SHORT PAPER: LINCOS - A Storage System Providing Long-Term Integrity, Authenticity, and Confidentiality[*]

Johannes Braun, Johannes Buchmann, Denise Demirel, Matthias Geihs
TU Darmstadt, Germany

Mikio Fujiwara, Shiho Moriai, Masahide Sasaki, Atsushi Waseda
NICT, Japan

ABSTRACT

The amount of digital data that requires long-term protection of integrity, authenticity, and confidentiality grows rapidly. Examples include electronic health records, genome data, and tax data. In this paper we present the secure storage system LINCOS, which provides protection of integrity, authenticity, and confidentiality in the long-term, i.e., for an indefinite time period. It is the first such system. It uses the long-term integrity scheme COPRIS, which is also presented here and is the first such scheme that does not leak any information about the protected data. COPRIS uses information-theoretic hiding commitments for confidentiality-preserving integrity and authenticity protection. LINCOS uses proactive secret sharing for confidential storage of secret data. We also present implementations of COPRIS and LINCOS. A special feature of our LINCOS implementation is the use of quantum key distribution and one-time pad encryption for information-theoretic private channels within the proactive secret sharing protocol. The technological platform for this is the Tokyo QKD Network, which is one of worlds most advanced networks of its kind. Our experimental evaluation establishes the feasibility of LINCOS and shows that in view of the expected progress in quantum communication technology, LINCOS is a promising solution for protecting very sensitive data in the cloud.

1. INTRODUCTION

1.1 Motivation and problem statement

Today large amounts of data are digitally stored, increasingly in cloud-based data centers, and this amount will massively grow in the future. For example, Japanese hospitals use redundant cloud storage to protect sensitive medical data from loss due to natural catastrophes [17]. Also, in his state of the union address 2015, the U.S. President Barack Obama announced a Precision Medicine Initiative which will require to digitally store the health data of virtually all U.S. citizens.

Protection requirements. Digitally stored data require protection throughout their whole lifetime which may be very long. Important protection goals are *integrity*, *authenticity*, and *confidentiality*. Integrity means that illegitimate and accidental changes of the data can be discovered. Authenticity refers to the origin of the data being identifiable. Confidentiality guarantees that only authorized parties are able to access the data. For example, consider medical data. Their integrity is extremely important because changes may lead to incorrect treatment with serious health consequences. Authenticity is required for liability reasons and confidentiality protects the privacy of the involved individuals. Medical data may have to be kept as long as the respective patients are alive or even beyond this time. So the required protection period may be more than 100 years. Other examples for sensitive long-lived data are genome data, governmental secrets, and tax data.

Current cryptography is unsuitable. Unfortunately, current technology does not provide integrity, authenticity, and confidentiality protection over such a long time. The cryptographic algorithms used today for such protection, such as AES encryption and RSA signatures, fail to provide sufficient security guarantees. They are *complexity-based* which means that their security relies on the intractability of certain algorithmic problems, e.g., integer factorization. However, cryptanalytic power is steadily increasing. According to Moore's law, the computing speed doubles every 18 months. Also, there is algorithmic progress. Hence, keys chosen today will be too short in the future. For example, in their original RSA paper [22], the authors estimate the required RSA modulus size: "using 200 digits provides a margin of safety against future developments." However less than 30 years later factoring 200 decimal digit numbers became feasible [2]. This situation is very critical. Adversaries may store encrypted data now and decrypt them later when the encryption algorithm becomes broken which may happen during the lifetime of the protected data. Technologically, this appears to be quite feasible. For instance, the Utah Data Center of the NSA has an estimated capacity of

[*]A full version of this paper is available as ePrint report 2016/742. This work has been co-funded by the DFG as part of project S6 within the CRC 1119 CROSSING, by the European Union's Horizon 2020 research and innovation program under Grant Agreement No 644962, and by the ImPACT Program of the Council for Science, Technology and Innovation (Cabinet Office, Government of Japan).

ASIA CCS '17, April 02-06, 2017, Abu Dhabi, United Arab Emirates
© 2017 ACM. ISBN 978-1-4503-4944-4/17/04...$15.00
DOI: http://dx.doi.org/10.1145/3052973.3053043

4 to 12 Exabytes (10^{18} bytes) which allows to store huge amounts of encrypted data for a long time.

The question arises whether it is possible and feasible to provide long-term protection of integrity, authenticity, and confidentiality of digital data. Here and in the remainder of this paper we define long-term protection as protection for an indefinite time period.

There exist several partial solutions to this problem. For overviews of confidentiality and integrity/authenticity related solutions see [5] and [27], respectively. These surveys also contain the relevant references.

