On Key-Sizes for Electronic Voting

Sriramkrishnan Srinivasan Chris Culnane James Heather Steve Schneider
Zhe Xia
University of Surrey, Guildford, GU2 7XH

Abstract

A wide array of end-to-end (e2e) verifiable voting schemes have been proposed. A central feature of these schemes is that votes are encrypted and posted to a publicly accessible bulletin board. It has been observed that the encrypted votes persist indefinitely and can in principle be decrypted to reveal plaintext votes although this should be made infeasible through judicious selection of security parameters i.e. key-sizes. However, there is little discussion or consensus on the specific details of what constitutes an acceptable key-size in an e2e scheme. This paper attempts to discuss this question in detail. We believe that the general key-size estimations must be interpreted specifically when considering an application like e2e voting, especially in high stakes national elections. We discuss the implications of interpreting and applying these key-sizes recommendations to e2e schemes. We also make some comments to encourage a broader discussion and consensus on the issue of determining key-sizes for e2e schemes and the challenges arising in transforming the theory of e2e schemes to practice.

1 Introduction

There has been a significant interest in electronic voting in recent years. Governments are eager to move towards new technology to make the voting process easier and the counting faster and more accurate and there has been an explosion in the use of Direct Recording Equipment (DRE) in many countries. Unfortunately these new technologies have rather poor verifiability properties and the transition from traditional voting to DRE has not been smooth, with many experts expressing their reservations [15, 17]. Citizens are eager not only to obtain assurances of the secrecy of their vote but want to be assured of the integrity of the tally, rather than blindly trusting the authorities and the chain of custody of the ballot boxes and voting machines. There is often a tension when it comes to the general population being assured of the correctness of the tally, especially when the tallies are so close that a swing of a few votes can mean the difference between win or lose for the candidates in question. Moreover, the many vulnerabilities that have been found [4, 25, 5] in these often closed source technologies have eroded public confidence.

Governments are also eager to roll out Internet voting with the aim of improving voter participation and convenience. While there are many advocates of Internet voting and this is seen as an important major step forward in enabling the democratic process for disabled voters who can vote from the comfort of their homes and with their customized accessibility equipment [39] and for making the voting process easier for offshore voters, there is also a concern within the security community that Internet voting is infeasible due to the many threats from malicious hackers, denial-of-service attacks and due to the inherent insecurity of the client-side devices [21, 35]. Again, verifiability of Internet voting remains a concern. Never-the-less, there has been a strong push towards Internet voting. Professional Bodies such as ACM and IEEE have already been conducting voting over the Internet and Estonia even conducts legally binding national elections where voters have the option of casting their votes over the Internet.

1.1 End-to-End Verifiable Voting

While the efficiency and convenience aspects of these new technologies are laudable, it is important that they are perceived to be trustworthy by the voting public and by experts. "Trust but verify" seems to be appropriate way to capture this sentiment. End-to-End verifiable (e2e) voting technologies, for both DRE and Internet voting, aim to provide voters the ability to verify that their votes have been "cast as intended" and allow any interested party to verify that their votes have been "counted as cast". The former is termed "individual veri-

fiability" and the latter "universal verifiability" in the literature. These new verifiability properties are achieved through the use of complex and intricate cryptographic techniques spanning the complete spectrum of modern cryptography from mixnets [9], homomorphic encryption [33] and Zero Knowledge Proofs [19] to name but a few.

A wide array of e2e voting schemes have been proposed. They are classified in a number of ways, depending on the way they are meant to be used. There are e2e Internet voting schemes [22, 12, 1], e2e DRE based schemes[31] and finally schemes such as [11, 8, 37] which use a combination of paper and electronics. e2e schemes are also classified based on the cryptographic techniques employed, for example mixnet based [38] or homomorphic encryption based [36]. The Scantegrity e2e scheme [11] has been trialled in a legally binding election [7] and the IACR has trialled the Helios system in its 2010 presidential election [20].

Whatever the classification, a central feature of these schemes is that votes are encrypted and posted to a publicly accessible bulletin board, allowing voters to verify that their votes have not been modified and have contributed to the tally, while also allowing the encrypted votes to be verifiably tallied by a threshold set of authorities in a way that the voter-vote relationship is broken. Voters and observers can obtain very strong guarantees that all votes have contributed in the manner intended, that no ballots have been added or deleted and that the tally is correct. At the same time, voters are protected from adversaries by not being able to produce a receipt corresponding their votes. This simultaneously upholds the integrity of the election as it deters vote buying/selling.

1.2 On Integrity vs Confidentiality

These new verifiability features provide very strong guarantees in the integrity of the election. While these new verifiability properties are desirable, it is important to note that given the essentially digital nature of the encrypted votes published on the bulletin board, they must be assumed to persist indefinitely. It is well understood that even if all other security issues are addressed, votes in the majority of the cryptographic e2e voting scheme cannot in principle be secured "ad infinitum". This means that it is always probable that in the near or distant feature, the encrypted votes on the publicly accessible bulletin board may be decrypted to reveal the plaintext votes thus compromising the confidentiality of voters in an election which is in principle a secret ballot. This introduces an additional dynamic to the world of e2e voting which is often overlooked.

