performance evaluation of a commercial encryption appliance. The focus of this testing was restricted to performance analysis of the appliance. No other stress, stability or data integrity testing was performed. The Medusa Labs Tool Suite is a publicly available set of performance and data integrity tools that were used for the raw device portion of the testing. The medusa labs test tools was used, with the specific tool Pain utilized for raw device tests. All patterns were run in a series IOPS and throughput test cases that were designed to determine the best case capabilities of the appliance. The tests included a series of write/re-read, random write and random read workload, with and without encryption. Each workload included the following I/O test cases.

The medusa labs test tools tests consisted of sequential write, read and 50/50 write/read workloads. These test cases were performed on a raw block device.

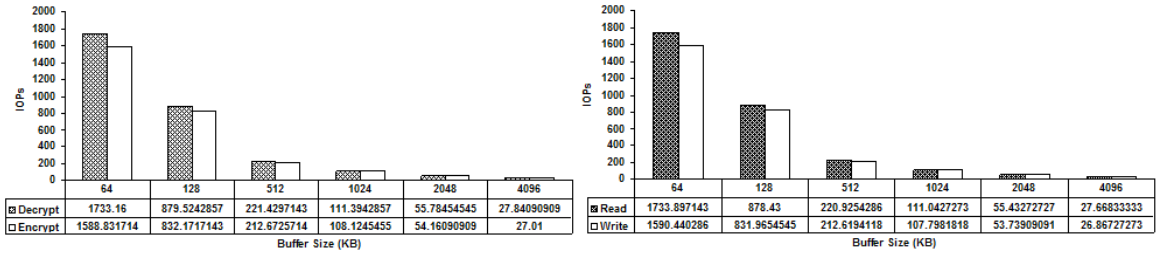
We perform performance analysis of encryption and decryption with various buffer sizes from 64 KB to 4 MB. We also run the same way for clear text read and write. Figure 3 and 4 illustrates the different throughputs and IOPs between with EA and without EA. The benefit of encryption appliance is that they are doing this in hardware; they can do it at line speed.

Fig.2 Zoning Configuration of Host and Encryption Appliance



(a) (b)

Fig. 3 (a) Throughputs with EA (b) Throughputs without EA



(a) (b)

Fig. 4 (a) IOPs with EA (b) IOPs without EA

5. Conclusion

This is a preliminary study of key management system. We have identified possible areas of research directions and our further work is to further explore. We have introduced and studied the area of storage security emphasizing on encryption and key management systems. Although a considerable amount of research has dedicated to encryption algorithms, key management is still an issue. We have outlined the importance of key management system in storage. Re-key is a process of decrypt the entire Logical Unit Number (LUN) then encrypt with a new key. This result in performance degradation thus, affecting the overall performance of the system. The advantages of encryption in the SAN include centralization and heterogeneity. It should be transparent and no disruptive. The benefit of the encryption appliance is that they are doing this in hardware; they can do it at line speed.

Acknowledgments

This work was supported by the grant no "DSI/08-100002" under Network Storage Technology Division, Data Storage Institute, A*STAR.

References

[1] A. Baldwin and S. Shiu. "Encryption and key management in a SAN", Proceedings of the First International IEEE Security in Storage Workshop (SISW'02), pages 1–10, 2002.

[2] T. Dierks and C. Allen. „The tls protocol version 1.0", Technical Report, The Internet Engineering Task Force IETF, 1999.

[3] D. Dunn. "Seagate expands full disk encryption to data center hard drives", Information Week Magazine, pages 1–2, October 2007.

[4] FIPS-197 "Advanced encryption standard (AES)", Federal Information Processing Standards Publication 197, pages 1–51, November 2001.

[5] J. F. Gantz, C. Chute, A. Manfrediz, S. Minton, D. Reinsel, W. Schlichting, and A. Toncheva "The diverse and exploding digital universe: An updated forecast of worldwide information growth through 2011", IDC White paper, pages 1–16, March 2008.

[6] W. Hubis "An introduction to key management for secure storage", http://www.snia.org/education/tutorials/2008/spring/security/Hubis-W_Introduction_to_Key_Management.pdf SNIA: Storage Security Presentation Slides, pages 1–55, April 2008.

[7] IEEE-P1619 "Ieee p1619 standard for cryptographic protection of data on block-oriented storage devices draft 14", IEEE Standards, pages 1–32, March 2007.

[8] IPSec Working Group "Ip security protocol(ipsec)", Technical Report, The Internet Engineering Task Force IETF, 2002.

[9] S. Lucks "Attacking triple encryption", Lecture Notes in Computer Science: LNCS 1372, 1372:239–253, 1998

[10] NeoScale "Privacy compliance: Tape media protection and data privacy issues", NeoScale System: White paper, pages 1–6, May 2004.
[11] NeoScale "Achieving pci compliance with storage security solutions" NeoScale System: White paper, pages 1–8, 2006.
[12] NeoScale "Criteria for selecting and deploying a global key management solution", NeoScale System: White paper, pages 1–9, 2006.
[13] NeoScale "Global key management for storage security encryption" NeoScale System: White paper, pages 1–20, 2006.
[14] NeoScale "Protecting tape-based storage", NeoScale System: White paper, pages 1–12, 2006.
[15] NeoScale "The next evolution in global key management", NeoScale System: White paper, pages 1–8, April 2007.
[16] NetApp-Decru "San security threats", White paper: Decru proprietary, page 20, 2003
[17] NetApp-Decru "Netapp-decru compliance solutions: Advanced file shredding and security", White paper, pages 1–3, 2005.
[18] NetApp-Decru "Decru datafort storage security appliances", White paper, pages 1–4, 2006
[19] NetApp-Decru "Decru lifetime key management appliance: Secure, automated, enterprise-wide key archival and recovery", White paper, pages 1–2, 2007.
[20] NetApp-Decru "Decru lifetime key management software" , White paper, pages 1–2, 2007.
[21] G. Schulz. "Decrypting enterprise storage security: Trends and options for securing enterprise data and storage", Industry Trends and Technology Perspective White paper, pages 1–5, December 2006.
[22] Secude. Protecting digital assets: Full disk encryption white paper. White paper, pages 1–13, 2007.
[23] T. Wong, C. Wong, and J. M. Wing. Verifiable secret redistribution for threshold sharing schemes. Technical Report CMU-CS-02-114-R, Carnegie Mellon University, October 2002.
[24] T. Ylonen, T. Kiviene, M. Saarinen, T. Rinne, and S. Lehtinen. Ssh protocol architecture. Technical Report, The Internet Engineering Task Force IETF, 2002