Confidentiality of data in transit. In 1949 Claude Shannon presented his model of *information-theoretic confidentiality* protection and proved that *one-time-pad encryption* (OTP) provides such protection for transmitted data (*data in transit*). However, OTP keys are as long as the protected data, can only be used once, and are required to be exchanged in an information-theoretically secure fashion. Therefore, OTP encryption has been only used for special applications such as military applications with key exchange by trusted couriers. In the past decades, other methods of such key exchange have been developed, including schemes based on the *bounded storage*, *noisy channel*, or *limited access* models, and *quantum key distribution (QKD)*. Among these options, QKD is by far the most advanced, both theoretically and experimentally. For example, many countries such as Austria, China, Japan, Switzerland, and the USA are currently deploying QKD-protected backbones. For example, they use QKD to protect keys for complexity-based symmetric encryption. However, in this case no information-theoretic security is achieved.

Confidentiality of data at rest. Unfortunately, OTP encryption is unsuitable for stored data. This is because OTP requires using and protecting one-time keys that are as long as the original data. Hence, nothing is gained by using OTP. Instead, *proactive secret sharing* can be used to provide information-theoretic confidentiality protection of stored data. Proactive secret sharing decomposes the secret into n shares in such a way that a threshold number $k \leq n$ of shares is required to reconstruct the secret while any smaller number of shares reveals no information about the secret. The shares are renewed on a regular basis in order to prevent attacks of *mobile adversaries* who may be able to learn more and more shares over time. Such solutions are well suited for cloud storage systems and are already used in this context [19]. However, as in currently used secret sharing solutions communication protection is only complexity-based, they do not provide information-theoretic confidentiality.

Integrity and authenticity. There are standardized solutions for long-term integrity and authenticity protection (see [11]) which are already used in practice. They utilize timestamp chains to prolong the validity of complexity-based digital signatures thereby protecting integrity and authenticity for any length of time. However, these solutions prohibit long-term confidentiality protection. This is because they submit hashes of the protected data to timestamp authorities. As cryptographic hash functions only offer complexity-based security, they may leak information over time. This is also why the solutions in [16] and [21] do not support long-term confidentiality protection.

In summary, the problem of long-term protection of integrity, authenticity, and confidentiality of digital data is urgent and a comprehensive solution that provides such protection is not known so far.

1.2 Contribution

In this paper we present the first storage solution that simultaneously protects integrity, authenticity, and confidentiality of digital data for an indefinite period of time. We analyze its security and experimentally study its feasibility. As our solution uses a distributed storage system, it is suitable for cloud applications.

Confidentiality-preserving long-term integrity protection. Our first contribution is the new scheme COPRIS. It is the first long-term integrity scheme that is confidentiality preserving, i.e., it does not leak any information about the protected data to third party services. It also provides authenticity protection if the protected data is signed and the signature is protected together with the data. The idea in COPRIS is to no longer timestamp the protected documents. Instead, information-theoretically hiding commitments to these documents are timestamped. These commitments never leak any information about the documents. Information-theoretically hiding commitments can only be computationally binding [4]. Therefore, commitments are renewed on a regular basis.

Long-term integrity, authenticity, and confidentiality protection. Our second contribution is the secure storage system LINCOS. It is the first storage system that simultaneously protects integrity, authenticity, and confidentiality of stored data in the long-term. We present a security analysis and report on our implementation and thorough experimental evaluation of LINCOS.

In LINCOS, a document owner communicates with an *integrity system* and a *confidentiality system.* The integrity system is based on COPRIS which we implemented as a Java application. The confidentiality system uses private channels and proactive secret sharing. We realize information-theoretically secure private channels using QKD and OTP in the Tokyo QKD Network. This network is one of worlds most advanced QKD networks and allows for a reliable feasibility study.

Experimental evaluation. We report on an experiment that simulates protecting documents of different sizes for 100 years (Appendix C). Integrity protection of our solution is very efficient in storage space and computation. In the case of confidentiality protection, the limiting factor turns out to be the speed of QKD key generation. The average key supply that we currently achieve is 40 kb/s. So transmitting 1 GB of data requires 2.3 days of prior key accumulation. This allows for proactive secret sharing of 158 GB with a share renewal period of 2 years. However, in the near future key rates of 1 Mb/s can be expected which will reduce the time for distributing a 1 GB key to 2.2 hours. Thus it will be possible to protect 4 TB with a share renewal period of 2 years. For example, 4 TB is the size of the genomes of roughly 5000 persons.

LINCOS is well suited for long-term storage systems. Availability requirements in such systems, in particular in case of natural or other catastrophes, suggest to redundantly store the data in multiple locations which are far apart from each other. In fact, redundant storage is already common practice in many scenarios (e.g., [17]). LINCOS can be used in these scenarios and additionally achieves long-term integrity, authenticity, and confidentiality protection.

2. CRYPTOGRAPHIC COMPONENTS

In this section we give a brief overview of the cryptographic components used in COPRIS and LINCOS. For a more detailed description we refer to the full paper. Cryptographic components may provide computational security or information-theoretic security. Some components, such as commitment schemes, may even have both properties for different functionalities. A computationally secure component is usually parametrized with a security parameter which determines the hardness of the underlying computational problem. The security parameter is chosen such that the cryptographic component remains secure for the intended usage period.