We believe that it is up to the wider civil society to reach a consensus on whether it is acceptable that the confidentiality of a voter can be breached after the lifetime of the voter (possibly even hundreds of years after) and whether the benefits from e2e voting far outweigh this speculation. In this paper we will refrain from debating on this issue. Rather, we will focus on the technical aspects on what constitutes an acceptable key-size for an e2e scheme.

While it is often mentioned that "large" key-sizes must be used to ensure the secrecy of the encrypted votes for any reasonable length of time and to make the task of any potential adversary infeasible, what these key-sizes should actually be is typically less specified. Comprehensive studies exist on the selection of key-sizes for the general applications of cryptography, but to the authors' knowledge no study has specifically addressed the issue of what key-sizes can and should be used in practice in an e2e voting scheme. We believe this is an underspecification that merits further study and this work is an attempt in that direction.

1.3 Our Contributions

In this paper, we will first discuss the general methodology of key-size estimation employed in the available studies. We will discuss the assumptions made in these studies and discuss how many of these assumptions need to be carefully analysed when considering a high security setting like a national election which has its own unique requirements when compared to the more general application scenarios for which key-size recommendations are made. We will also attempt to interpret these recommendations specifically for the e2e setting and briefly discuss the implications of interpreting and applying these key-size recommendations to the deployment of e2e voting schemes.

1.4 Related Work

Relatively few works exist that tackle the many real life problems associated with implementing e2e schemes in practice. Notable among these is the work by Karlof et. al [23] that studies the Direct Recording Equipment (DRE) based e2e schemes of Neff [31] and Chaum [10] and lists many possible attacks on these schemes when used in practice. Many of these attacks can be extended in principle to cover most of the well known schemes. The authors also propose some mitigations against some of the attacks but note that they cannot be completely excluded given that these and other existing academic works and real world implementation of e2e voting schemes are typically underspecified. We believe that the issue of key-sizes is also a fundamental underspecification that surfaces when translating the theory to

practice and we attempt to tackle this issue in this paper.

We mention here the important works undertaken by Moran et. al. [29, 30]. They give e2e voting schemes that can provide "everlasting privacy". However, we note that in the former scheme, the voter communicates the vote directly to the voting terminal and while the DRE cannot change the results if the voter carries out the interaction with the DRE as specified in the protocol, a malicious (passive) DRE could simply record all the voter-vote relationships and announce it to an adversarial entity via another channel. The interaction of the voter with the terminal is also rather intricate and there are many subtle ways a malicious (active) DRE can cheat the voter undetectably by deviating from the protocol steps [23]. Trust is also not distributed. In the latter scheme, the user interface is rather complicated. Although trust is distributed between two authorities, we believe that ideally trust must be distributed between as many different entities as possible. Never-the-less, the works remain important landmarks in the world of e2e voting.

1.5 Structure of this paper

In Section 2 we give an overview of the methodology employed and the accepted state of the art results in key-size estimation. In Section 3 we use the information from Section 2 to reason about key-sizes in the e2e setting.

2 Key-size Estimation

The only "perfect" or unconditionally secure cryptosystem is the one-time pad. Shannon showed that a message which is encrypted with a random key that is as long as the message, such that the key is never re-used, cannot be broken even by a computationally unbounded adversary [40]. The one-time pad, despite having been deployed (supposedly to encrypt a hotline between the USSR and the USA at the height of the cold war and to encrypt very high value transactions by banks) is unusable more widely in practice, due to the overwhelming requirements on the key. Unconditional security is not achievable in the public key cryptography (PKC) setting. Given a public key, and given the time and computational power, an adversary can always deduce the private key and decrypt all messages.

In the real world, we are limited to using cryptographic schemes, both symmetric and asymmetric, based on computational assumptions i.e. schemes that are secure against computationally bounded adversaries. Symmetric key cryptography, on its own is generally limited in the scope of its use due to the challenges posed by over-whelming key-management issues. Until the advent of PKC, cryptography remained in the purview of government agencies and large financial institutions and out of reach of the masses. Compared to symmetric key encryption, PKC is typically a rather expensive operation. However, the most widespread use of PKC is to transport symmetric keys. Symmetric keys which can be used to encrypt bulk data are encrypted with the recipient's public key such that the recipient can recover the symmetric key with their private key. This greatly simplifies symmetric key-management.

