Privacy-Preserving Data Sharing for Medical Research

Michael J. Fischer[1](✉), Jonathan E. Hochman[2], and Daniel Boffa[3]

[1] Computer Science, Yale University, New Haven, USA
michael.fischer@yale.edu
[2] UNS Project and Hochman Consultants, Hartford, USA
jonathan@unsproject.com
[3] School of Medicine, Yale University, New Haven, USA
daniel.boffa@yale.edu

Abstract. Electronic patient medical records contain vast amounts of information of potential value to researchers striving to increase understanding of diseases, treatments, and outcomes. Effective use of such data is limited by privacy and technical concerns. Privacy laws require the removal of Personally Identifiable Information (PII) from the released data. Technical concerns are that the data must be abstracted for consistency across different providers. To be most useful, data from different providers for the same patient must be linked together. This paper applies cryptographic techniques to the problem of privacy-preserving linking of medical records.

Keywords: Medical research · Privacy · Cryptographic data linking

1 Introduction

The goal of this work is to design a national system for medical data sharing that meets several criteria:

1. Understandable to stakeholders
2. Supportable by stakeholders
3. Simple and explainable
4. Actionable with minimal startup investment
5. Sustainable
6. Secure

Our contribution includes a new cryptographic primitive, called a *blinding-completion pair*, which addresses the practical problem of linking anonymous medical records. Blinding-completion pairs provide a method for generating a multitude of anonymous pseudonyms for an entity, to be used by data sources, and then consolidating that multitude into a single anonymous pseudonym at the destination database. We also describe a distributed system and protocol for

C. Johnen et al. (Eds.): SSS 2021, LNCS 13046, pp. 78–89, 2021.
https://doi.org/10.1007/978-3-030-91081-5_6

sharing medical data that can preserve privacy when confronted with occasional data breaches.

This paper is organized as follows: Sect. 2 summarizes related work. Section 3 describes the stakeholders and sketches the existing flow of clinical data from patient to researcher. Section 4 focuses on the problem of privacy-preserving linking of patient data. Section 5 suggests possible modifications of the workflow to shift responsibility for maintaining privacy to specialized "security nodes". Section 6 provides a brief threat model and discloses known limitations of the system. Section 7 summarizes our contribution and directions for future research.

2 Related Work

The problem of connecting records from multiple databases is called *record linking*. What we aim to achieve is called *privacy-preserving record linking (PPRL)*, where the goal is to link patient records while protecting the PII of the patient. The interest in PPRL of health records goes back at least 16 years to the paper of Demuynck and De Decker [8], who propose a complicated multi-stakeholder protocol that uses cryptographic techniques to achieve PPRL. The idea of using pseudonym identifiers to substitute for PII linkage appears in Alhaqbani and Fidge in 2008 [1].

Many other linkage techniques have subsequently been studied. Vatsalan, Christen, and Verykios [20] give a taxonomy of PPRL techniques in a 2013 paper containing 143 references to the extensive literature on the subject! They observe that preserving the privacy of shared data such as medical records is a difficult problem and that existing approaches have a variety of drawbacks. For example, some systems have focused on joining records from just two sources, which would not satisfy our design requirements.

Recent work aims to address one or more of the shortcomings of prior work. Camenisch and Lehmann [6] add user-auditability to pseudonym systems. The PRIMAT system [9] handles multiple sources of medical data, but it assumes the existence of a "trusted linkage unit (LU) [that] performs the actual linkage of encoded records submitted".

Our approach is different because we leverage existing universal identifiers and anonymize them by generating two levels of pseudonyms. This results in a system with resilience to limited breaches that is easy to understand by stakeholders. Moreover, our system does not put sensitive medical data in the hands of any third party. Medical data flows directly from a health care provider to a research database.

Our proposed system also differs from many others in the location of the three types of data: *identifier* is a segment of coded information that is unique for each person, *identifying information* enables a specific person to be identified, and *sensitive information* is desired to be kept private and not shared publicly as being attributable to an individual.