Timestamps. Timestamps are issued by timestamp services using *timestamp schemes* [13, 1]. A timestamp scheme involves a protocol Stamp for retrieving timestamps and an algorithm Verify for verification of timestamps. We require timestamps to be computationally *unforgeable* [9].

Authenticated channels. An *authenticated channel* is a mutually authenticated connection between a sender and a receiver. We require an authenticated channel to guarantee computationally secure mutual authentication of the sender and the receiver [3].

Private channels. We also use *private channels.* In addition to computationally secure mutual authentication, private channels also provide information-theoretic confidentiality of the transmitted data [26].

Commitment schemes. A *commitment scheme* allows a party to commit to some document without revealing it. A commitment scheme consists of algorithms Commit for creating a commitment and Verify for revealing the commitment and verifying its validity. We require commitment schemes to be *computationally binding* and *information-theoretically hiding* [20, 10].

Proactive secret sharing. A proactive secret sharing scheme allows a dealer to distribute a secret among a set of shareholders such that each shareholder does not learn anything about the secret. Protocol Share is used to distribute the secret, protocol Reshare is used to renew the shares to protect against mobile adversaries who successfully attack one shareholder after another over time, and protocol Retrieve is for retrieving the secret from the shareholders. We require proactive secret sharing schemes to provide information-theoretic confidentiality in the *mobile adversary model* [14].

3. COPRIS: CONFIDENTIALITY PRESERVING LONG-TERM INTEGRITY SCHEME

In this section we present our first contribution of this paper: the scheme COPRIS which ensures long-term integrity and authenticity protection and is long-term confidentiality preserving, i.e., it does not leak any information about the protected data. The security of COPRIS is discussed in Appendix A.

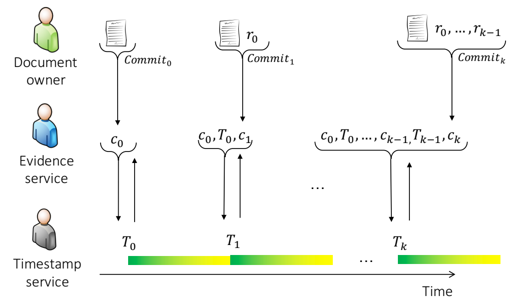

Figure 1: Schematic of COPRIS.

Figure 1 illustrates the functionality of COPRIS. The setting is as follows. A document owner stores a document d at some time t. He keeps d secret and constructs a *proof of integrity* PI for d. Later he may choose to reveal d to another party. This party then uses PI to verify that d existed at time t. To preserve the confidentiality of d, the proof of integrity PI is constructed in such a way that no information about d is revealed to third parties involved in the construction process. Confidential storage of the secret data is out of the scope of COPRIS and is dealt with in LINCOS (Section 4).

We now explain the construction of the proof of integrity and its verification. The integrity proof is a pair (E, R), where E is an *evidence record* and R is a list of decommitment values. The evidence record is constructed interactively between the document owner and an *evidence service* which, in turn, interacts with a timestamp service. The list of decommitment values is constructed and kept secret by the document owner. The document owner may decide to reveal the decommitment values together with the document to a *verifier*. In the following we describe the protocols of COPRIS. For more details see the full paper.

Initial protection. The initial integrity proof is constructed as follows. The document owner runs algorithm Protect. Input is the document d and the (initially empty) list of decommitment values R. He selects a commitment scheme CS and computes a commitment $(c, r) \leftarrow \mathsf{CS.Commit}(d)$. He sets $R = (r)$ and sends the commitment value c to the evidence service. When the evidence service receives c, it runs algorithm AddEv. Input is c and the (initially empty) evidence record E. It requests a timestamp T on c from a timestamp service TS using protocol TS.Stamp at time t. The first evidence record is $E = (c, T, t)$.

Timestamp renewal. Before the last timestamp becomes insecure it must be renewed. In this case, the evidence service executes algorithm RenewTs, where the input is the current evidence record E. It selects a new timestamp scheme TS and obtains a timestamp T on E at time t using protocol TS.Stamp. Then, it appends (c, T, t) to E, where c is the last commitment value contained in E.

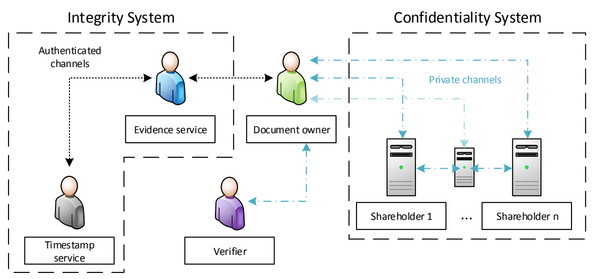

Figure 2: Schematic of LINCOS.

Commitment renewal. Before the last commitment created by the document owner becomes insecure, it must be renewed. The document owner runs the algorithm RenewCom. Input is the document d and the decommitment value list R. The document owner selects a new commitment scheme CS and computes $(c, r) \leftarrow \mathsf{CS.Commit}(d, R)$. He appends r to R and sends c to the evidence service. When the evidence service receives c, it runs algorithm AddEv. Input is c and the evidence record E.

