Cryptographic Application Scenarios

Vangelis Karatsiolis, Lucie Langer, Axel Schmidt, Erik Tews, Alexander Wiesmaier

Technische Universität Darmstadt,
Department of Computer Science,
Hochschulstraße 10, D-64289 Darmstadt, Germany
karatsio,langer,axel,e_tews,wiesmaie@cdc.informatik.tu-darmstadt.de

Abstract. The applications that use cryptography as well as the employed devices pose various requirements and constraints. These have to be considered during the development or analysis of cryptographic algorithms that are secure and practicable. This paper presents several real-world cryptographic applications. It also discusses typical cryptographic devices like smartcards. We provide therefore necessary data for evaluating the applicability of cryptographic algorithms.

Keywords: Cryptographic applications, PKI, eVoting, cryptographic devices, DECT.

1 Introduction

Applications like electronic commerce or electronic voting use cryptography in order to be used securely and safely. They use cryptographic primitives like digital signatures, hash functions, or MAC algorithms. Although many efficient and secure algorithms exist, only a limited number of algorithms is practically used. For example there are three cryptographic algorithms that are widely used for digital signatures. These are RSA [RSA78], DSA [NT94], and ECDSA [ANS05]. Most of the hardware devices like smartcards and Hardware Security Modules (HSM) implement a subset of these algorithms. Even software implementations use exclusively these algorithms, although implementations of alternative algorithms already exist (see for example FlexiProvider [Pro]). We estimate that at least 90% of the applications that use electronic signatures employ one or more of these three algorithms.

However, other cryptographic algorithms offer features that are not present in these standard schemes. For example there are algorithms that are post-quantum secure. Before using these algorithms, it is useful to know how they perform when they are used in real-world applications. However, it is not clear which are the basic conditions of various applications and cryptographic devices. In this paper we provide the necessary data for clarifying this.

This paper is organised as follows. In Section 2 we take a look at the different devices that are used for storing key pairs and performing cryptographic operations. Section 3 describes electronic voting systems that are used to replace traditional votings. The cryptographic constraints of a DECT device are discussed in Section 4. PKI based applications are presented in Section 5. In Section 6 we give examples to demonstrate the use of the data presented in the paper and in Section 7 we conclude the paper.

2 Devices

We list a selection of devices that are typically used in cryptographic applications. We provide a minimal technical specification concentrating on the computing power represented as the number of processors and the bit architecture, the available memory, the costs, and the bandwidth for the communication with the device. These are represented as tables.

2.1 Personal Computer

This is the device that most users are working with. It is a standard home computer for personal use.

Type: Standard PC					
Processor	**#Processors**	**Architecture**	**Memory**	**Costs**	**Data rate**
2-3 GHz	1 (2 cores)	32/64 bit	2048 MB	600 €	500 MBit/s

2.2 RFID Tag

RFID tag is a very small hardware device that is able to communicate with other devices over electromagnetic waves. We consider only application specific implementations on RFID tags, namely implementation of a cryptosystem directly in hardware. Therefore, it is not possible to provide general values for the computational strength. These depend on the implementation. The data rate of a typical RFID tag is 50Kbit/s up to 100 Kbit/s. The cost of an RFID tag depends on the number of gates and the number of tags that are produced. It is estimated that every 1000 gates the cost of an RFID increases by 1 cent. For testing the implementation of a cryptosystem it is possible to use field programmable RFID tags. For more on RFID we refer the reader to [Hen08].

2.3 Smartcards

We consider only cards that contain a programmable processor. Many smartcards implement concrete algorithms in hardware and cannot be programmed. We do not consider these cards.

Type: Smartcard					
Processor	**#Processors**	**Architecture**	**Memory**	**Costs**	**Data rate**
1-33 MHz	1	16 bit	504 Kbyte (for code and data)	15-25 €	9600 Bit/s

2.4 Web Server

This is a strong reliable computer which is normally used as a web server.

Type: Web Server					
Processor	**#Processors**	**Architecture**	**Memory**	**Costs**	**Data rate**
2-3 GHz	1 (4 cores)	32 bit	2048 MB	1000-2000 €	1000 Mbit/s

2.5 Mobile Phone

A typical mobile phone that has a processor.