An identifier by itself is meaningless and is just a code. For example, any random combination of nine numbers very well may be a social security number, but without identifying information, there is no relevance, utility or vulnerability. Identifying information alone is not overly relevant, because it simply notes the existence of a person, without any detail of that person. For example, names (and addresses and phone numbers), have historically been distributed in phone books. Finally, sensitive information that cannot be linked to a specific person poses no risk to privacy and is the principal that allows large databases to exist for medical research. Any medical textbook could contain what most would consider sensitive information, such as data related to the treatment of specific patients. When such information is not attributed to any person, it poses no vulnerability. An important caveat is that if the anonymized sensitive information is sufficiently detailed, it may serve as a fingerprint that can be correlated with publicly available data and used to identify the subject [15].

3 Medical Information Workflow

Data enters the health care system when a patient contacts a provider, whether a primary care physician or a hospital. At that point, the provider determines and records the patients PII and begins or updates the patient's chart.

To curate data for statistical and research purposes, trained registrars extract select data elements from the medical record according to specific data field definitions, resulting in highly structured data sets. The datasets are then stripped of PII and exported in a deidentified manner to one or more of several national databases. Importantly, not every health care entity submits to every database, and each database only requests certain fragments of the patient's medical information [5]. As a result, each patient's care is captured by the databases in a piecemeal fashion. Because the databases do not collect PII, there is no way to consistently reunite the fragments of the health care data back together to create a complete picture of a patient's journey through the diagnosis, treatment, and outcome of their medical condition.

The analysis of patient outcomes captured within the databases has led to dramatic improvements in the safety and effectiveness of care for almost every medical condition [18]. However, the ability of the database research to characterize relationships between variables and outcomes in medical care is critically dependent on the breadth of information available for analysis (e.g., to control for bias and confounding effects). Because databases are only capturing fragments of the medical journey, there are limitations to the types of improvements that currently can be made with database research [16].

For example, the database that best captures cancer stage does not capture the specific type of chemotherapy that patients received [5]. If there were a way to reunite all the fragments of data back together, medical research using existing databases would become far more powerful, and many more improvements would be possible. (See Daniel Boffa, *Comparing Comparisons*, in comments to [16]).

A simple but unacceptable "solution" to the linking problem is to give each patient a universal health identifier to be included with the patient's record in each database. This is used in some other countries (e.g., Norway [2]). However, in the United States, the topic of a national identifier has become highly polarizing, making this a less feasible option. Moreover, anyone with access to the curated databases would be able to join the several databases into one master health record for the patient with that identifier. This is often sufficient, when combined with other information readily available on the internet, to deanonymize the health record and reveal the patient's PII.

4 Privacy-Preserving Linking of Patient Data

We describe a new cryptographic primitive for generating identifiers that allows a database to link records from different data providers while preserving privacy in the face of many kinds of breaches.

4.1 Blinding-Completion Pairs

Let x be the identifier used by a data provider h to identify an entity. Let $b(x)$ be a one-way hash function. A *blinding-completion pair* for h is a pair of one-way functions $(b_h(x), c_h(y))$ such that $b(x) = c_h(b_h(x))$ for all x, and $b(x)$ is also one-way. Like a cryptosystem $(E_h(x), D_h(y))$, the composition of the second function in the pair with the first yields the same function for all keys h. In our case, the composition is the fixed blinding function $b(x)$, which defines the alias $y = b(x)$ for x. While neither x nor y can be recovered from $y_h = b_h(x)$, y *can* be recovered from *any* single value y_h if the corresponding completion function c_h is available, since $y = c_h(y_h)$.

4.2 Implementation

There are several ways to implement blinding-completion pairs. One way is to use cryptographic accumulators. Let $Q = \{b_1, \ldots, b_N\}$ be a set of *quasi-commutative cryptographic hash functions* [3]. They have the property that the N-way composition of these functions in any order yields the same function B. Hence, for any subset $S \subseteq Q$, the composition of those functions in S, call it b_S, can be used as the first element of a blinding-completion pair, and the composition of $b_{(Q-S)}$ becomes the completion function c_S. The drawback of this scheme is that N must be known in advance, and the time complexity of finding b_S and c_S grows with N.

We use a different scheme based on discrete logarithms. First we introduce some standard number theory. For positive integer n, let $\mathbf{Z}_n^*$ be the set of positive integers less than n that are relatively prime to n. The size of $\mathbf{Z}_n^*$ is given by Euler's totient function $\phi(n)$.

