Cryptographic Algorithms for Communicating Results from Distributed Electronic Voting Systems

Horacio A. VILLAGARCÍA WANZA
Comisión de Investigaciones Científicas de la Provincia de Buenos Aires.
Facultad de Informática, Universidad Nacional de La Plata.
La Plata, 1900 – ARGENTINA.
<hvw@info.unlp.edu.ar>

ABSTRACT

Electronic voting systems are increasingly used in electoral processes ranging from specialized stand alone machines, up to complete paperless and remote voting system. Votes secrecy and confidence are necessary in any electoral process. Public or private key cryptographic systems can be used in LAN or WAN facilities. Low level cryptographic structures and basic algorithms are mentioned. Enhancement of security levels in distributed voting schemes, are shown based in concatenated operations before transmission. Finally, processing time reduction with specialized hardware and mixed cryptosystems are discussed.

Keywords: electronic voting systems, message communication, key based cryptosystems, secrecy algorithms, digital signatures.

VI Workshop de Procesamiento Distribuido y Paralelo (WPDP)

1. INTRODUCTION

Elections allow the populace to choose their representatives and express their preferences for how they will be governed. A voting system must preserve the anonymity of a voter's ballot, must be tamper-resistant, and be comprehensible to and usable by the entire voting population.

In traditional elections, voters go to their home precinct and prove that they are allowed to vote there, by presenting an ID card. After this, the voter is given a validated envelope that allows them to approach a voting booth, choose a piece of paper, make a mark in a preprinted paper or similar for their candidates of choice, save the paper in the official envelope and close it. Later, in presence of voting authorities, the voter put the envelope in a box. When the contest time expired, a hand made count and tabulate vote process must be do in each precinct. Later, the communication of results to a central general office (sometimes a hierarchical path) will produce a preliminary final score subject to a recounting process.

Different types of voting equipment are used to speed up the ballot emission and counting process, but the technologies implemented do not capture the power of the information revolution. There have been several studies on voting systems using computer technologies especially the Internet [1][2][3]. These studies caution against the security risks in tasks of election process: voters authentication, ballot secrecy, communications confidence. Different cryptographic algorithms are suited to provide or enhance security levels.

2. ELECTRONIC VOTING SYSTEMS

Researchers have been working in the electronic voting research area after 1980, with an emphasis in the last decade. Currently there is a consolidate taxonomy for classifying electronic voting systems and well-defined sets of protocols for implementing them. Researchers in the electronic voting field have already reached a consensus pack of four core properties that an electronic voting system should have:

- Accuracy: (1) it is not possible for a vote to be altered, (2) it is not possible for a validated vote to be eliminated from the final tally, and (3) it is no possible for an invalid vote to be counted in the final tally.
- Democracy: (1) it permits only eligible voters to vote and, (2) it ensures that eligible voters vote only once.
- Privacy: (1) neither authorities nor anyone else can link any ballot to the voter who cast it and, (2) no voter can prove that he voted in a particular way.
- Verifiability: anyone can independently verify that all votes have been counted correctly.

Accuracy, democracy and verifiability are, in most cases of today´s electoral systems, assured by the presence of representatives of opposite parties. The privacy property is currently assured by the existence of private voting booth, allowing voters to cast their votes in secrecy.

Electronic voting systems could be a great improvement over current paper system. There are many protocols proposed for electronic voting including prototypes that deal with failures in real word scenarios, such as machine or communications failures [4]. Moreover, "direct recording electronic" voting systems are increasingly worldwide adopted. In these systems, the entire election hinges on the correctness, robustness, and security of the software within the voting terminal. Security analysis conclusions made on a widely used, paperless DRE voting system shows vulnerabilities that made it unsuitable for use in a general public election. Also, the authors suggest that the most viable solution for securing electronic voting machines are voting systems having a “voter-verifiable audit trail”, where an attached printer might print a paper ballot [5].