Verification. When the document owner reveals d to the verifier, he also transmits the asserted existence time t and the integrity proof (E, R). Using this information, the verifier can validate the existence of d at time t as follows. Let $R = (r_0, \ldots, r_n)$ and $E = (c_0, T_0, t_0, \ldots, c_n, T_n, t_n)$.

We describe the verification procedure. We define t_{n+1} to be the time of verification and for $i \in \{0, \ldots, n\}$ we set $E_i = (c_0, T_0, t_0, \ldots, c_i, T_i, t_i)$ and $R_i = (r_0, \ldots, r_i)$. Furthermore, for $i \in \{0, \ldots, n\}$ let CS_i denote the commitment scheme associated with c_i and TS_i the timestamp scheme associated with T_i. The verifier uses his trust anchor TA to verify that

$$\mathsf{TS}_i.\mathsf{Verify}(TA, (E_{i-1}, c_i), T_i, t_i; t_{i+1}) = 1$$

and

$$\mathsf{CS}_i.\mathsf{Verify}(TA, (d, R_{i-1}), c_i, r_i; t_{i+1}) = 1 \ ,$$

for $i \in \{0, \ldots, n\}$. The trust anchor contains the required root certificates and commitment scheme parameters.

4. LINCOS: SYSTEM FOR LONG-TERM INTEGRITY, AUTHENTICITY, AND CONFIDENTIALITY

In this section we describe our new long-term storage system LINCOS which provides information-theoretic confidentiality and long-term integrity and authenticity protection. The security of LINCOS is discussed in Appendix B.

An overview of LINCOS is shown in Figure 2. It consists of an *integrity system*, which is based on COPRIS, for constructing an integrity proof PI, and a *confidentiality system*, which is based on private channels and secret sharing, for information-theoretic confidential storage of the secret data. The involved parties are a *document owner*, an *evidence service*, a *timestamp service*, a set of *shareholders*, and a *verifier*. These parties are connected by private or authenticated channels as shown in Figure 2. While LINCOS is running, the respective channels are instantiated securely whenever a connection is established. For integrity proof construction we use the same notation as in COPRIS, that is, the document owner maintains a list of decommitment values R and the evidence service maintains an evidence record E.

Initial document protection. For initial protection of a document d, the document owner runs COPRIS.Protect. Input is a document d and the (initially empty) list of decommitment values R. The document owner chooses a confidentiality system involving several shareholders. The document owner uses protocol Share to distribute (d, R) among the shareholders.

Renewal of timestamps. COPRIS requires timestamp renewal on a regular basis. For this, the evidence service runs COPRIS.RenewTs.

Renewal of commitments. COPRIS also requires commitment renewal on a regular basis. For this, the document owner does the following. First, he retrieves d and the sequence of decommitment values R from the confidentiality system by running protocol Retrieve. Then, he runs the algorithm COPRIS.RenewCom, thereby updating the list of decommitment values R and the evidence record E. Finally, the document owner selects a potentially new confidentiality system and runs protocol Share to distribute the document d and the updated sequence of decommitment values R among the shareholders in the confidentiality system.

Renewal of secret shares. The shares stored by the shareholders are renewed on a regular basis. This prevents a mobile adversary to take advantage of shares he may have been able to obtain in the past. In this process, the current set of shareholders of the confidentiality system may also be replaced by a new set of shareholders operated by the same confidentiality system. This resharing is done by running protocol Reshare.

Verification. When the document owner decides to reveal the document d to a verifier and prove that it existed at time t, he executes the following steps. He requests the current evidence record E from the evidence service. He also retrieves the document d and the list of decommitment values R from the confidentiality system by running the protocol Retrieve. He sends the document d, time t, evidence record E, and the list of decommitment values R to the verifier over a private channel. The verifier uses his trust anchor TA and checks that $\mathsf{COPRIS.Verify}(TA, d, t, E, R) = 1$. This proves that d existed at time t and has not been changed.

5. IMPLEMENTATION

In this section we describe our implementation of LINCOS. LINCOS uses COPRIS for its integrity system and proactive secret sharing combined with appropriate private channels for its confidentiality system. One important feature of our implementation is the possibility of replacing cryptographic components. This is required because of Assumptions I1 and I2. Another feature is the realization of private channels using the Tokyo QKD Network [23].

5.1 Implementation of COPRIS

We implemented COPRIS using Java following the specification given in Section 3. Here we focus on the selection of the cryptographic schemes.

Commitment scheme. As commitment scheme we use the Pedersen commitment scheme [20]. This scheme is computationally binding and information-theoretically hiding (Assumptions I1 and C1). It is parametrized with two prime numbers p and q and its binding security is based on the discrete logarithm problem. We use the hash-then-commit approach to allow for committing to data of arbitrary length. Our implementation uses the SHA-2 hash function family. The hash function and the parameters of the commitment scheme need to be chosen such that computational bindingness is achieved for the intended usage period (Assumption I1). In practice, these choices can be made on the basis of trustworthy recommendations. For an overview of recommendations see [15].

