Secured Electronic Health Record Management Protocol

Salekul Islam
Department of Computer Science and Engineering
United International University, Dhaka, Bangladesh
salekul@cse.uiu.ac.bd

Syeda Farzana
Department of Computer Science and Engineering
United International University, Dhaka, Bangladesh
sfarzana142014@mscse.uiu.ac.bd

ABSTRACT
Since digitization is now a common practice for storing and retrieving data, Electronic Health Record (EHR) management is becoming very popular. EHR management will bring many benefits including easy to store, cost effective, shareable with health professionals to a remote location, etc. However, EHR stores very sensitive data and thus, a number of security properties including privacy, secrecy, integrity, authenticity of data and availability must be ensured during data transmission, storing and sharing with health professionals. In this paper, we have studied symmetric key based technique in designing EHR management protocol. In light of the existing efforts, we have developed a simple protocol for secured EHR management and validated that protocol using AVISPA, an industry-strength security protocol validation tool. AVISPA has reported our protocol free from known attacks and confirmed the desired security properties as well.

CCS Concepts
• Security and privacy→Management and querying of encrypted data

Keywords
Electronic Health Record (EHR); symmetric key based encryption; AVISPA; secured EHR; patient-controlled

1. INTRODUCTION
Electronic Health Record (EHR) is any health related information or data of patient stored digitally using electronic methods. This information can only be accessed instantly by authorized users, e.g., doctors or the patients themselves, maintaining top notch security to assure privacy. EHR contains sensitive information including patients' medical and treatment histories, administrative and billing data, prescriptions provided by the doctors, medical test reports, personal information (e.g., age, weight and height), etc. As EHR contains valuable information security and privacy is the prime concern. The EHR system can also be built to go beyond standard clinical data collected in a medical institution and can add various other facilities to allow a broader view of a patient's care [3]. For example it can allow access to tools to help doctors to make decisions about a patient's care. As EHRs are stored digitally they can be easily shared from one place to another through intra connected private networks, e.g., from one branch of a hospital to another distant branch [9]. Since they are stored digitally they are able to reduce the incidence of medical error as a result of improving the accuracy and clarity of medical records. Moreover the availability of all up to date health related information of a patient in one place prevents duplication of tests, helping to speed up the process of diagnosis or treatment. Even if the patient is unconscious the emergency department can sought out the EHR of the patient to take necessary steps and helps the patient to recover from life threatening condition. On the other hand a patient can get motivated to organize his lifestyle accordingly when visually seeing the trend of the lab results over the medication period by logging on to his own record.

ICISPC 2017, July 26–28, 2017, Penang, Malaysia

ACM ISBN 978-1-4503-5289-5/17/07…$15.00
DOI: https://doi.org/10.1145/3132300.3132311

EHR involves the transfer of health information through electronic media, including the Internet [18]. Therefore the need for privacy, confidentiality and security [13, 17] is inevitable as essential components between healthcare consumers and providers. Such properties are essential not only for patient health information but also for clinical care, research, payment, and healthcare policymaking. Thus various Cryptographic protocols that use cryptographic techniques have been developed to achieve secured management of EHR. Two popular methods are found in the literature to secure EHR: symmetric key encryption [5, 7] and Attribute-Based Encryption (ABE). Since no public keys are involved, symmetric key based systems are less expensive to implement and maintain. The concept of ABE was first introduced in [8, 16] to store and share encrypted data without using symmetric key. In ABE, if A encrypts data using K_A, B can decrypt this data using K_B, as long as the identities of A and B are close to each other. Here, identities are considered as a set of descriptive attributes, and thus it was termed as Attribute-Based Encryption (ABE). Recently, different efforts have been carried out to use ABE in securing EHR management system [10, 14].

In this paper, we have developed a simple symmetric key based EHR management protocol. Although our proposal is primarily based on symmetric key, it identifies the doctors using their attributes [10, 14]. The security properties of this protocol have been validated by modelling the protocol using Automated Validation of Internet Security Protocols and Applications (AVISPA) [1]. AVISPA is a push-button tool with industrial-strength technology for the analysis of different Internet security protocols and applications. It is being used by the developers of different security protocols and by academic researchers as well [11, 12]. In our validation, AVISPA reports the proposed protocol free from attacks. The rest of the paper is organized as follows: section 2 explains the security properties and an architecture for secured EHR management, section 3 briefly explains AVISPA, section 4 illustrates how we develop the AVISPA model and the validation results, and finally section 5 concludes the paper.