Type: Mobile Phone					
Processor	**#Processors**	**Architecture**	**Memory**	**Costs**	**Data rate**
400-600 MHz	1	32 bit	120 MB	400-800 €	56-7200 Kbit/s

3 eVoting Application Scenarios

In this section we see two eVoting applications. The first is the election of the Austrian Student Association and the second the parliamentary election in Estonia.

3.1 Electronic election of the Austrian Student Association 2009

The Austrian Students Association (ÖH) is the general university students' representative body in Austria. The ÖH provides students with political and academic representation, information, service, and advice. The ÖH is member of the European Students' Union (ESU). The statutes of the ÖH are regulated in a federal law and an ordinance, the "Hochschülerinnen und Hochschülerschaftsgesetz" (HSG) [HSG98], [HSW05]. The legal regulation explicitly allows for electronic voting [HSW05]. Every two years all Austrian students are entitled to elect the representative bodies of the ÖH. The most recent ÖH election was held in May 2009 (see [oeh09] for details). All students of the 21 Austrian universities matriculated in summer term 2009 were eligible to vote. The students were enabled to cast their vote electronically via Internet. To this end, they were allowed to vote from their home computers or alternatively from voting computers in official polling stations. Electronic votes could be cast from Monday to Friday. After the electronic election period, a second voting period based on classic paper based voting took place. Around 2200 students cast their vote over the Internet [ho09]. Since every voter could cast for several institutions, the number of cast votes was even higher. The deployed software system was the Pnyx.core voting system by Scytl [Scy09]. The system was implemented at the Austrian Federal Computing Centre (Bundesrechenzentrum, BRZ) [BRZ09]. For identification and authentication the students used their electronic Austrian Citizen Card (a smartcard) and respective card reader devices which were distributed at no charge. No further registration process was necessary; the electoral roll was generated using information from the university data network. The description given in this section is taken from [SVLB09].

Table 1: Statistics about the 2009 ÖH elections in Austria

Eligible voters	230,526
Total votes	59,241
Voter turnout	25.7%
E-voters	2,161
Electronic votes	5.363
E-voter turnout	3.6%
E-Voting period	5 days

Technical data

Austrian Students Election 2009	
Cryptographic Primitive	**Specific applications and requirements**
Digital signature generation	Signing of configuration information. Digitally signing ballot data using voter private key. Ballot box servers digitally sign ballot boxes. Electoral Board digitally signs list of decrypted votes and list of receipts identifiers. Election officers use RSA 2048 digital signatures.
Digital signature verification	Server checks digital signature of receipt signing request corresponds to the digital envelope. Validated receipt is issued to the voter by the server, voter prints it. Digital signatures of the votes (receipt signing requests) are checked against the digital certificates of eligible voters and the signed contents. The digital signature of ballot box is validated to verify (authenticity and integrity).
Asymmetric encryption (RSA)	Election private key (RSA 2048) is created and protected by secret sharing scheme (Shamir).
Symmetric encryption (3DES)	Ballot is encrypted using symmetric random key (3DES 192bits, 112bits in practice), this key is encrypted using RSA (hybrid).
Probabilistic encryption	Random asymmetric key provides semantic security to the vote encryption, preventing that votes with the same selected voting options generate the same ciphertext.

Hash	Log entry chaining: Each log entry is chained with the previous one using hash function. Therefore, if any log entry is manipulated or deleted the chain verification will file in the place were the manipulation is done. Log checkpoint: Every configured number of lines or time, a digital signature of the last chain is generated. Therefore, any log entry manipulation attempt invalidates the digitally signed section where this entry is located.
Mixing	Random number permutation is generated and applied to the memory stored votes.
Untappable channel	Air-gapping approach (non-cryptographic).
Secure channel SSL/TLS	Authenticated and confidential connection between client and server.
Secret sharing (EA secret key, Shamir's scheme)	Shares of election key are stored in cryptographic smartcards. The secret sharing scheme allows to define a threshold of members (e.g., 5 of 7) to reconstruct.
Smartcards	Storage of key shares, PIN-protected. Voters use Austrian citizen card (a smartcard) with digital ECC signature algorithm for authentication purposes.
Random numbers	Used for receipt, probabilistic encryption.