3. GETTING SECRECY

Election protocols are built from a number of low level cryptographic structures. These structures, alone or in combination, create the various properties we desire in election systems. Symmetric key or Public key cryptosystems are possible solutions [6].

In Symmetric key cryptosystems users share an algorithm for encryption and decryption processes and a unique secret key. Fig.1 shows a common application, DES algorithm was one of the most used in this type of system.

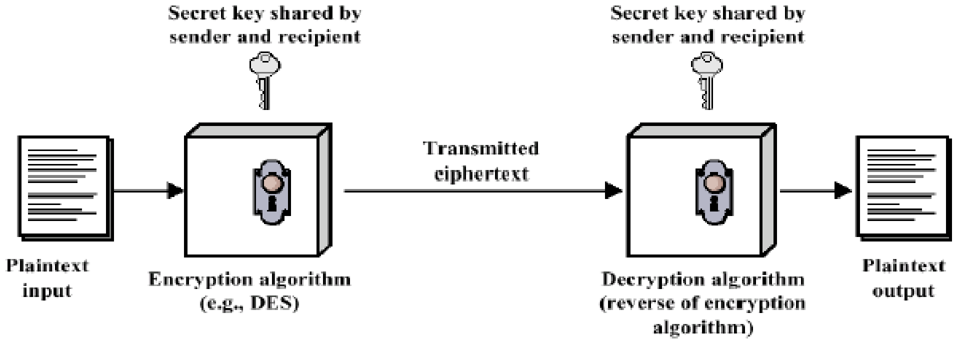
Secret key shared by sender and recipient
Secret key shared by sender and recipient
Transmitted ciphertext
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Fig. 1 Symmetric Key Encryption

Public key cryptosystem have encryption and decryption algorithms, one inverse of the other, and a pair of mathematically related keys where one is of public domain. A description of how a secret message can be sent to Bob in this cryptosystem is depicted in Fig. 2.

In this system, Bob selects the key pair *(e; d)*. Bob sends the encryption key *e* (called the *public key*) to Alice over any channel but keeps the decryption key *d* (called the *private key*) secure and

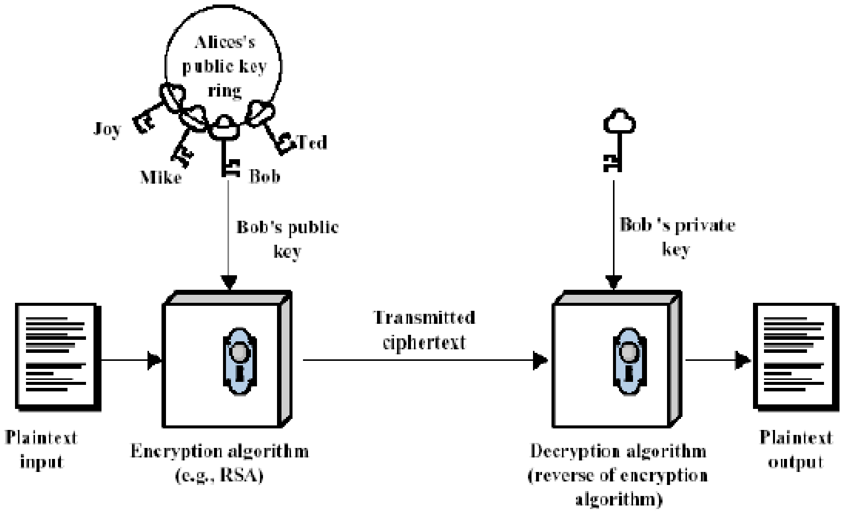

Fig. 2 Public Key Encryption

secret. Alice may subsequently send a plaintext M to Bob by applying the encryption transformation determined by Bob's public key to get $C = E_e (M)$. Bob decrypts the received ciphertext C by applying the inverse transformation D_d uniquely determined by d obtaining $M = D_d (C)$. The system provide confidentiality.