2. SECURED EHR MANAGEMENT

2.1 Required Security Properties

We summarize the required security properties that our proposed secured EHR management should support. **Privacy and Secrecy**: Patient privacy and secrecy is crucial while records on remote servers leave the data open to security exploits and data theft [15]. Encryption is the mechanism, in general, applied to ensure secrecy [10]. Encryption converts the original message or information into encoded text and is only decrypt able or is readable by authorized persons only. Moreover account security is defined by secure passwords and strict user access levels. Secure password, which allows access to the EHR account, is indispensable to ensure that information does not fall into the wrong hands.

Integrity: Integrity, internal consistency, and accuracy of information in the patient's EHR are imperative to ensure accuracy of the complete health record [2]. It involves information such as patient identification, authorship validation, amendments and record corrections. As wide variety of data is collected in healthcare it must be collected accurately, completely, and consistently and ensure documentation integrity to avoid wrong information documented on the wrong patient health record. It is important to guarantee that appropriate care and billing activity is subjected to the correct patient.

Access Control: User authentication is required to determine whether someone is, in fact, who it is declared to be. The purpose of authentication is to allow authorship and assign responsibility for an act, event, opinion, or diagnosis made by the doctor [2]. Entries in the healthcare record should be authenticated by the patient [13]. Patient can allow or deny sharing their information with other healthcare practitioners [10]. To implement patient consent in a healthcare system, patient may grant rights to users on the basis of a role or attributes held by the respective user [17]. The access and sharing of EHRs could be provided by end-to-end source verification through signatures and certification.

Availability: It is essential for any EHR system to be available and make the information accessible when it is needed [2]. This means that the computing systems used to store and process the EHR data, the security controls used to protect it, and the communication channels used to access it must be operating properly all the time and are reachable.

2.2 EHR Management Architecture

EHR management architecture basically means the model that the system should follow to make the storage and transfer of EHR feasible. Many different approaches have already been carried out and more are being produced to make the system advanced having more security measures and enhanced upgrades in par with the always improving latest technologies. A general and very basic EHR management architecture is shown in Figure 1.

The structure basically depicts that the EHRs are stored centrally by the authorization of a patient. The patient updates or stores the medical records after receiving treatment from doctor-1. This same patient might be referred to doctor-2 for further medical support. Doctor-2 can view the patient's previous record after the patient grants him access to data at the central server. Thus doctor-2 can easily check the patient's past medication history and provide further help without any kind of hassle of missing data or records. The above model is modified in various ways to suit the need of protocols or designs, applied in various works, to enhance the structure and meet the objectives. For example [10] shows the incorporation of encryption-decryption function along with transaction code service (TAC) and private key generator (PKG) to provide flexibility along with the security measures to data which was the objective of that paper.