3.2 Parliamentary elections: Estonia

In Estonia, legally binding political elections were carried out over the Internet in local elections 2005 and in the 2007 parliamentary elections. Estonia was the first country in the world to institute remote electronic voting for parliamentary elections [Eur06]. In particular, the widespread use of national ID cards was vital for introducing the new voting channel as these cards could be used for online voter authentication.

In total there were 897,243 eligible voters in the 2007 election (see Table 3, cf. `http://www.vvk.ee/index.php?id=11178`). 5.4% of the participating voters cast an electronic vote, which corresponds to 30,275 e-voters. In the 2009 European Parliament Elections, this rate increased significantly to 14.7%.

The Estonian election system allows multiple online votes to be cast by the same person, with each subsequent vote cancelling out the previous one [Rya07]. Internet voting is carried out in advance of the election day. Anyone who has voted online can as well go to a polling station on election day and cast a paper ballot, thus cancelling out the vote cast online. The electronic votes are counted separately and are later added to the rest of the votes.

Table 3: Statistics on Internet Voting in Estonia

	Parliamentary Elections 2007	EU Parliament Elections 2009
eligible voters	897,243	909,326
total voters	555,463	399,181
e-voters	30,275	58,669
e-voter turnout	5.4%	14.7%
Internet voting period	3 days	7 days

The e-voting concept system mimics the method used for absentee voting: The voter fills in the ballot, encrypts it (inner envelope), and signs the encrypted ballot (outer envelope). If the voter is eligible, the outer envelope is removed and the anonymous inner envelope is put into the ballot box (cf. [Est05]).

Architecture

Server-side The following description has been taken from [Est05].

Vote Forwarding Server (VFS) Authenticates the voter with the means of IDcard, displays the candidates of voter's constituency to the voter and receives the encrypted and digitally signed e-vote. The e-vote is immediately sent to the Vote Storage Server and the confirmation received from there is then forwarded to the voter. Ends its work after the close of advance polls. The VFS is the only component of the Central System that is directly accessible from the Internet; all other Central System components are behind an inner firewall and access to them is provided only from the VFS.

Vote Storage Server (VSS) Receives e-votes from the VFS and stores them. After the close of advance polls it removes double votes, cancels the votes by ineligible voters and receives and processes e-vote cancelations. Finally it separates inner envelopes from outer envelopes and readies them for the Vote Counting Application.

Vote Counting Application (VCA) Offline component to which encrypted votes are transmitted with the digital signatures removed. The Vote Counting Server uses the private key of the system, tabulates the votes and outputs the results of e-voting.

Client-side

Clients The devices used by the voters. The clients must have Internet access to establish a network connection to the voting system. The voters need a smartcard reader for their ID cards. For MacOS / Linux, the client-side voting software must be downloaded. For Windows, a Java Applet is loaded into the voter's browser, allowing to encrypt the vote and digitally sign the resulting ciphertext.

Procedures

Setup. A system key pair is generated in a HSM. The public key is integrated into the client software and is used to encrypt the vote. The private key used to decrypt the votes never leaves the HSM and is destroyed after the period for filing complaints has expired. The keys necessary to activate the HSM are distributed among the Election Authorities: The National Election Commission (NEC) holds 7 keys, and two keys belong to the administrators. To activate the HSM, both administrator keys and 4 out of the 7 NEC keys are needed. The key managers have physical (for example a keycard) as well as knowledge-based (PIN-code) authentication devices for communicating with the HSM.

eVoting. The voter accesses the VFS via HTTPS-protocol and identifies himself using his ID-card. The VFS verifies the voter's eligibility and queries the VSS whether this voter has already voted (if so, the voter is informed). Next, the VFS identifies the voter's constituency and queries the candidate list database for the list of candidates in that constituency. The list is displayed to the voter. The voter selects a candidate and confirms his or her choice. The client-side voting application encrypts the vote and a random number with the system public key. The voter signs the ciphertext using his or her national ID card. The client-side voting application transmits the encrypted signed vote to the VFS. The VFS verifies that the digital signature was issued by the same person who authenticated at the start of the session. The VFS forwards the received vote to the Vote Storage Server (VSS). The VSS gets a certificate confirming the validity of the digital signature from the validity confirmation server. This certificate is then added to the signed vote.