In the special case that n is a prime p, $\mathbf{Z}_p^* = \{1, \ldots, p-1\}$, so $\phi(p) = p - 1$. Also, p has primitive roots. We say g is a *primitive root* of p if every number $a \in \mathbf{Z}_p^*$ can be expressed as $a = g^k \bmod p$ for some $k \in \mathbf{Z}_p^*$. The number k is called the *discrete logarithm of* a *modulo* p. Computing the discrete logarithm is believed to be computationally difficult when p and g are chosen carefully.

For our purposes, we choose $p = 2q + 1$, where q is a *Sophie Germain prime* and p is called a *safe prime*. Such prime pairs are widely used in cryptography, so a suitable supply exists for our purposes. An estimate of the number of Sophie Germain primes less than n is $\Theta(n/(\log n)^2)$ [19, pp.123–124].

There are $\phi(\phi(p)) = \phi(p-1)$ primitive roots in $\mathbf{Z}_p^*$. This makes it possible to find a primitive root g by a guess-and-check method. Guess a number $g \in \mathbf{Z}_p^*$ and check that $g^q \equiv -1 \pmod{p}$. The expected number of guesses required to find g is $(p-1)/\phi(p-1) = \mathcal{O}(\log\log p)$ [14, p. 391]. How big is $\phi(p-1)$? Because we've chosen $p-1 = 2q$, then $\phi(p-1) = \phi(2)\phi(q) = q-1 = (p-3)/2$.

Let r, u be positive integers in $\mathbf{Z}_{\phi(p)}^* = \mathbf{Z}_{p-1}^*$, and let v be a positive integer less than $\phi(p)$ such that $r = (u+v) \bmod \phi(p)$. Define $b_u(x) = xg^u \bmod p$ and let $c_v(y) = yg^v \bmod p$. Then (b_u, c_v) is a blinding-completion pair for the blinding function $b(x) = xg^r \bmod p$. This follows since

$$c_v(b_u(x)) \equiv c_v(xg^u) \equiv (xg^u)g^v \equiv xg^{u+v} \equiv xg^r \pmod{p}.$$

The last identity follows from Euler's Theorem, which states that for $a \in \mathbf{Z}_p^*$, $a^{\phi(p)} \equiv 1 \pmod{p}$.

The parameters p, q matter both for convenience and security. To choose r from $\mathbf{Z}_{\phi(p)}^*$, we need to find an r that is relatively prime to $(p-1)$. But an arbitrary $p-1$ might have many small factors, e.g., $p = 71$. However, since we choose p, q so that $p-1 = 2q$, we know the only factors of $p-1$ are 2 and q. Choosing a safe prime p makes it easy to find random numbers in $\mathbf{Z}_{\phi(p)}^*$. As for security, the discrete logarithm problem is hard in general, but a solution may be feasible via the *Pohlig–Hellman algorithm* when $p-1$ has no large prime factors [17]. A safe prime does not have this weakness.

As explained in Sect. 5 below, we will allow security nodes to independently choose random values of $r, u \in \mathbf{Z}_{\phi(p)}^*$ and calculate $v = (r-u) \bmod \phi(p)$.

Because r and u may be chosen independently by different security nodes, there is a theoretical risk of $v = 0$, which would produce the undesirable result $y_h = y$. From the point of view of a cryptographer, such a result is not a problem, but to satisfy our design requirements we want to provide an unqualified guarantee that the identifier used by a data provider h does not appear in a database that links its records.

The value r is secret and may not be shared, therefore the security node choosing u cannot "peek" at r to make sure it chooses a safe value for u. In practice, when p is very large, the probability of $r = u$ is vanishingly small. Should this ever happen, a simple remedy is to choose a new value for r via key rotation, as explained next.

4.3 Key Rotation

The values of r, u, v should be rotated periodically in case they are ever compromised. We provide a sketch of how such rotation could be implemented.