In the academic community where cryptographic research is conducted, the security of cryptographic schemes is typically stated in terms of an abstract security parameter(s). In the real world, when a cryptographic scheme needs to be implemented, this abstract security parameter needs a concrete value. In practice, this translates to a decision regarding the key-sizes. It is well understood that in the absence of other weaknesses, longer keys, whether symmetric or asymmetric, give more security - at the cost of reduced performance. However it is typically less understood what concrete key-sizes should be used for differing security requirements.

Many academic studies that follow a concrete and well established methodology on the selection of key-sizes for different security requirements, in both the symmetric and asymmetric setting exist. The study by Blaze et. al [6] on determining key-sizes for symmetric schemes and the studies by Lenstra and Verhaul [26, 27] on determining key-sizes for symmetric schemes as well as the more complicated and nuanced task of determining key-size equivalences between symmetric and asymmetric schemes remain important guidelines for the wider community. NIST publishes its own key-size recommendations [32] based on various sources as does the ECRYPT II network [16].

In the following section, we will discuss the salient points from these studies. We will not attempt to cover the methodology used in these studies in detail but instead summarize the main points and leave the details to the original sources. We note that the various studies are broadly in alignment. As the ECRYPT II document [16] is the most recent publication and distils information from all the other sources and provides clear guidelines for practitioners, we will frequently refer to it in the following sections.

2.1 Methodology

Although there is no magic formula for estimating key-size, the various studies broadly follow the strategy outline below. The following factors are considered.

- The lifetime and value of the data being secured.
- The attack model.

- Who is the attacker?
- What are the resources (hardware/software) at their disposal?
- How will the resources develop during the lifetime of the data? (for example Moore's Law)
- The inevitability of brute force attacks, state-of-the-art brute force attacks and existing better-than-brute-force attacks.
- The possible cryptanalytic progress that may occur during the lifetime of the protected data.

So for example, the lifetime of the asset may be 20 years and may need to be secured against an intelligence agency which has huge resources at its disposal. Or, the security may only be desired against curious hackers with desktop PC's for a period of 3 weeks. Basically, a "security level" is determined for the asset in question by this process and key-sizes are extrapolated to match the desired security level. Establishing the lifetime and value of the data is a central factor in key-size estimation and has important implications when discussing the security e2e voting in high-stakes elections. Implementation issues such as faults in the implementation, where there is deviation from the theory or from a prescribed standard, biased keys etc. are out of scope of the key-size estimation process, as are attacks on live implementations like timing, power, cache and fault analysis.

2.2 Symmetric Key-size Estimation

The first paper that attempted to provide clear guidelines to businesses as to the size of the symmetric keys they should employ was the paper by Blaze et. al [6]. They estimated the time various classes of attackers would take to recover a 40 bit (RC4) and a 56 bit (DES) key. The study only considered brute force attacks against these ciphers so the results and the methodology are applicable to all symmetric ciphers which have no "structural" weaknesses. Table 1 is reproduced from [6] and summarizes their results.

It is important to recall the conclusions of this early study. As early as 1995, a 40 bit key was considered to be trivial to recover by brute force. A hacker using only scavenged computer time could recover a key in a week. DES was increasingly inadequate - a small business could expect data secured with DES to be secure for 18 months against a competitor willing to spend $10,000 but only for 19 days against someone willing to spend $300K.

The authors estimated that "for an opponent whose budget lay in the $10M–$300M range, the time required to search out keys in a 75 bit key-space would be between 6 years and 70 days...under our amortization assumptions a cost of $19M and a recovery rate of only 5 keys a year. The victims of such an attack would have to be fat targets indeed". It is instructive to ask "How fat a target is a high stakes election?" We will return to this question in Section 3.

The authors concluded their report by recommending a minimum key-length of 90 bits for symmetric cryptosystems. They arrive at this recommendation by imposing the requirement that encrypted data should still be secure for 20 years and taking Moore's Law into account, adding an additional 14 bits to the minimum 75 bits required to protect against an attack in 1995.

Although the estimates in Table 1 are largely conservative, it is useful to recall that hackers have in recent times carried out attacks considered to be way beyond their capabilities. In response to Simon Singh's Code Book Challenge, hackers recovered 48-bits of a DES key in 3 weeks with a small number of PC's. The key-search was distributed as part of a screen saver application over the internet [3].

2.2.1 The Current State of the Art

Table 2 is reproduced from the ECRYPT II report [16] shows the recommendations regarding symmetric key-sizes as of 2011-2012 and provides a simple-to-interpret reference for practitioners. The overall spirit of the recommendations is in alignment with the recommendations from [6]. This study specifies various security levels, what kind of attacker they can be considered to protect against and finally the size of symmetric key necessary for each security level.