Tallying. After the end of the eVoting phase double votes are cancelled, leaving only the last vote cast by each voter. The digital signatures are removed and a list of eVoters is compiled.

The anonymous encrypted votes are transferred to the VCA on an external storage medium (CD). The VCA is connected to the HSM and uses a local database with the candidate lists. The HSM is activated by the key managers. The VCA sends the encrypted votes to the HSM and receives back the decrypted votes. Based on this output the VCA computes the election result.

Technical data The cryptographic primitives used in the Estonian Election System are listed in Table 4.

Table 4: Primitives used in Estonian Election System

Estonian Election System	
Cryptographic Primitive	**Specific requirements**
Digital signature	A vote can be signed using any digital signature certificate which does not have an application field restriction forbidding e-voting [Nat03]. It is not required, but recommended to use an ID card, since this combines authentication and signature functions.
Asymmetric encryption	It is recommended to use the standard PKCS#1 v2.1 encrypting scheme RSAES-OAEP for encrypting votes, i.e. votes are encrypted directly using the RSA algorithm, without interim symmetric encrypting [Nat03]. This limits the length of the vote and does not suit complex voting schemes (multi-choice, with space for remarks).
Secure channel	Communication between the web server and the voter application must be secure. Authentication of the server is primary, encryption of the channel is secondary [Nat03]. The certificate of the web server does not have to be signed by the certification server which the voter's computer trusts as the voter can check the fingerprint of the server certificate.
Smartcards	Storage of private signing key, PIN-protected. The ID card authentication certificate should be used for voter authentication. The application must not buffer the access codes of the voter's ID card digital signature certificate [Nat03].
Data	
Quantity	30,000 – 550,000 voters (100,000 on average)
Time	72 hours

4 Telephony Application Scenarios

In this section we discuss an application scenario from the telephony area, specifically DECT.

4.1 DECT

DECT is a wireless protocol [Ins08a,Ins08b] to transmit (telephony) data over a short distance of 50-300 meters. European DECT phones operate at 1880 to 1900 MHz and use at most 250 mW transmit power. DECT uses TDMA to allow multiple calls on the same frequency. DECT divides 10 ms of time into 24 time slots, each of a length of 0.4167 ms. Usually, if a station sends on time slot i, the other station sends on time slot i+12 mod24. Every time slot transmits in theory 480 bits. A DECT frame consists of a 32 bit long radio preamble (named S-field), a 64 bit long A-field used for control traffic, and a 320 bit long B-field, used for payload. Then, a 4 bit long X-field containing a checksum follows. After that, 60 guard bits follow, which are not sent. Instead, the 60 bits are used as a safety margin to make sure that two neighboring stations do not collide. DECT also supports half-frames (only one half of the frame is used) or double-frames (two consecutive frames are used without any interruption). We concentrate on the full-frames, however implementations should be able to handle other formats as well.

The S-field is never encrypted and cannot be encrypted [Ins08c], because it is just used for synchronization. The X-field is also never encrypted. The contents of the A-field can be partially encrypted. The contents of the B-field, except for checksums can be encrypted.

Requirements for the baseband processor One can safely assume, that every baseband processor is running at a rate of at least 1.152 MHz. If a DECT crypto processor is able to perform a single crypto operation every 0.4167 ms, then it is sufficient for all currently known applications. Assuming that only half of the timeslots available are used actively we have that one cryptographic operation per 0.8334 ms is sufficient. For a full frame using GFSK modulation, at most 360 bits need to be encrypted.

Modulations and keystreams used in DECT Besides full frames, DECT also support double-frames (two consecutive time slots are used to send a single frame) and half-slots (only half a time slot is used to send a single frame). DECT also supports other modulations besides GFSK like DBPSK, DQPSK, D8PSK, 16-QAM, and 64-QAM). As a result, the number of keystream bits varies. The size of the A-field and B-field and the maximum number of keystream bits is shown in Table 5.

Table 5: DECT frame formats, field and keystream sizes (in bits).

Configuration	D-field	A-field	B-field	KSS-max-size	keystream bits
1a	868	64	800	840	1680
1b	868	64	800	840	1680
2	1672	64	1600	1640	3280
3	2476	64	2400	2440	4880
5	3280	64	3200	3240	6480
6	4888	64	4800	4840	9680

5 PKI Application Scenarios

In this section we discuss two projects that regard the installation and use of a public key infrastructure (PKI). The first one is the smartcard based PKI of a university and the second one is the PKI that is required for the issuance of electronic prescriptions.