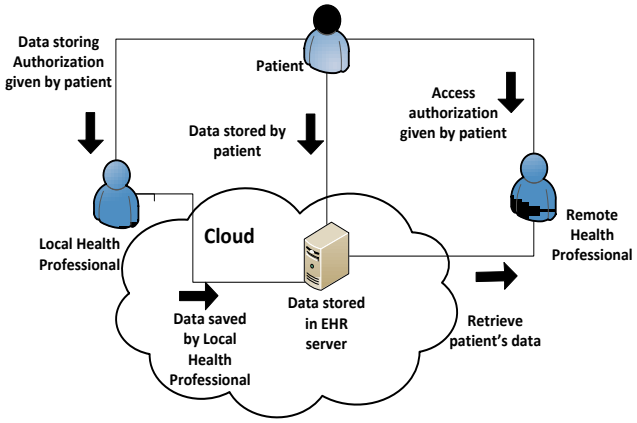

Figure 1. General EHR Management Architecture

3. AVISPA

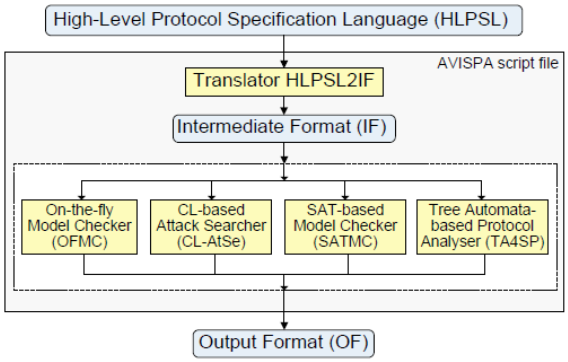

Figure 2. Architecture of the AVISPA tool

AVISPA uses modular and expressive formal language to specify a protocol and its security properties [1]. AVISPA depends on its back-ends, which use an automatic analytic technique to detect any flaws if they exist. The architecture of AVISPA is shown in Figure 2. In AVISPA, protocol roles are modeled in the High-Level Protocol Specification Language (HLPSL) as state transition systems. The HLPSL2IF tool translates a HLPSL specification into an Intermediate Format (IF), a transition system of infinite state. This IF specification is being analyzed by any of the four back-end tools: On-the-Fly Model-Checker (OFMC), CL-based Attack Searcher (CL-AtSe), SAT-based Model Checker (SATMC) and Tree Automata-based Protocol Analyzer (TA4SP). Using the back-end tools of AVISPA, secrecy and different forms of authentication goals could be validated. Secrecy is verified with the help of goal predicate secret (E, id, S), which confirms that the secret information E should be known only by the agents of set S. The label id (of type protocol_id) is used to identify the goal. In the HLPSL goal section, the statement secrecy_of id should be given to refer to it. An intruder attempting to break a secret mounts different attacks. On a successful attack, the intruder learns a value that is considered as secret and that he is not allowed to know, and hence, the secrecy property is violated.

4. EHR Management PROTOCOL

4.1 Security Requirements

The proposed protocol has been designed to maintain the following security requirements:

1. Patient will control the access of his/her Electronic Health Record (EHR). Only patient-authorized health professional (e.g., a doctor) will have access to EHR.
2. Access control will be both doctor and health record specific, i.e., doctor D has granted access to patient's health record with a record-specific transaction code (TAC).
3. Although symmetric encryption will be used, an indirect authentication of the doctor will be performed by the key server based on the doctor's attributes.
4. Data encryption key will be generated by the key server and will be sent to the authenticated doctor only. This key will be known by the key server and the doctor only.
5. EHR will be known by the patient and the doctor only.
6. The patient shares a master key, M with the key server. M will be used in calculating data encryption key.

4.2 Proposed Protocol

The proposed protocol is shown in Figure 3. Table 1 shows the notations being used in this protocol.

Message 1: P → KS : P.TAC

Message 2: P → D : TAC

Message 3: D → KS : P.TAC

Message 4: KS → D : {K}_K_{dks}

Message 5: D → DS : {EHR}_K

Figure 3. Message sequence of the proposed protocol

The following four entities or principals are involved here:

- Patient (P): Carries a health card with secret key chip.
- Doctor (D): Generates Electronic Health Record (EHR) and stores encrypted EHR to the Data Store (DS).
- Key Server (KS): Authenticates the doctor (D) based on attributes. Generates encryption key to encrypt EHR. Forwards the key to the authenticated and authorized doctor (D).
- Data Store (DS): Stores encrypted EHR, however does not understand the content of the EHR.

Table 1. The notations being used in this protocol

Notations	Meaning
P, D, KS, DS	Principals: Patient, Doctor, Key server and Data store
K_{chip}	Secret key written in the patient's chip card
timestamp	Time of an instance
genTAC(), h()	One way function used to generate TAC and hash values
TAC	Transaction Code
K_{dks}	A symmetric key between Doctor and Key server
K	Data encryption key
M	Master key of a patient shared between the key server and the patient
⊕	XOR function
EHR	Electronic Health Record to be stored

In the following we explain each message and how the message ensures the Security Requirements (SR) presented in section 4.1. For example, the first message warrants SR2 (i.e., the second security requirement).