One of the most representative algorithm used in public key cryptosystems was developed by Rivest, Shamir and Adleman in 1977 and is known as RSA algorithm. The RSA scheme is a block cipher, where plaintext M and ciphertext C are integers between 0 and n-1, which are obtained by module n exponentiation. The principles are:

$$C = M^e \bmod n$$
$$M = C^d \bmod n = M^{ed} \bmod n$$

The key pair is defined as:

Public key KU = {e, n}
Private key KR = {d, n}

Private key cryptosystems require secure and safe storage of the involved secret key. Public key encryption, as described, assumes that knowledge of the public key e does not allow computation of the private key d.

4. GETTING AUTHENTICATION

Suppose that Bob, as shown in Fig. 3, send a plaintext M to Alice applying the encryption transformation determined with he's private key d to get $S = E_d(M)$. Because Bob is the only person who knows his private key, he is the only person who can create the ciphertext S for the object M. Alice –as anyone- can verify sender authenticity by applying the inverse transformation D_e determined by Bob public key e to the received ciphertext S, obtaining $M = D_e (S)$. The message S is also known as *digital signature*. As described, the system provide authentication.

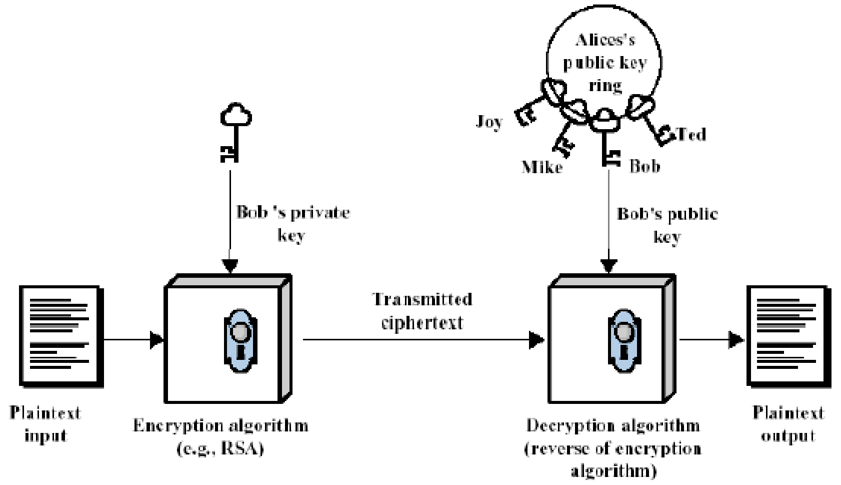

Fig. 3 Public Key Authentication

5. GETTING CONFIDENCE

More complete but computationally intensive application can be achieved if we concatenate authentication and secrecy processes, as shown in Fig 4. We transmit signed secret message Z as result of applying the encryption transformation to the original X plaintext with the sender private key KR_a (signed Y message) and then repeat the encryption transformation with the recipient public key KU_b.

In this case, only the recipient with the correct key, KR_b, can decrypt the received Z message and later confirm senders signature authenticity by decrypting with emitter public key, KU_a. The crytosystem provide authentication of senders and secrecy of the transmitted message.