5.1 TU Darmstadt PKI

This is a smartcard based PKI. About 30.000 smartcards are given to the students and employees of the university. They contain RSA key pairs.

This PKI uses 30.000 smartcards for storing the keys of the end-user. The personalisation[1] of the cards was a time consuming task. It lasted almost two weeks, although one week was desired. The reason for this is the slow key generation in software and the slow transfer of the keys to the card.

The registration authority (RA) is installed in a web server. This is because it processes lots of certification requests especially in the roll-out phase of the PKI. The certification authority (CA) is located in a standard PC. During roll-out the production of certificates lasted several days.

Scenario #1 - Personalisation

TU Darmstadt - Student Certificates	
Cryptographic Primitive	
Key Generation	One 1024 RSA key-pair in software
Key Storing	One 1024 RSA key-pair in a smartcard
Data	
Quantity	25,000
Time	40 hours

[1] Generating key pairs, writing them and the card, and producing information related to the card which is sent to the end-user.

TU Darmstadt - Employee Certificates	
Cryptographic Primitive	
Key Generation	One signature 1024 RSA key-pair in software
Key Generation	One encryption 1024 RSA key-pair in software
Key Storing	Two 1024 RSA key-pairs in a smartcard
Asymmetrical Encryption	1024 RSA of about forty bytes

Data	
Quantity	5,000
Time	20 hours

Scenario #2 - Certification

TU Darmstadt - Certificate Issuance	
Cryptographic Primitive	
Hash Function	SHA-1
Digital Signature	One 1024 RSA key-pair in software

Data	
Quantity	30,000
Time	1-2 days

5.2 Electronic Prescriptions PKI

In Germany it is planned that prescriptions are issued electronically. A big distributed PKI is used. This PKI uses OCSP for verifying the status and the existence of certifcates.

Scenario #3 - OCSP Server for HPC According to [Bun09] it is estimated that about 1 billion OCSP requests are sent to an OCSP server in a year, if every electronic prescription requires an OCSP query. These requests are distributed to the different PKIs that are active. We assume that about 100 PKIs are present.

OCSP Server for HPC	
Cryptographic Primitive	
Hash Function	SHA-256
Digital Signature	One 2048 RSA key-pair in evaluated hardware

Data	
Quantity	1 billion (for 100 PKIs)
Time	1 year (mostly working hours)

6 Examples

6.1 PKI

Consider the personalisation scenario for the students participating in the TU Darmstadt PKI. Let t_k be the time required for creating one key pair in software and t_s the time needed to store the key pair on the card. Therefore following the data extracted from the scenario's table we have:

$$25000(t_k + t_s) = 40 * 60 * 60 \text{ sec.} \tag{1}$$

Since we are using a smartcard roughly $8 * 1024$ bits need to be written. Following the data from Subsection 2.3 we have that $t_s \approx 1$ sec. Therefore the time needed to create a key pair is $t_k \approx (144000 - 25000)/25000 = 4.76$ sec. Note that the time t_s is generously calculated. Relevant timings like the time needed to insert and remove the card are not calculated. Therefore, it is expected that time t_s is about 2–3 seconds which gives a smaller period for the creation of the key pair.

6.2 Telephony

Consider the telephony application scenario. In general, an A-Field has 64 Bit, however only 40 of them are used for C-channel messages. A B-Field can have an arbitrary length and can be fully encrypted. In general, the total length of a keystream segment in bits is:

$$l_f = 40 + l_b - c_b$$

Where l_b is the length of the B-field, and c_b is the number of bits used for checksums in the B-field. The most common value for l_b is 320, usually no checksums in the B-field are used, resulting in $c_b = 0$. The maximum length of a keystream is twice the length of the key stream segment. A device which uses all available timeslots on a single frequency (there are only 12 double-slots instead of 24 slots) must be able to encrypt:

$$12 * 100 * l_f$$

bits per second. A device which holds a single call (24 timeslots, 2 of them used) needs to be able to encrypt:

$$2 * 100 * l_f$$

bits per second. The most common value for l_f is 360, the maximum value for l_f is 4840, which results in at most 5808000 bits per second which need to be encrypted. At most $2 * 24 * 100$ different frames per second are processed on a channel, when half-frames are used.