Message 1: The patient generates the TAC where TAC = genTAC(Kchip, timestamp) and then sends the TAC to the key server with the patient's ID, P. Here, due to the use of genTAC() function, K_{chip} and timestamp, the TAC value will be unique for every transaction (storing or retrieving data). (SR2)

Message 2: The patient forwards this TAC to the doctor. (SR1, SR2)

Message 3: The doctor sends the patient's ID, P and the TAC to the key server. The key server performs the following two tasks:

1) Matches the received TAC with the one sent by the patient earlier. (SR2)
2) Checks if the tuple P.D is member of its attribute set. It is assumed that this attribute set had been defined by the patient earlier. (SR1, SR3)

Message 4: The Key server generates the data encryption symmetric key K where K = h(P.TAC) ⊕ M and then forwards K to the doctor encrypted with K_{dks}. Although, M is being used in calculating K, this will be through a one-way function and M could not be derived from K through inverse calculation. (SR4, SR6)

Message 5: Finally, the doctor encrypts EHR with the data encryption key, K and sends the encrypted EHR to the Data Store to be saved in. (SR5)

5. VALIDATION USING AVISPA

Before explaining the model we have developed we would like to present how attributes could be modeled in AVISPA.

5.1 Modeling Attributes in AVISPA

According to ABE-based authentications, the patient should have provision to select the attributes of the doctor, up on which the doctor will be allowed to access the patient's EHR. The patient may select a number of attributes including the identity of the doctor, the field of specialization and the location of the doctor. While modeling matching of the attributes, we use compound data type set, whose elements are of the compound type, e.g., agent. agent. public_key. The identities selected by the patients will be values of the compound type. HLPSL provides a set membership function, in() to check if a tuple is already member of the set.

5.2 AVISPA Model

In the AVISPA model we develop, there are four agent roles: patient (P), doctor (D), keyserver (KS) and datastore (DS). Figure 4 shows the HLPSL specification that we develop to model these four roles. AVISPA is able to validate the secrecy of any message component. As we have mentioned in section 4.1, data encryption key (K), EHR and shared secret (M) should be known by specific agents. To validate the secrecy of these three parameters, the HLPSL goal secrecy_of and the following three secret facts have been added:

- secret(K', sec_k,{KS,D})
- secret(EHR, sec_ehr,{D})
- secret(M, sec_m,{KS,P})

Here, the labels sec_k, sec_ehr and sec_m identify the goals and {KS,D}), {D} and {KS,P} are the sets of agents that are allowed to learn the value K, EHR and M respectively.

```
role patient(P, D, KS, DS: agent, Kchip: symmetric_key, GenTAC : hash_func, SND, RCV: channel (dy))
                                                        played_by P def=

 local State : nat, Tstamp : text, TAC : hash(symmetric_key.text)
 init State := 0

 transition
  0.  State  = 0 /\ RCV(start) =|> State':= 2 /\ Tstamp' := new() /\ TAC' := GenTAC(Kchip.Tstamp')
                                  /\ SND(P.TAC') /\ SND(TAC')
end role


role doctor(P, D, KS, DS: agent, GenTAC : hash_func, EHR : text, Kdks : symmetric_key,
                        SND, RCV: channel (dy)) played_by D def=

 local State : nat, TAC : hash(symmetric_key.text), K : message
 init State := 1

 transition
  1.  State  = 1 /\ RCV(TAC') =|> State':= 3 /\ SND(D.P.TAC')
  3.  State  = 3 /\ RCV({K'}_Kdks) =|> State':= 5 /\ SND({EHR}_K') /\ secret(EHR, sec_ehr,{D})
end role


role keyserver(P, D, KS, DS: agent, M : symmetric_key, H : hash_func, Attribute: (agent.agent) set,
            Kdks : symmetric_key, SND, RCV: channel (dy)) played_by KS def=

 local State : nat, TAC :  hash(symmetric_key.text), K: message
 init State := 10

 transition
  10. State = 10 /\ RCV(P.TAC') =|> State':= 20
  20. State = 20 /\ RCV(D.P.TAC) /\ in(P.D, Attribute) =|> State':= 30/\ K':=xor(H(P.TAC),M)
            /\SND({K'}_Kdks)/\ secret(K', sec_k,{KS,D}) /\ secret(M, sec_m,{KS,P})
end role


role datastore(P, D, KS, DS: agent, SND, RCV: channel (dy)) played_by DS def=

 local State : nat, Edata : {text}_message
 init State := 15

          transition  1. State  = 15 /\ RCV(Edata') =|>  State':= 25
end role
```

Figure 4. Roles of the four agents

5.3 Freshness, Replay and MitM Attacks

AVISPA supports a new() function that generates a fresh value at runtime. Since To ensure the freshness of time, we use the new() function inside the patient role that generates the timestamp (Tstamp). Note that in a transition a primed variable (e.g., Tstamp') represents the new value of the variable (Tstamp). This new value has been either learned in the left-hand side or assigned in the right-hand side of the transition. An active intruder, playing simultaneously the role of any communicating node, has been introduced. First the intruder_knowledge has been added in the environment() role.