2.3 Asymmetric Key-size Estimation

It is relatively straightforward to estimate the key-sizes in the symmetric setting as this is done under the basic assumption that brute force attacks are the only attacks possible. However, the situation is not quite so simple when determining key-sizes in the asymmetric setting. The majority of asymmetric encryption schemes in widespread use are based either on the infeasibility of the factorization or the discrete logarithm problem [34, 18]. Unfortunately, increasingly effective attacks have been found against both the factorization and the discrete logarithm problem over Finite Fields, all better than brute force. It is also difficult to gauge what cryptanalytic progress will be made in the future, as a consequence of which, asymmetric key-sizes grow very conservatively at higher security levels.

Table 3 shows the ECRYPT II recommendations with equivalent (symmetric) key-sizes for cryptosystems based on factorization and the discrete logarithm prob-

Type of Attacker	Budget	Tool	Time(Cost) Per Key		Key-length in 1995
			40 bits	56 bits	
Hacker	Tiny	PC	1 week	infeasible	45
	$400	FPGA	5 hours ($0.08)	38 years ($5,000)	50
Small Business	$10,000	FPGA	12 min ($0.08)	18 months ($5,000)	55
Corporate	$300K	FPGA	24 sec ($0.08)	19 days ($5,000)	60
		ASIC	0.18 sec ($0.001)	3 hours ($38)	
Big Company	$10M	FPGA	0.7 sec ($0.08)	13 hours ($5,000)	70
		ASIC	0.005 sec ($0.001)	6 minutes ($38)	
Agency	$300M	ASIC	0.0002 sec ($0.001)	12 sec ($38)	75

Table 1: Symmetric key-size recommendations, from [6]

Security Level	Security (bits)	Protection	Comment
1	32	Attacks in "real time" by individuals	Only acceptable for authentication tag size
2	64	Very short-term protection against small organizations	Should not be used for confidentiality in new systems
3	72	Short term protection against medium organizations, medium-term protection against small organizations	
4	80	Very short term protection against agencies, long-term protection against small organizations	Smallest general- purpose level ≤ 2 years protection (e.g. use of 2-key 3DES, $< 2^{40}$ plaintext/ciphertext)
5	96	Legacy standard level	2-key 3DES restricted to 10^6 plaintext/ciphertexts$\approx$ 10 years protection
6	112	Medium-term protection	$\approx$20 years protection (e.g. 3-key 3DES)
7	128	Long-term protection	Good, generic application independent recommendation $\approx$30 Years
8	256	"Foreseeable future"	Good protection against quantum computers unless Shor's algorithm applies

Table 2: Security Levels, from [16]

Security (bits)	RSA	DLOG		EC
	modulus	field size	sub-field	
48	480	480	96	96
56	640	640	112	112
64	816	816	128	128
80	1248	1248	160	160
112	2432	2432	224	224
128	3248	3248	256	256
160	5312	5312	320	320
192	7936	7936	384	384
256	15,424	15,424	512	512

Table 3: Asymmetric Key-Size Equivalence, from [16]

RSA/DLOG key	Security(bits)
512	50
768	62
1024	73
1536	89
2048	103

Table 4: Effective Key-size of commonly used RSA/DLOG Keys, [16]

lem over finite fields, and cryptosystems in the Elliptic Curve setting.

Note the dramatic increase in size of the RSA modulus and DLOG field size at higher security levels, so much so that at the 256-bit security level, an RSA modulus thought to provide equivalent security must have a minimum of 15,424 bits. This has catastrophic implications on performance. It can be argued that at higher security levels, cryptography in the Elliptic Curve setting is currently the only viable option.

Table 4 shows the security levels offered by commonly used sizes of RSA moduli and finite fields in the DLOG setting. Note that the RSA modulus size of 1024 bits which is often used as a default implementation reference only provides security equivalent to a 73-bit symmetric key.

2.4 Key-size Estimation: Summary

There is an important difference between the use of symmetric and asymmetric cryptosystems in practice. By design, the available key-sizes in the standardized block cipher AES are extremely conservative. AES only supports key-sizes of 128, 192 and 256 bits. So, even when the security level required is much lower, in reality a great margin is obtained when using a standardized block cipher like AES. A block cipher like AES takes away the complicated decision regarding key-sizes from the implementer.

The situation is unfortunately not so straight forward when it comes to asymmetric cryptosystems and the decision regarding key-sizes gets delegated to the implementer. When determining the size of the modulus to use in a scheme based on integer factorization or the size of the field in the case of a cryptosystem based in the DLOG setting, much more flexibility is available. Unfortunately this flexibility may mean that the overall security level of the system may not be as desired.

The most widespread use of PKC is in key-transport. Consider a 256-bit AES key. This key-length provides the maximum security level under ECRYPT II recommendations. What is the overall security level when this AES key is encrypted with an RSA public key with a 1024-bit modulus as part of a key-transport mechanism? An attacker against the system could attack the RSA modulus rather than the block cipher, as a 1024-bit RSA modulus only provides security equivalent to an 73-bit symmetric key. In practice this may be acceptable as we assume there is a lifetime associated with the data that is being encrypted. In the time that it may take to factor a 1024-bit RSA modulus it is likely that the data in question is no longer as critical as it originally was. However, if the intention is in fact to have an overall security level corresponding to 256-bits then the RSA modulus used in the key-transport must have 15,424 bits as per ECRYPT II recommendations.