To rotate the value of r, choose a random $0 < s < \phi(p)$ and calculate $r' = (r+s) \bmod \phi(p)$. Check that $r' \in \mathbf{Z}_{\phi(p)}^*$, and if not, choose a different random

s and try again. Then recalculate $v' = (r' - u) \bmod \phi(p)$. Finally, to update the blinded value $y = xg^r \bmod p$, calculate a new blinded value $y' = yg^s \bmod p$. Note that x is not needed for this calculation. Then y' is the new blinded value for the same x, since $y' \equiv xg^r g^s \equiv xg^{r'} \pmod{p}$.

To rotate the value of u, choose a random $0 < s < \phi(p)$ and calculate $u' = (u + s) \bmod \phi(p)$. Check that $u' \in \mathbf{Z}^*_{\phi(p)}$, and if not, choose a different random s and try again. To update the blinded value $y_h = xg^u \bmod p$, calculate a new blinded value $y'_h = y_h g^s \bmod p$. Then y'_h is the new blinded value for the same x, since $y'_h \equiv xg^u g^s \equiv xg^{u'} \pmod{p}$.

5 Proposed Workflow for Enhanced Security

We propose three additions to the existing workflow to maintain security while still permitting research data sharing.

1. We envision a system of restricted *local patient identifiers* (LPIDs) that can be used to identify the medical records of a given patient within the context of a single health care provider. Local identifiers are obtained from a patient's PII via a one-way cryptographic function. This prevents the local identifier from being reverse engineered to obtain PII.
2. Using the cryptographic technique of *blinding-completion functions*, the local patient identifiers for different health care providers can be used to calculate an anonymized patient identifier (APID). The APID allows a medical database to link patient records across providers while still providing no clear path to finding the corresponding PII.
3. To further protect patient anonymity and privacy, we propose to separate the security services from the servers and databases holding the actual PII (in the case of hospitals) and medical data (in the case of curated database).

5.1 Trust

Our model of trust has two dimensions: whether the party has *good intentions* to keep sensitive information private, and whether the party is *competent* to do so. For example, while we may trust health care systems to do their best to keep sensitive patient information private, they are not always good at cybersecurity, as evidenced by the large number of cyber-attacks against health care organizations.[1] Even when a health care system has a central information technology department capable of maintaining network security, that competency may be a scarce resource.

Our model of trust is different from traditional adversarial models that consider the worst possible outcomes from an untrusted party. Our model is informed by one author's experience in analyzing dozens of actual lawsuits related to online

[1] In a recent survey of health care organizations, 70% of respondents reported that their organizations had experienced significant security incidents in the prior 12 months [12].

identity and privacy. While malice is sometimes present, incompetence is much more likely. In the modern era this idea has been called Hanlon's razor, which states, "Never attribute to malice that which is adequately explained by stupidity" [21]. Earlier such attributions go back at least to Goethe [10]: "I have realized once again that misunderstandings and lethargy can cause more going wrong in the world than cunning and wickedness do. At least, those two are certainly less common".

While health care providers are expected to be competent at medical treatment, there is no reason to expect them to be competent at cryptography (nor would we trust the average cryptographer to perform surgery). To mitigate the risk of health care providers performing cryptographic functions insecurely or leaking secret keys, we restrict access to certain functions and the secret keys that power them. To keep everyone safe, we introduce additional parties to the transaction, called "security nodes", which have demonstrated technical competence. Each health care provider will choose a security node to work with, and so will each medical database.

A security node is a network service that can be trusted to implement cryptographic functions correctly and to hold secret keys without leaking them. Security nodes could be independently operated, or they might be operated by a department within a medical organization with the required competence. Importantly, a security node isolates the secret keys used by cryptographic functions or for signing messages in a single location. This makes it easier to protect secret keys by storing them in specialized computing hardware, such as a hardware security module (HSM). Security nodes also provide authentication services to health care providers. Each security node has a public-private key pair it can use to sign and authenticate messages for other security nodes. A special "executive" security node keeps a list of all security nodes and their public keys. This list may be periodically updated and distributed, enabling security nodes to reliably authenticate each other's messages.

5.2 Parties

A transaction at minimum includes six parties:

1. A patient w who is identified with a *user identifier* (UID),
2. A health care provider h who treats patients and gathers medical data,
3. The health care provider's security node that provides LPIDs that can be attached to medical data in lieu of UIDs,
4. The database's security node that attaches an APID to medical data in lieu of LPIDs,
5. A database d that collects anonymous patient profiles, identified only by APID,
6. Researchers that receive anonymous patient profiles.