- To detect the existence of any replay attack, we instantiate two identical session()s of the model in parallel.
- Since the key server and the data store have been considered as trusted entities, the intruder is not allowed to play the roles of these two. Thus, the intruder is playing the roles of the doctor and the patient in the last two session()s. Hence, the agent identities p and d have been replaced by the intruder identity, i. This will also detect Man-in-the-Middle (MitM) or connection hijack attacks if any such attack exists.

5.4 Model Validation and Analysis

We use SPAN (Security Protocol Animator) [6], a graphical tool that executes an HLPSL protocol specification. SPAN provides the message sequence charts (MSC) that shows the simulations steps of all messages. Figure 6 shows the MSC of the model we have developed. During protocol simulation no intruder's role has been added. The sequences of the MSC confirm that the model we have developed is successfully exchanging all messages.

AVISPA supports Dolev–Yao channels [4] (denoted by channel (dy) in the roles) for message transfer, where the intruder has full control over message transfer. The intruder can overhear, intercept and synthesize any message, and his actions are only limited by the constraints of the cryptographic methods used. As we have already mentioned, we have added an active intruder who can play the roles of the patient or the doctor simultaneously. We verify our model using all four back-ends of AVISPA to find an attack if any exists. AVISPA cannot simultaneously detect multiple attacks during a single run. Therefore, we verify the security goals one-by-one. The first two back-ends, OFMC and CL-AtSe for BOUNDED_NUMBER_OF_SESSIONS have reported SAFE. The other two, SATMC and TA4SP, have reported NOT_SUPPORTED and produced INCONCLUSIVE results. Figure 7 shows the validation output of OFMC. Due to the complexity of the model, we have to run OFMC with a bounded depth. Finally, we can conclude that the AVISPA model that we have developed is free from the attacks that AVISPA is able to find. Hence, the security goals of the model that we have mentioned earlier have been validated.

```
role session(P, D, KS, DS: agent, K_chip, M_m,
    K_dks: symmetric_key, Gen_TAC,
    H_h : hash_func, E_HR : text) def=

 local Attributes: (agent.agent) set,
      SP, RP, SD, RD, SKS, RKS, SDS, RDS:
                    channel (dy)
 init Attributes := {p.d}

 composition
   patient(P,D,KS,DS,K_chip,Gen_TAC,SP,RP)
 /\ doctor(P,D,KS,DS,Gen_TAC,E_HR,K_dks,SD,RD)
 /\ keyserver(P,D,KS,DS,M_m,H_h,Attributes,
                  K_dks,SKS,RKS)
 /\ datastore(P,D,KS,DS,SDS,RDS)
end role

role environment() def=
  const p, d, ks, ds : agent,
    kchip,kichip,m,kdks,kiks : symmetric_key,
    gentac, h : hash_func,
    ehr : text,
    sec_ehr, sec_k, sec_m : protocol_id

  intruder_knowledge =
        {p.d, ks, ds, gentac, kiks, kichip}
  composition
   session(p,d,ks,ds,kchip,m,kdks,gentac,h,ehr)
/\ session(p,d,ks,ds,kchip,m,kdks,gentac,h,ehr)
/\ session(p,i,ks,ds,kchip,m,kiks,gentac,h,ehr)
/\ session(i,d,ks,ds,kichip,m,kdks,gentac,h,ehr)
end role

goal
  secrecy_of sec_ehr, sec_k, sec_m
end goal
```