3 On Key-sizes for Voting Schemes

From the general theory of e2e voting it is implicit that there is direct correspondence between individual voters and their encrypted ballots on the Bulletin Board as this is how individual verifiability is achieved - by the voter being able to look up his encrypted vote and ensuring that it has not been modified. In principle, we must consider the scenario that the encrypted ballot can be decrypted to violate voter confidentiality.

The general theory of end-to-end voting has not directly addressed what the required security level of the data on the bulletin board must be in any detail. Most papers on voting mentions key-sizes in passing if at all. We feel this is an oversight which should be addressed in detail. We will now discuss the various factors to be considered when using an e2e voting scheme in a high stakes national election.

3.1 Adversaries and Lifetime of Bulletin Board Data

As we have already mentioned, the most common general use of asymmetric encryption is in key-transport i.e.

to facilitate the sharing of a symmetric key between two or more parties, which is then used for bulk encryption of data. The "heavy lifting" is done by symmetric cryptography as PKC is typically expensive even for small key-sizes. In most application scenarios, frequent re-keying also takes place so that not too much data is encrypted with a single key. There are also many users in the system each with a public/private key-pair rather than a single global key that is the only attack vector.

In the conventional key-size estimation scenarios, assumptions are made on the lifetime of the data as well as the capabilities of the attacker. For a high value bank transaction for example, the cost to the bank should the encryption be broken may be very high, but the time for which the data must be kept secure may be rather small. Assuming the most powerful of adversaries, a key-size which can thwart the adversary for a matter of days may be suitable. A much smaller key-size may be suitable when considering much less capable adversaries.

e2e voting is a rather unique application of cryptography where many of the subtleties mentioned above are pushed to their extremes. The sheer range of cryptographic techniques employed in e2e voting as well as the quantity of pure asymmetric cryptography performed makes it different from any fielded application of cryptography to the authors' knowledge. For example, ballots in schemes based on homomorphic encryption [2, 41] consist of homomorphic ciphertexts corresponding to fixed encodings of the candidates (we will ignore ZK proofs of correctness etc. in this discussion). Cast votes correspond to one or more of these encryptions which are posted to the Bulletin Board. A single global key is used to encrypt all bulletin board data. Furthermore, there are rather severe requirements on both the lifetime of the data as well as the nature of the adversary. It can be argued that the data needs to be secured for the longest possible time against the most powerful of adversaries.

We stated earlier that we believe that it is up to the wider civil society to reach a consensus on the lifetime of the data on a bulletin board i.e. whether it is acceptable that the confidentiality of a voter can be breached at some point in the future. Assuming that we have a consensus that bulletin board data must be secured for the lifetime of the voter at the very least, then, assuming that a voter casts a vote in a national election at age 18 and lives to the ripe old age of 100, the data must be secured for a minimum of 80 years. We see immediately from Table 2 that the only acceptable security level is the highest possible i.e. Level 8.

We note that the overwhelming integrity guarantees can be achieved with much smaller key-sizes. and that our reasoning is primarily due to the confidentiality requirements on a secret ballot. It is tempting to argue that in the light of the overwhelming integrity guarantees obtained from fielding an e2e election scheme with even small asymmetric key-sizes, it may be acceptable to overlook the issue of data lifetime. Given current cryptanalytic progress, it is highly unlikely that a 1024-bit RSA modulus will be factored in the one month (say) from the time it takes to set-up an election to the date the result is announced (the largest RSA modulus that has been factored in a public challenge to date is RSA-768 on December 12, 2009 [24]) we could argue that election integrity cannot be violated and that election integrity is more important that voter confidentiality. However, we feel that such an argument may not find favour with the voting public or lawmakers. It is reasonable to assume that the general population, unaware of the intricate details of cryptography would desire a guarantee of privacy post election, that is at the highest level possible.

In e2e schemes that are based on computational assumptions, the best we can hope to do is to secure the data for the longest possible time and against very powerful adversaries and indeed it seems this is a necessary requirement. This is under the assumption there is a direct correspondence between BB data and voter identities. If it were possible to set up the system so that it is possible to achieve individual verifiability without there being a direct link between voters and their encrypted votes, then, decrypting the bulletin board in a sense only confirms the election tally. However, to our knowledge, such an approach has not as yet been considered in the general theory of e2e voting. It is also important to analyse such an approach thoroughly to make sure it does not introduce other vulnerabilities such as ballot stuffing attacks.