5.3 Identifiers

There are three levels of identifiers, each with distinct properties:

1. UID is an invariant identifier, such as a name-birthday pair or a Social Security number. The UID is readily available to the patient and widely used by health care providers. A patient's UID must never be shared because it constitutes PII.
2. LPID identifies patients relative to a health care provider and has no apparent connection to any PII. A patient's LPID is different at every provider and is used for sending anonymized records to a database.
3. APID identifies patients relative to a database and has no apparent connection to any PII or to any LPID. Anonymized records sent to a database by different health care providers for the same patient are associated with the same APID, which enables record linking.

The LPID and APID identifiers are rotated periodically to frustrate any attacker who manages to breach the system. Rotation can be done if a breach is detected, or on a regular schedule to limit the damage from an undetected breach, and to provide other benefits. Key rotation is a widely accepted good practice in cloud computing [11].

5.4 Initialization

Initially, one or more databases join our proposed system, which provides the values p, q, and g. Each database d chooses a security node which generates a random value r_d such that $r_d \in \mathbf{Z}^*_{\phi(p)}$. The value r_d is used to generate blinding-completion function pairs and must be kept secret.

Each health care provider h joining the system chooses a security node. The health care provider obtains a public-private key pair for signing messages (e.g., an X.509 security certificate [4]), using a digital signature algorithm such as DSA or ECDSA, and verifying the health care provider's identity to its security node. The public key is registered, or "pinned", to the security node. Upon registration, the health care provider's security node will chose a random value u_h such that $u_h \in \mathbf{Z}^*_{\phi(p)}$. The value u_h must be kept secret and is used to generate a blinding function $b_h()$.

To join a database d, a provider h causes its security node to send u_h to the security node of database d. The security node of d then calculates a value $v_d = (r_d - u_h) \bmod \phi(p)$. The value v_d must be kept secret and is used to generate a completion function $c_h()$.

These blinding-completion functions are constructed in such a way that:

1. Each blinding function for each provider h produces a different pseudorandom identifier LPID$_h$ for each patient.
2. Each completion function for each provider h to each database d maps each LPID$_h$ to APID$_d$.

If a health care provider h participates in multiple databases, it uses the same LPID$_h$ identifiers, but each database d will generate different APID$_d$ identifiers. Conversely, when multiple providers contribute medical data to a database, each provider h has different LPID_h identifiers and the database d has the same APID_d identifiers. To preserve privacy, no provider knows any of the APID_d values, and no database knows any of the LPID_h values. The patient identifier equivalence pairs (LPID_h, APID_d) are only known to, or computed by, security nodes.

5.5 Contribution of Patient Profiles

Medical providers may contribute patient profiles to a database. A profile contains demographic and medical information of interest to researchers. For example, a patient profile might include age, medical diagnosis codes and dates, occupation, ethnicity, treatment history, and other target characteristics. Existing standards for storing digital medical records can be used.

To contribute a profile to a database d, a health care provider h performs several steps.

1. The provider hashes the patient's UID, w, with a standard, widely available hash function such as SHA256 [7], to generate a value x that it sends to its security node. The security node then applies the blinding function for that health care provider to x, resulting in the value LPID_h. The security node returns LPID_h to the health care provider, and h adds it to the patient's medical record.
2. The provider generates a random transaction number t. The relevant profile data m is then composed into a message (h, t, m, d) and sent to the database d. The medical data will only be added to the database after it is authenticated by the database's security node.
3. The provider creates a token (such as a JSON web token [13]) containing the quadruple (LPID_h, h, t, d) and signs it using its secret key. The provider sends the signed token to the provider's security node, which then authenticates the signature using the health care provider's public key and appends its own signature to the token.

The provider's security node sends token (LPID_h, h, t, d) to the security node of database d which does the following steps:

1. Authenticates the signature of the health care provider's security node.
2. Verifies that the health care provider's name h in the token matches the name in the message.
3. Applies the appropriate completion function $c_h()$ to LPID_h to generate APID_d.
4. Creates a new token (APID_d, p, t, d), signs, and sends it to database d.