Timestamp scheme. The timestamp service used by the evidence service is implemented in accordance with standard RFC 3161 [1]. Implementing it requires choosing a hash function and a digital signature scheme. We use the SHA-2 hash function family and the RSA digital signature scheme. The security of the used RSA instance depends on the bitlength of the RSA-modulus. Hash function and RSA-modulus need to be chosen such that unforgeability is achieved within the usage period of the timestamp scheme (Assumption I2).

Authenticated channels. Authenticated channels are realized using TLS [7], instantiated such that computationally secure mutual authentication is achieved. We do not discuss parameter choices for TLS because they do not affect our measurements presented in Section C.

5.2 Secret sharing and private channels

We describe the implementation of the confidentiality system of LINCOS that consists of private channels and proactive secret sharing.

Private channels. LINCOS uses private channels to connect the document owner with the shareholders. By Assumption C2, these channels are required to provide information-theoretic confidentiality and computational authenticity. For establishing such private channels we use the Tokyo QKD Network [23], which is shown in Figure 3. A combination of Wegman-Carter authentication, QKD, and OTP encryption is used to achieve information-theoretic private and authenticated channels [24]. The network consists of three layers; the *quantum layer*, the *key management layer*, and the *application layer*. Secret sharing is run on the application layer. Parties on the application layer request and receive key material from the key management layer. The key management layer establishes an interface to the quantum layer where the raw key material is generated using QKD technology. To improve the capabilities of the network, keys are relayed on the key management layer by key management agents. In order to allow for Assumption C2 to hold, further technical protection measures are in

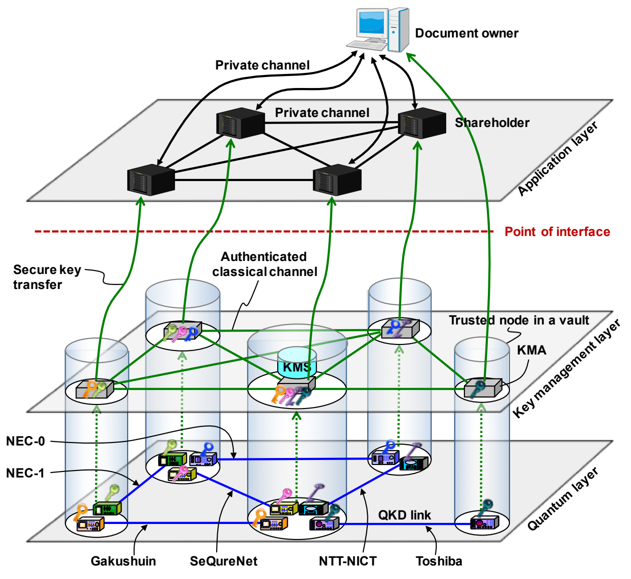

Figure 3: The secret sharing scheme supported by the Tokyo QKD Network.

place. Further details on the QKD network can be found in the full paper and [23, 8].

Secret sharing. Our implementation of secret sharing uses Shamir's secret sharing [25]. It provides information-theoretic confidentiality for the stored data (Assumption C3). We use a (3,4)-threshold secret sharing, suiting the network structure of the Tokyo QKD Network. This means that the document owner distributes shares to 4 shareholders and 3 shareholders are needed for the reconstruction of the data. To allow for sharing data of arbitrary size, these data are decomposed into parts of appropriate size. Our implementation supports a basic resharing protocol involving the document owner. The document owner first retrieves and reconstructs the data and then generates and distributes new shares. We assume that resharing happens before the adversary corrupts more than 2 shareholders (Assumption C4). In the future we plan to implement proactive secret sharing as suggested in [14] for resharing without the document owner.

6. CONCLUSION

Our experimental evaluation (Appendix C) shows that the long-term integrity system based on COPRIS has very good performance, in particular in view of the expected growth of computing power; the time and space cost for time-stamping commitments instead of hash values and for renewing these commitments is negligible. As expected, information-theoretic confidentiality protection is expensive. One limiting factor is the additional space required by secret sharing. However, it does not exceed the additional storage space required by cloud storage solutions that use secret sharing for robustness reasons. The second limiting factor is QKD. It is technically complex and transmission rates are not yet fully satisfactory. On the positive side the development in this area is promising so that practical solutions can be expected in the future.