Having reached the conclusion that the highest possible security levels are required on the basis of data lifetime alone, it seems unnecessary to reason about the capabilities of the attacker. However, we will continue to apply the same line of questioning in a systematic manner. It seems to us that Bulletin Board data must be secured against the strongest of adversaries. A bulletin board seems to be an ideal target for any party wishing to discredit the election. This may be a foreign power or disgruntled internal powers. Either way, it seems that in the electronic voting scenario we must assume the most powerful of adversaries. It is also important to note the recent trend of enthusiastic hackers carrying out powerful attacks against challenges [3] as a hobby. It seems that a Bulletin Board with encrypted votes may be a natural curiosity for the hacking community and university computer science departments looking for a challenge. There may be no malice intended but it is easy to see that the repercussions of success may be far reaching and negative. Attempts to discourage attempts to cryptanalyse the Bulletin Board by law may seem excessive and

make the task more enticing. It seems the only alternative is to make the task of cryptanalysing as difficult as possible. This means that that assuming no other flaws exist and attacks on the setup and infrastructure are discounted (for example, key-generation is done securely and private keys are never leaked), the data should be encrypted to the highest security levels possible.

3.2 Impact of Large Key-sizes on e2e Schemes

Many e2e schemes in the literature give estimates for the time it takes to setup the election, create ballots, tabulate an election etc. The estimates are often given for a 1024-bit RSA modulus or 1024-bit field for the DLOG setting. See for example the estimates in Scratch & Vote [2] with a 1024-bit modulus. The authors of Civitas [12] have conducted experiments on the feasibility of implementing their scheme with a 2048-bit modulus.

Table 4 shows the security levels offered by commonly used sizes of RSA moduli and finite fields in the DLOG setting. Unfortunately, a 1024-bit RSA modulus or 1024-bit DLOG field size, which is often used as a default implementation reference only provides security equivalent to a 73-bit symmetric key. 2048-bit keys provide security equivalent to a 103-bit key. As discussed earlier, it can be argued that when considering scenarios like high stakes national elections, with their specific requirements on data lifetime and security against very powerful adversaries, these key-sizes are unsuitable and much larger may be required in practice.

As we have already seen, asymmetric key-sizes grow very conservatively at higher security levels, so much so that at the 256-bit security level, an RSA modulus that is thought to provide equivalent security must have a minimum of 15,424 bits. This has catastrophic implications on performance. In fact for a scheme like Paillier which does its operations modulo n^2 where n is an RSA modulus, the performance hit is even greater, with a Paillier ciphertext having over 30,000 bits at the 256-bit security level.

Table 5, reproduced from [14], gives an indication of the degradation in performance of the El-Gamal [18], Generalized Paillier [13] and Paillier [33] cryptosystems (the generalized Paillier scheme with $s = 1$ is equivalent to the Paillier scheme) which are arguably the most common cryptosystems used in e2e voting schemes. Timings for RSA [34] are provided for reference. Even with a 4096-bit moduli, which gives a security level that lies between 128 and 160 bits, the performance implications are evident.

To reason about speeds for the 15,424-bit security level, we can assume that efficient algorithms like Karatsuba multiplication are used where each doubling of the modulus size, triples the time for a modular multiplication [28] and extrapolate the speeds. The timings are for unoptimized Java code, but even with optimized code and powerful hardware it seems that the requirement for very high security levels will impact the viability of many of the e2e schemes in the literature in practice, especially for elections with a large number of voters. In reality, when reasoning about the viability of e2e schemes we need to think of more complicated operations such as the generations of proofs, checking of proofs of correctness, proofs of mixing, distributed decryption and various communication and storage overheads rather than just primitive operations, but it can be argued that all of these eventually boil down to the speed and nature of these primitive operations.

The Elliptic Curve setting seems to provide the only reasonable key-sizes for very high security levels. Unfortunately, no secure homomorphic cryptosystems exists in this setting. Of course this does not rule out schemes based on mixnets and significant performance gains can be achieved by moving to the elliptic curve setting.

Taking the discussion so far into account, we attempt to provide our own recommendations on the acceptable security levels for various types of elections in Table 6. We base our recommendations on the conventional wisdom on key-size estimation and take into account the issues specific to an application scenario like electronic voting.

4 Conclusion

There is a comprehensive body of literature on the theory of e2e voting. However, like all new technologies, many new challenges are thrown up and many details remain unspecified when it comes to the translation of the theory into practice. We have attempted in this work to address some of these new challenges. Specifically, we have attempted to extend the standard wisdom on key-size estimation to the e2e setting. If e2e voting is to become widespread, it is important to study the trade-off involved when balancing integrity and confidentiality and voters must be made awareness of the new integrity properties as well as these trade-offs. We hope our work will engage the cryptographic as well as non-experts in understanding the subtle implications of these new technologies and help in arriving at a consensus and in the establishment of clear guidelines.