Figure 5. session, environment and goal roles

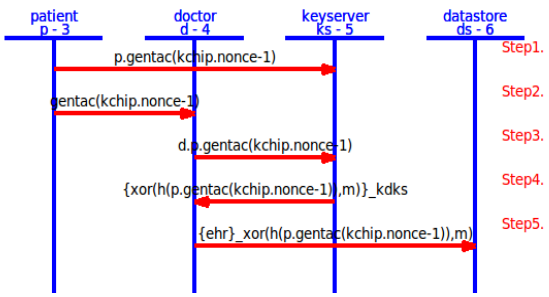

Figure 6. Protocol simulation of the model

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS

PROTOCOL
  /home/salekul/span/testsuite/results/test.if

GOAL  as specified

BACKEND  OFMC

STATISTICS
  TIME 7620 ms
  parseTime 0 ms
  visitedNodes: 14520 nodes
  depth: 11 plies
```

Figure 7. Validation output of OFMC

6. CONCLUSION

The major contribution of the paper is developing a symmetric key-based EHR management protocol. We have successfully introduced the attribute-based access control in symmetric-key solution. Although the protocol we have developed is simple and has not considered many complexities that may arise during deployment in a real-life world, but to our knowledge this is the first AVISPA model of symmetric key-based protocol that also adds attributes. In future, this model can be extended to validate other complex EHR management protocols.

7. REFERENCES

[1] Armando, A. et al. 2005. The AVISPA tool for the automated validation of internet security protocols and applications. *CAV'05: Proceedings of the 17th international conference on Computer Aided Verification*, pp. 281–285.

[2] Barrows, R.C. and Clayton, P. D. 1996. Privacy, confidentiality, and electronic medical records. *Journal of the American Medical Informatics Association: JAMIA*. 3, 2, pp. 139–48.

[3] Blobel, B. 2007. Comparing approaches for advanced e-health security infrastructures. *International Journal of Medical Informatics*. 76, 5–6, pp. 454–459.

[4] Dolev, D. and Yao, A. 1983. On the Security of Public Key Protocols. *IEEE Transactions on Information Theory*. 29, 2, pp. 198–208.

[5] Geron, E. and Wool, A. 2007. CRUST: Cryptographic Remote Untrusted Storage without Public Keys, *Fourth International IEEE Security in Storage Workshop*, pp. 3-14.

[6] Glouche, Y., Genet, T., and Houssay, E. 2006. SPAN: Security Protocol ANimator for AVISPA; User Manual. IRISA/Universit é de Rennes 1: Rennes, France.

[7] Goh, E. et al. 2003. SiRiUS: Securing Remote Untrusted Storage. *Proceedings of the Tenth Network and Distributed System Security Symposium*. pp. 131–145

[8] Goyal, V. et al. 2006. Attribute-based Encryption for Fine-grained Access Control of Encrypted Data. *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 89–98.

[9] Gunter, T. and Terry, N. 2005. The Emergence of National Electronic Heath Record Architechtures in the United States and Australian: Models, Costs, and Questions. *Journal of Medical Internet Research*, 7, 1, e3.

[10] Hupperich, T. et al. 2012. Flexible patient-controlled security for electronic health records. *Proceedings of the 2nd ACM SIGHIT symposium on International health informatics*, pp. 727-732.

[11] Islam, S. 2014. Security analysis of LMAP using AVISPA. *International Journal of Security and Networks*. 9, 1 (2014), 30–39.

[12] Islam, S. 2015. Security Property Validation of the Sensor Network Encryption Protocol (SNEP). *Computers*. 4, 3 (2015), 215–233.

[13] Katt, B. et al. 2009. Privacy and access control for IHE-based systems. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, pp. 145–153.

[14] Narayan, S. et al. 2010. Privacy preserving EHR system using attribute-based infrastructure. *Proceedings of the ACM workshop on Cloud computing security workshop*, pp. 47-52.

[15] Rodrigues, J. J. et al. 2013. Analysis of the security and privacy requirements of cloud-based electronic health records systems. *Journal of Medical Internet Research*. 15, 8, pp. 1–9.

[16] Sahai, A. and Waters, B. 2005. Fuzzy Identity-Based Encryption, *Proceedings of the 24th annual international conference on Theory and Applications of Cryptographic Techniques*, pp. 457–473.

[17] Shang, N. et al. 2010. A privacy-preserving approach to policy-based content dissemination. *International Conference on Data Engineering*, pp. 944–955.

[18] Win, K. T. 2005. A review of security of electronic health records. *Management*. 34, 1, pp. 13–19.