7 Conclusion

In this paper we have discussed typical cryptographic applications and analyzed scenarios based on them that are used in practice. We therefore provide the designers, analysts, and implementors of cryptographic algorithms with realistic data that can be used for examining the feasibility and applicability of a cryptosystem. We plan to extend and update these application scenarios.

References

[ANS05] American National Standards Institute ANSI. X9.62: Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), November 2005.

[BRZ09] Federal Computing Centre of Austria (in German: Bundesrechenzentrum, BRZ). `http://www.brz.gv.at`, 2009.

[Bun09] Bundesärztekammer, Berlin. Funktionale Spezifikation der OCSP Responder für die PKI der e-Arztausweise Version 2.3.1, May 2009. `http://www.bundesaerztekammer.de/downloads/Funktionale_Spezifikation_der_OCSP_Responder_fuer_die_PKI_der_e-Arztausweise_Version_2.3.1.pdf`.

[Est05] Estonian National Election Committee. E-Voting System – Overview, 2005. `http://www.vvk.ee/public/dok/Yldkirjeldus-eng.pdf`.

[Eur06] European Commission. Online Availability of Public Services: How Is Europe Progressing? Web Based Survey on Electronic Public Services Report of the 6th Measurement, June 2006. `http://ec.europa.eu/information_society/eeurope/i2010/docs/benchmarking/online_availability_2006.pdf`.

[Hen08] D. Henrici. *RFID Security and Privacy – Concepts, Protocols, and Architectures.* Springer-Verlag, 2008.

[ho09] heise online. Österreich: Nur 0,9 Prozent Wahlbeteiligung bei E-Voting. `http://www.heise.de/newsticker/Oesterreich-Nur-0-9-Prozent-Wahlbeteiligung-bei-E-Voting--/meldung/138303`, 2009.

[HSG98] Austrian Students Association Act (in German: Hochschülerinnen- und Hochschülerschaftsgesetz, HSG). `http://www.bmwf.gv.at/submenue/wissenschaft/national/gesetze/studienrecht/hsg_1998/`, 1998.

[HSW05] Austrian Students Association Ordinance (in German: Hochschülerinnen- und Hochschülerschaftswahlordnung, HSWO). `http://www.bmwf.gv.at/wissenschaft/national/gesetze/studienrecht/hswo_2005/`, 2005.

[Ins08a] European Telecommunications Standards Institute. ETSI EN 300 175-3 V2.1.1: Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 3: Medium Access Control (MAC) layer, Nov 2008.

[Ins08b] European Telecommunications Standards Institute. ETSI EN 300 175-4 V2.1.1: Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 4: Data Link Control (DLC) layer, Nov 2008.

[Ins08c] European Telecommunications Standards Institute. ETSI EN 300 175-7 V2.1.1: Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 7: Security Features, Nov 2008.

[Nat03] National Electoral Committee. E-voting conception security: analysis and measures. Annex 2B. to Order Contract TL7 Realisation of e-voting software, 2003.

[NT94] National Institute of Standards NIST and Technology. FIPS 186 – Digital Signature Standard (DSS). http://www.itl.nist.gov/fipspubs/fip186.htm, May 1994.

[oeh09] Official website for the Austrian Students Association Election. https://oeh-wahl.gv.at, 2009.

[Pro] The FlexiProvider Project. FlexiProvider. www.flexiprovider.de.

[RSA78] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. 21(2):120–126, February 1978.

[Rya07] Mark D. Ryan. Electronic voting: theory and practice, 2007. http://www.cs.bham.ac.uk/~mdr/teaching/modules06/netsec/lectures/evoting4.pdf.

[Scy09] Scytl. Pnyx.core: The Key to Enabling Reliable Electronic Elections. http://www.scytl.com/pdf/PNYXDREWhitePaper.pdf, 2009.

[SVLB09] A. Schmidt, M. Volkamer, L. Langer, and J. Buchmann. Specification of a Voting Service Provider. 17th IEEE International Requirements Engineering Conference 2009, 2009.