The data that can be protected given that resharing happens every two years has maximum size $\mathsf{size}_s = 2$ years $*$ $\mathsf{keyRate}_{\mathsf{QKD}}/2 = 1$ year $*$ $\mathsf{keyRate}_{\mathsf{QKD}}$. For the current key supply throughput of 40 kb/s we obtain $\mathsf{size}_s = 158$ GB. This data size approximately corresponds to human genomic data of 195 persons. In the near future (4 to 5 years), QKD technology with key rates of 1 Mb/s over 50 km is expected to be available. Then, data of size up to 3942 GB can be protected, which is roughly 4 TB or the size of the genomes of 4926 persons. If the key supply throughput can be increased to 1 Gb/s, data of size 4 PB can be handled, which corresponds to human genomic data of 4.9 million persons. Such a QKD performance can be expected to be realizable using dense wavelength division multiplexing of 1000 quantum channels as well as fast key distillation processing. This is a challenge, but will be feasible by employing integrated photonic technologies and dedicated key distillation engines on semiconductor chips.

There are two main directions of further research. The first concerns improvement of QKD performance. As mentioned, dense wavelength division multiplexing of many quantum channels combined with fast key distillation processing is promising. The second research direction concerns the performance of the commitment renewal and resharing process. Currently, in both processes the document owner is required to retrieve the document regularly. However, it is desirable to take the document owner out of the loop and let the confidentiality and integrity systems deal with these issues independently. For proactive secret sharing, we will use a more advanced resharing protocol, e.g., [12]. It allows for renewing the shares without the help of the document owner. We also aim at developing a commitment renewal protocol that does not involve the document owner.

7. REFERENCES

[1] C. Adams, P. Cain, D. Pinkas, and R. Zuccherato. Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). RFC 3161 (Proposed Standard), Aug. 2001. Updated by RFC 5816.

[2] F. Bahr, M. Boehm, J. Franke, and T. Kleinjung. Factorization of RSA-200. *Public announcement on May 9th*, 2005.

[3] M. Bellare and P. Rogaway. Entity authentication and key distribution. In D. R. Stinson, editor, *CRYPTO' 93*, pages 232–249, 1994.

[4] G. Brassard, C. Crépeau, D. Mayers, and L. Salvail. A brief review on the impossibility of quantum bit commitment. *arXiv preprint quant-ph/9712023*, 1997.

[5] J. Braun, J. Buchmann, C. Mullan, and A. Wiesmaier. Long term confidentiality: a survey. *Designs, Codes and Cryptography*, 71(3):459–478, 2014.

[6] R. Canetti, L. Cheung, D. K. Kaynar, N. A. Lynch, and O. Pereira. Modeling computational security in long-lived systems. In *CONCUR*, 2008.

[7] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), Aug. 2008. Updated by RFCs 5746, 5878, 6176, 7465, 7507, 7568, 7627, 7685.

[8] M. Fujiwara, A. Waseda, R. Nojima, S. Moriai, W. Ogata, and M. Sasaki. Unbreakable distributed storage with quantum key distribution network and password-authenticated secret sharing. *Scientific Reports*, 6, 2016.

[9] M. Geihs, D. Demirel, and J. Buchmann. A security analysis of techniques for long-term integrity protection. In *Privacy, Security and Trust 2016*, 2016.

[10] O. Goldreich. *Foundations of Cryptography – Volume 1*, chapter Perfectly Hiding Commitment Schemes. Cambridge University Press, 2001.

[11] T. Gondrom, R. Brandner, and U. Pordesch. Evidence Record Syntax (ERS). RFC 4998 (Proposed Standard), Aug. 2007.

[12] V. H. Gupta and K. Gopinath. G_{its}^2 VSR: An information theoretical secure verifiable secret redistribution protocol for long-term archival storage. In *Security in Storage Workshop*, 2007.

[13] S. Haber and W. S. Stornetta. How to time-stamp a digital document. In *CRYPTO' 90*, pages 437–455, 1990.

[14] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. *CRYPTO '95*, chapter Proactive Secret Sharing Or: How to Cope With Perpetual Leakage, pages 339–352. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995.

[15] https://www.keylength.com. Cryptographic key length recommendation, 2016.

[16] D. Hühnlein, U. Korte, L. Langer, and A. Wiesmaier. A comprehensive reference architecture for trustworthy long-term archiving of sensitive data. In *Conference on New Technologies, Mobility and Security*, pages 1–5, Dec 2009.

[17] T. Kuroda et al. Simulating cloud environment for HIS backup using secret sharing. *Studies in health technology and informatics*, 192:171–174, 2012.

[18] A. K. Lenstra. Key lengths. In *The Handbook of Information Security*. Wiley, 2004.

[19] T. Loruenser, A. Happe, and D. Slamanig. Archistar: Towards secure and robust cloud based data sharing. In *CloudCom 2015*, pages 371–378, Nov 2015.

[20] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO '91*, 1992.

[21] T. A. Ramos, N. da Silva, L. C. Lung, J. G. Kohler, and R. F. Custódio. An infrastructure for long-term archiving of authenticated and sensitive electronic documents. In *EuroPKI*, pages 193–207, 2010.

[22] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, Feb. 1978.

[23] M. Sasaki et al. Field test of quantum key distribution in the tokyo qkd network. *Opt. Express*, 19(11):10387–10409, May 2011.

[24] V. Scarani et al. The security of practical quantum key distribution. *Rev. Mod. Phys.*, 81:1301–1350, Sep 2009.