Database d receives the token (APID_d, h, t, d) and then performs these steps:

1. Authenticates the signature of its own security node.
2. Finds the message (h, t, m, d) with the same transaction number t.
3. Verifies that the health care provider's name h in the message matches the health care provider's name in the token.

4. Adds the medical data m, the provider h, and APID_d to its data store. If there is an existing record with APID_d, the new data is linked to the existing record.

5.6 Accessing Medical Data for Research

A researcher can connect to a database and search for patient profiles that match desired criteria for the study. Upon approval by an appropriate medical research ethics board, the researcher can then requisition specific medical data from the database that is relevant to the research being conducted.

Upon receiving an approved request for medical data related to a patient profile, the database then retrieves from its database the patient medical data that meets the researcher's specific criteria.

When releasing data to a researcher, the identifier for each record, APID_d, should be removed or hashed with a one way function such as SHA256. Researchers are not security experts. Therefore, they should not be trusted to keep the APID_d identifiers private.

6 Threat Analysis

The system we describe, like all such systems, does not confer perfect security. If a security node were compromised, an attacker might learn the secret values r_d, u_h, or v_h. These secret values could enable an attacker to recover some or all of the blinding-completion functions $b()$, $b_h()$, or $c_h()$ and their inverses. Having one or more of these inverse functions could give an attacker who possesses APID_d the ability to calculate LPID_h or x, the hash of the patient's UID. While it is not practical to invert the hash function used to generate x, an attacker could test whether a known UID value, when hashed, equals x.

Assume an attacker gathers medical data from researchers. This data would contain hashes of the APID_d for each record. If the attacker additionally compromises health care providers and security nodes, it is conceivable that they could eventually link a UID to anonymous research data. Given UID and r_d, an attacker can calculate APID_d, hash this value and then compare it to the data collected from researchers. The difficulty of such an attack is high because it requires compromising at least one health care provider and at least one security node of a database containing data from that health care provider within a limited time frame (the key rotation period). Moreover, if such an attack were to succeed, it would likely deanonymize only a limited number of medical records, especially if there are many independent health care providers, databases, and security nodes.

The security nodes described in this system only need to communicate with other security nodes and with the medical providers or databases they serve. Consequently, a firewall can protect each security node so that it only communicates with systems on an "allow" list. This type of protection increases the difficulty of breaching a security node because even if the system has vulnerabilities, an attacker needs to gain access to a system on a security node's "allow" list even to commence a remote attack on the security node.

We believe that the difficulty of attacking our proposed system is sufficiently high, and the profitability sufficiently low, that attackers would prefer to attack health care providers directly and aggregate data via UID. Therefore, our proposed system does not materially increase the risk of private medical data being exposed in a data breach versus the status quo.

7 Conclusion

We have presented a new cryptographic technique called blinding-completion pairs and demonstrated how they could be used to enable the sharing of private data without revealing personally identifiable information (PII).

Based upon blinding-completion pairs maintained by security nodes, we have drawn a sketch of how health care providers could supply medical data to one or more databases that would aggregate data for each patient and then make those consolidated records available as anonymous data to researchers. Our system could release data for medical research in a way that protects patient PII while still enabling qualified researchers to identify records from different health care providers that belong to the same patient.

Possible areas for future work include constructing a prototype system, developing new blinding-completion functions with improved security properties, and investigating alternative sharing protocols that may offer stronger privacy guarantees in the event of data breaches.

Acknowledgments. We are grateful to Ewa Syta of Trinity College (Connecticut) for a thorough reading of an early draft of this paper and for providing many helpful comments and pointers to the relevant literature. We thank Bonnie Kaplan of the Yale School of Medicine for sharing her vast knowledge of the world of electronic health data with us. Lastly, we are indebted to Alice Fischer from the University of New Haven, who scrutinized our nearly final draft.