5 Acknowledgements

References

[1] ADIDA, B. Helios: Web-based open-audit voting. In *USENIX Security Symposium* (2008), P. C. van Oorschot, Ed., USENIX

	El-Gamal		Generalized Paillier						RSA	
			s=2	s=4	s=3	s=1	s=2	s=1		
Security	2048	4096	1024	1024	1366	2048	2048	4096	2048	4096
Ciphertext Size	4096	8192	3072	5120	5464	4096	6144	8192	2048	4096
Encryption(ms)	1980	15205	578	1591	2370	1969	4397	15264	8	32
Decryption(ms)	996	7611	312	873	1281	1030	2290	7779	272	2001

Table 5: Performance of cryptosystems from [14]

Type of Election	Security Level	Comments
Low Stakes: Online poll, game shows etc	2,3	Confidentiality and Verifiability may be desired by voters but desire to attack election confidentiality post-factum low.
Legally Binding, Low stakes: e.g. Student Union Election	4,5,6	The desire for confidentiality may be high. Adversaries may be sufficiently motivated but limited in their capabilities.
Legally Binding, Medium Stake: e.g. Local Council Elections	5,6,7	Security Levels 4,5,6,7 should provide a sufficient financial deterrent
Legally Binding, High Stakes: e.g.National elections	8 (?)	Clearly the highest possible security level is desired. However, it is an open question if this is adequate.

Table 6: Key-size recommendations for e2e Voting schemes

Association, pp. 335–348.

[2] ADIDA, B., AND RIVEST, R. L. Scratch & vote: self-contained paper-based cryptographic voting. In *WPES* (2006), A. Juels and M. Winslett, Eds., ACM, pp. 29–40.

[3] ALMGREN, F., ANDERSSON, G., IVANSSON, L., GRANLUND, T., AND ULFBERG, S. The code book cipher challenge solution page. `http://www.codebook.org/` Last Accessed: 16th April 2011.

[4] BANNET, J., PRICE, D. W., RUDYS, A., SINGER, J., AND WALLACH, D. S. Hack-a-vote: Security issues with electronic voting systems. *IEEE Security & Privacy 2*, 1 (2004), 32–37.

[5] BISHOP, M., AND WAGNER, D. Risks of e-voting. *Communications of the ACM 50* (2007).

[6] BLAZE, M., DIFFIE, W., RIVEST, R., SCHNEIER, B., SHIMOMURA, T., THOMPSON, E., AND WIENER, M. Minimal key lengths for symmetric ciphers to provide adequate commercial security. *Online at http://www. counterpane. com/keylength. html* (1996).

[7] CARBACK, R., CHAUM, D., CLARK, J., CONWAY, J., ESSEX, A., HERRNSON, P., MAYBERRY, T., POPOVENIUC, S., RIVEST, R., SHEN, E., SHERMAN, A., AND VORA, P. Scantegrity II municipal election at Takoma Park: The first E2E binding governmental election with ballot privacy. In *Proceedings of the 19th USENIX Security Symposium* (2010).

[8] CHAUM, D. Punchscan. `http://www.punchscan.org`. Last Accessed: 12 November 2010.

[9] CHAUM, D. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM 24*, 2 (1981), 84–88.

[10] CHAUM, D. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security & Privacy 2*, 1 (2004), 38–47.

[11] CHAUM, D., CARBACK, R., CLARK, J., ESSEX, A., POPOVENIUC, S., RIVEST, R. L., RYAN, P. Y. A., SHEN, E., SHERMAN, A. T., AND VORA, P. L. Scantegrity II: end-to-end verifiability by voters of optical scan elections through confirmation codes. *IEEE Transactions on Information Forensics and Security 4*, 4 (2009), 611–627.

[12] CLARKSON, M. E., CHONG, S., AND MYERS, A. C. Civitas: A secure remote voting system. In *Frontiers of Electronic Voting* (2007), D. Chaum, M. Kutylowski, R. L. Rivest, and P. Y. A. Ryan, Eds., vol. 07311 of *Dagstuhl Seminar Proceedings*, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.

[13] DAMGÅRD, I., AND JURIK, M. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *Public Key Cryptography* (2001), K. Kim, Ed., vol. 1992 of *Lecture Notes in Computer Science*, Springer, pp. 119–136.

[14] DAMGÅRD, I., JURIK, M., AND NIELSEN, J. B. A generalization of paillier's public-key system with applications to electronic voting. *Int. J. Inf. Sec. 9*, 6 (2010), 371–385.

[15] DILL, D. L., SCHNEIER, B., AND SIMONS, B. Voting and technology: who gets to count your vote? *Commun. ACM 46*, 8 (2003), 29–31.

[16] ECRYPTII. Yearly report on algorithms and keysizes (2012), d.spa.20 rev. 1.0, ict-2007-216676, 09/2012.

[17] EVANS, D., AND PAUL, N. Election security: Perception and reality. *IEEE Security & Privacy 2*, 1 (2004), 24–31.