[25] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, Nov. 1979.

[26] C. E. Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4):656–715, Oct 1949.

[27] M. A. G. Vigil, J. A. Buchmann, D. Cabarcas, C. Weinert, and A. Wiesmaier. Integrity, authenticity, non-repudiation, and proof of existence for long-term archiving: A survey. *Computers & Security*, 50:16–32, 2015.

APPENDIX

A. SECURITY OF COPRIS

We show that COPRIS provides long-term integrity and authenticity protection and that no confidential data is leaked to the evidence and timestamp service. For this, we consider adversaries that may be active for an unbounded period of time, but can only do a bounded amount of work per unit of real time. This allows us to use computationally secure cryptographic primitives for a limited time period in the presence of an adversary who is overall unbounded. We refer to the full paper and [6, 9] for more details regarding this adversary model.

By long-term integrity and authenticity of COPRIS we mean that it is infeasible for an adversary as described above to forge an integrity proof, that is, to present a valid integrity proof for a document d and a time t even though d did not exist at time t. If the document is protected together with a digital signature, then long-term integrity achieves long-term authenticity. A more formal definition of long-term integrity is given in [9]. It is essential for the security of COPRIS that the following assumptions hold.

I1. The commitment schemes used in the proof of integrity are computationally binding in their usage period.

I2. The timestamp schemes used in the proof of integrity are computationally unforgeable in their usage period.

I3. The verifier has a valid trust anchor.

The *usage period* of a cryptographic scheme is defined as the time interval starting when the scheme is chosen and ending when it is replaced by a new scheme. By a valid trust anchor we mean a trust anchor that allows for the verification of all timestamps and commitments. The following theorem states that under the above assumptions, COPRIS provides long-term integrity protection.

Theorem A.1. *Under Assumptions I1, I2, and I3,* COPRIS *provides long-term integrity and authenticity.*

Note that there is a small security loss over time as the success probabilities of the adversary for each time period add up. For more details see [6, 9]. Next, we show that COPRIS is confidentiality preserving in the long-term, i.e., no information is leaked to the evidence and timestamp service (in an information-theoretic sense). This fact relies on the following assumption.

C1. The commitment schemes are information-theoretically hiding.

Theorem A.2. *Under assumption C1,* COPRIS *is information-theoretic confidentiality preserving.*

For proofs of Theorems A.1 and A.2 we refer to the full paper.

B. SECURITY OF LINCOS

We show that under appropriate assumptions, LINCOS provides integrity protection for an indefinite period of time and information-theoretic confidentiality protection. Adversaries are assumed to have the capabilities described in Section A. They run forever but are computationally bounded per unit of time. In addition, adversaries are assumed to be active and mobile. This means that adversaries may eavesdrop on channels or corrupt shareholders. A more detailed discussion of this model can be found in [14].

Security year	**SHA-2** instance	**RSA** $\log_2(n)$	**Pedersen** $\log_2(p)$, $\log_2(q)$
2040	SHA-224	2048	2048, 224
2065	SHA-224	3072	3072, 224
2085	SHA-256	4096	4096, 256
2103	SHA-384	5120	5120, 384
2116	SHA-384	6144	6144, 384

Table 1: Parameter selection according to Lenstra [18].

Integrity. Theorem A.1 states that in this adversary model, LINCOS provides long-term integrity and authenticity protection if Assumptions I1, I2, and I3 from Section A are satisfied.

Confidentiality. We say that LINCOS provides information-theoretic confidentiality protection if an adversary with capabilities as described above cannot recover any information about the stored document in an information-theoretic sense.

For information-theoretic confidentiality we require Assumption C1 from Section A and the following assumptions to hold.

C2. The private channels used in LINCOS provide information-theoretic confidentiality and computational authenticity at the time of data transmission.

C3. The proactive secret sharing schemes used in LINCOS provide information-theoretic confidentiality.

C4. During their usage periods, the secret sharing services used in LINCOS prevent mobile adversaries from learning k or more shares.

Theorem B.1. *Under assumptions C1, C2, C3, and C4 the system* LINCOS *provides information-theoretic confidentiality protection.*

For the proof of Theorem B.1 we refer to the full paper.

C. EXPERIMENTAL EVALUATION

In the following, we present a performance analysis of LINCOS. We estimate the storage space required by the system and investigate data transmission limits imposed by QKD. We also measure the time required for integrity verification. To do so, we run the following experiment. A document is stored and protected using LINCOS over a period of 100 years, starting in 2016 and ending in 2116. Share and timestamp renewal happen every two years. The share renewal period is to be chosen such that mobile adversaries are unable to recover more shares than permissible. Also, the typical storage hardware maintenance service interval is two years. The timestamp renewal period is chosen in accordance with typical certificate renewal periods. Such certificates are required to verify the timestamps. Finally, commitment renewal happens every ten years. This is in accordance with the heuristic security assumptions for the commitment scheme parameters. Parameter choices for the complexity-based cryptographic components are done according to the heuristics in [18]. The corresponding expected protection periods are presented in Table 1.