References

1. Alhaqbani, B., Fidge, C.: Privacy-preserving electronic health record linkage using pseudonym identifiers. In: 10th International Conference on e-health Networking, Applications and Services, HealthCom 2008, pp. 108–117 (2008). https://doi.org/10.1109/HEALTH.2008.4600120
2. Bakken, I.J., Ariansen, A.M.S., Knudsen, G.P., Johansen, K.I., Vollset, S.E.: The Norwegian Patient Registry and the Norwegian Registry for Primary Health Care: Research potential of two nationwide health-care registries. Scand. J. Public Health **48**(1), 49–55 (2020)
3. Benaloh, J., de Mare, M.: One-way accumulators: a decentralized alternative to digital signatures. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 274–285. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48285-7_24

4. Boeyen, S., Santesson, S., Polk, T., Housley, R., Farrell, S., Cooper, D.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, May 2008. https://doi.org/10.17487/RFC5280
5. Boffa, D.J., et al.: Using the national cancer database for outcomes research: a review. JAMA Oncol. **3**(12), 1722–1728 (2017)
6. Camenisch, J., Lehmann, A.: Privacy-preserving user-auditable pseudonym systems. In: 2017 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 269–284 (2017). https://doi.org/10.1109/EuroSP.2017.36
7. Dang, Q.: Secure hash standard, 04 August 2015. https://doi.org/10.6028/NIST.FIPS.180-4
8. Demuynck, L., De Decker, B.: Privacy-preserving electronic health records. In: Dittmann, J., Katzenbeisser, S., Uhl, A. (eds.) CMS 2005. LNCS, vol. 3677, pp. 150–159. Springer, Heidelberg (2005). https://doi.org/10.1007/11552055_15
9. Franke, M., Sehili, Z., Rahm, E.: PRIMAT: a toolbox for fast privacy-preserving matching. Proc. VLDB Endow. **12**(12), 1826–1829 (2019). https://doi.org/10.14778/3352063.3352076
10. von Goethe, J.W.: The Sorrows of Young Werther. Oxford World's Classics, Oxford, (tr.) David Constantine, online edn., December 2020. https://doi.org/10.1093/owc/9780199583027.001.0001, Accessed 25 Sept 2021
11. Google Cloud Key Management Service: Key rotation. https://cloud.google.com/kms/docs/key-rotation. Accessed 26 Sept 2021
12. HIMSS Cybersecurity Survey (2020). https://www.himss.org/sites/hde/files/media/file/2020/11/16/2020_himss_cybersecurity_survey_final.pdf
13. Jones, M., Bradley, J., Sakimura, N.: JSON Web Token (JWT). RFC 7519, May 2015. https://doi.org/10.17487/RFC7519
14. Knuth, D.E.: The Art of Computer Programming, vol. 2: Seminumerical Algorithms, 3rd edn. Addison-Wesley Professional (1998)
15. Kolata, G.: Your Data were 'Anonymized'? These Scientists Can Still Identify You. The New York Times, 23 July 2019. https://www.nytimes.com/2019/07/23/health/data-privacy-protection.html
16. Kumar, A., et al.: Evaluation of the use of cancer registry data for comparative effectiveness research. JAMA Netw. Open **3**(7), e2011985 (2020). https://doi.org/10.1001/jamanetworkopen.2020.11985
17. Pohlig, S., Hellman, M.: An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (Corresp.). IEEE Trans. Inf. Theory **24**(1), 106–110 (1978). https://doi.org/10.1109/TIT.1978.1055817
18. Salazar, M.C., et al.: Association of delayed adjuvant chemotherapy with survival after lung cancer surgery. JAMA Oncol. **3**(5), 610–619 (2017). https://doi.org/10.1001/jamaoncol.2016.5829
19. Shoup, V.: A Computational Introduction to Number Theory and Algebra, chap. 5.5.5 Sophie Germain Primes, 2nd edn., pp. 123–124. Cambridge University Press, February 2009. ISBN 9780521516440
20. Vatsalan, D., Christen, P., Verykios, V.S.: A taxonomy of privacy-preserving record linkage techniques. Inf. Syst. **38**(6), 946–969 (2013). https://doi.org/10.1016/j.is.2012.11.005
21. Wikipedia contributors: Hanlon's razor – Wikipedia, The Free Encyclopedia (2021). https://en.wikipedia.org/w/index.php?title=Hanlon's_razor&oldid=1045571584. Accessed 24 Sept 2021