[18] GAMAL, T. E. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO* (1984), pp. 10–18.

[19] GOLDWASSER, S., MICALI, S., AND RACKOFF, C. The knowledge complexity of interactive proof-systems (extended abstract). In *STOC* (1985), ACM, pp. 291–304.

[20] IACR ELECTION COMMITTEE. IACR 2010 election report. www.iacr.org/elections/2010/IACR2010ElectionReport.pdf. Last Accessed: 17 April 2011.

[21] JEFFERSON, D. R., RUBIN, A. D., SIMONS, B., AND WAGNER, D. Analyzing internet voting security. *Commun. ACM 47*, 10 (2004), 59–64.

[22] JUELS, A., CATALANO, D., AND JAKOBSSON, M. Coercion-resistant electronic elections. In *WPES* (2005), V. Atluri, S. D. C. di Vimercati, and R. Dingledine, Eds., ACM, pp. 61–70.

[23] KARLOF, C., SASTRY, N., AND WAGNER, D. Cryptographic voting protocols: A systems perspective. In *USENIX Security Symposium* (2005), pp. 33–50.

[24] KLEINJUNG, T., AOKI, K., FRANKE, J., LENSTRA, A., THOM, E., BOS, J., GAUDRY, P., KRUPPA, A., MONTGOMERY, P., OSVIK, D. A., TE RIELE, H., TIMOFEEV, A., AND ZIMMERMANN, P. Factorization of a 768-bit rsa modulus. Cryptology ePrint Archive, Report 2010/006, 2010. http://eprint.iacr.org/.

[25] KOHNO, T., STUBBLEFIELD, A., RUBIN, A. D., AND WALLACH, D. S. Analysis of an electronic voting system. In *IEEE Symposium on Security and Privacy* (2004), IEEE Computer Society, pp. 27–.

[26] LENSTRA, A. K., AND VERHEUL, E. R. Selecting cryptographic key sizes. In *Public Key Cryptography* (2000), H. Imai and Y. Zheng, Eds., vol. 1751 of *Lecture Notes in Computer Science*, Springer, pp. 446–465.

[27] LENSTRA, A. K., AND VERHEUL, E. R. Selecting cryptographic key sizes. *J. Cryptology 14*, 4 (2001), 255–293.

[28] MENEZES, A. J., VAN OORSCHOT, P. C., AND VANSTONE, S. A. *Handbook of applied cryptography*. CRC press, 1997.

[29] MORAN, T., AND NAOR, M. Receipt-free universally-verifiable voting with everlasting privacy. In *CRYPTO* (2006), C. Dwork, Ed., vol. 4117 of *Lecture Notes in Computer Science*, Springer, pp. 373–392.

[30] MORAN, T., AND NAOR, M. Split-ballot voting: everlasting privacy with distributed trust. In *ACM Conference on Computer and Communications Security* (2007), P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds., ACM, pp. 246–255.

[31] NEFF, C. A. A verifiable secret shuffle and its application to e-voting. In *ACM Conference on Computer and Communications Security* (2001), pp. 116–125.

[32] NIST. Recommendation for key management - part 1, general special publication 800-57, may 2006. http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf.

[33] PAILLIER, P. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT* (1999), pp. 223–238.

[34] RIVEST, R. L., SHAMIR, A., AND ADLEMAN, L. M. A method for obtaining digital signatures and public-key cryptosystems (reprint). *Commun. ACM 26*, 1 (1983), 96–99.

[35] RUBIN, A. D. Security considerations for remote electronic voting. *Commun. ACM 45*, 12 (2002), 39–44.

[36] RYAN, P. Prêt à Voter with Paillier encryption. *Mathematical and Computer Modelling 48*, 9-10 (2008), 1646–1662.

[37] RYAN, P. Y. A., BISMARK, D., HEATHER, J., SCHNEIDER, S., AND XIA, Z. Prêt à voter: a voter-verifiable voting system. *IEEE Transactions on Information Forensics and Security 4*, 4 (2009), 662–673.

[38] SAKO, K., AND KILIAN, J. Receipt-free mix-type voting scheme - a practical solution to the implementation of a voting booth. In *EUROCRYPT* (1995), vol. 921 of *LNCS*, Springer, pp. 393–403.

[39] SCOPE. Polls Apart 4. Campaigning For Accessible Democracy. http://www.pollsapart.org.uk/docs/reports/Polls%20Apart%20 Last Accessed: 29th April 2010.

[40] SHANNON, C. E. Communication theory of secrecy systems. *Bell System Technical Journal 28*, 4 (1949), 656–715.

[41] XIA, Z., CULNANE, C., HEATHER, J., JONKER, H., RYAN, P., SCHNEIDER, S., AND SRINIVASAN, S. Versatile Prêt à Voter: Handling multiple election methods with a unified interface. In *Indocrypt: 11th International Conference on Cryptology in India* (2010).