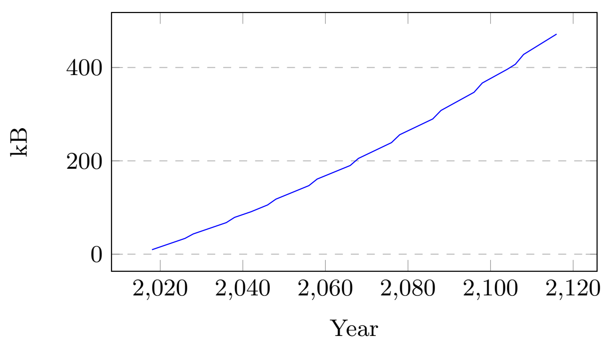

Figure 4: Size of evidence record.

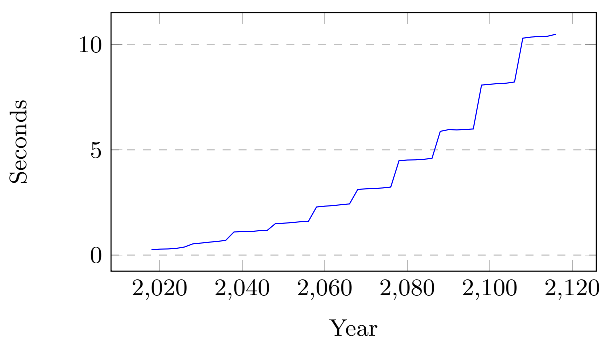

Figure 5: Performance of evidence verification.

C.1 Storage space

We analyze the storage space required by the shareholders and the evidence service as a function of the bitlength $\mathsf{size_d}$ of the protected document d.

Shareholders. Each shareholder stores one share s per document. Its size is $\mathsf{size_s} = \mathsf{size_d} + \mathsf{size_R}$. Here R is the list of decommitment values accumulated over time. Its size is independent of the document size. The size of a single decommitment value equals the size of the parameter q of the commitment scheme. At present, a secure instantiation of the Pedersen commitment scheme requires a decommitment value size of 224 bit. The experiments show that this data accumulates to $\mathsf{size_R} \leq 1$ kB over 100 years.

Evidence service. The evidence service stores one evidence record E per document. The size of the evidence record $\mathsf{size_E}$ is independent of the document size. It depends on the size and number of timestamps and commitments contained in the evidence record. It grows over time because a new timestamp and a new commitment are added with each renewal. The growth of $\mathsf{size_E}$ over time is shown in Figure 4. Our experiments show that the size of the evidence record accumulates over 100 years to $\mathsf{size_E} \approx 500$ kB.

C.2 Data transmission

Our system uses authenticated and private channels. Authenticated channels easily allow for a data rate of 1 Gb/s, while they are used for sending only a few hundred kB of evidence data. So the cost for data transmission via authenticated channels is negligible. Private channels are realized using OTP and QKD. The transmission rate of these channels is limited due to the key generation rate of QKD. Therefore, in our analysis we focus on the QKD transmission rate.

Data rate of private channels. The QKD performance in the Tokyo QKD Network differs from link to link because fiber channel lengths as well as specifications of QKD devices are different from each other. Furthermore, some nodes are directly connected by a QKD link, others have to use key relay. The achieved secret key rates of the QKD links range from 10 kb/s to 300 kb/s depending on the specification of the respective QKD link. To prevent being limited by the slowest QKD links (10 kb/s), keys are relayed between appropriate KMAs. Such key relaying balances the key material across the network. The resulting throughput in our current configuration is $\mathsf{keyRate_{QKD}} = 40$ kb/s.

Storage and retrieval. When the document owner stores data in the confidentiality system, he sends one share to each shareholder. Likewise, when retrieving the data, the document owner receives one share per shareholder. Since $\mathsf{size_s} = \mathsf{size_d} + \mathsf{size_R}$, the time required for generating the necessary OTP key material per share transfer in a private channel is $t_s = \mathsf{size_s}/\mathsf{keyRate_{QKD}}$ seconds. For example, 1 GB of data can be shared in 2.3 days at $\mathsf{keyRate_{QKD}} = 40$ kb/s.

Share renewal. For share renewal, the document owner retrieves the current set of shares and distributes new shares to the shareholders. So the time for communicating the key material required for resharing is $2 * t_s$. For example, 1 GB of data can be reshared in 4.6 days at $\mathsf{keyRate_{QKD}} = 40$ kb/s.

C.3 Evidence verification

Figure 5 shows timings for verification of an integrity proof. The timings were measured on a computer with an 2.9 GHz Intel Core i5 CPU and 8 GB RAM running our Java implementation of the verification algorithm. As the evidence record and the list of decommitment values grow over time, the verification time increases. Verification of evidence accumulated over 100 years takes approximately 10 seconds. It can be expected that, because computers are getting faster, in a hundred years from now integrity proof verification will only take a fraction of this time.