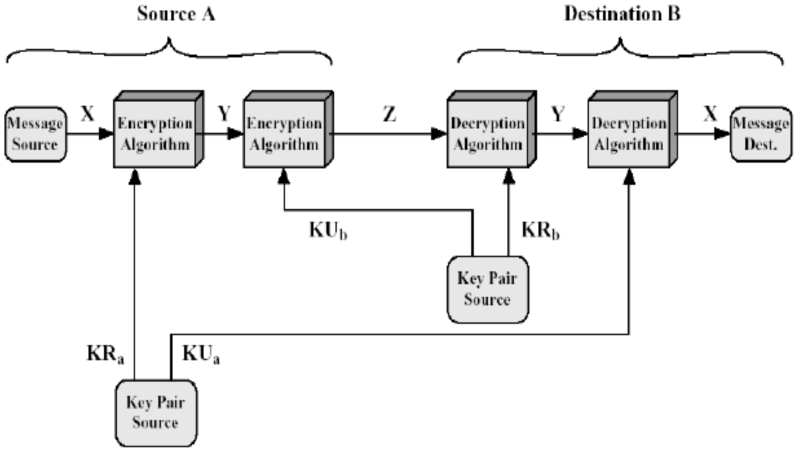

Fig. 4 Secrecy and Authentication

This scheme applied in a general election process have a processing workload peak when the contest time expired and the results, computed in the lower layer, must be transmitted to the upper one following a hierarchical path if necessary. For example, from each precinct to a county central office, then to a district office and following up to reach the general central repository. There exist alternatives to decrease the processing workload either in authentication processes or message cipher/decipher phases. These alternatives are based on improvements in signing methods, use of mixed public and secret key environment, specialized ciphering machines or a well balanced combination of them.

6. DIGITAL SIGNATURE AND HASH ALGORITHMS

The purpose of a digital signature is to provide a means for an entity to bind its identity to a piece of information. The process of *signing* entails transforming the message and some secret information held by the entity into a tag called a *signature*.

Based on particular properties or requirements, different signatures exist and description of two general classes of digital signature schemes, can be briefly summarized as follows:

- Digital signature schemes with appendix require the original message as input to the verification algorithm. These are the most commonly used in practice. They rely on cryptographic hash functions rather than customized redundancy functions, and are less prone to existential forgery attacks
- Digital signature schemes with message recovery do not require the original message as input to the verification algorithm. In this case, the original message is recovered from the signature itself. In practice, this feature is of use for short messages.

Belonging to these classes we can mention the schemes known as one time, arbitrated, blind, undeniable or fail stop signatures.

In particular, blind signature schemes are two-party protocols between a *sender* A and a *signer* B. The basic idea is: A sends a piece of information to B which B signs and returns to A. From this signature, A can compute B's signature on an a priori message M of A's choice. At the completion of the protocol, B knows neither the message M nor the signature associated with it. The purpose of a blind signature is to prevent the signer B from observing the message it signs and the signature; hence, it is later unable to associate the signed message with the sender A. Different electronic voting protocols use blind signature [4][7].

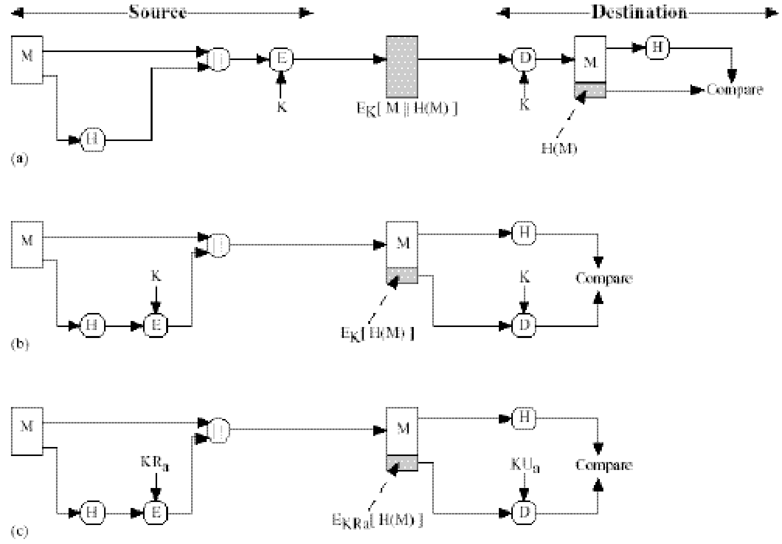

Fig. 5 Basic uses of hash functions

Other one of the fundamental primitives in modern cryptography is the cryptographic hash function, often informally called a one-way hash function. A *hash function* is a computationally efficient function mapping binary strings of arbitrary length to binary strings of some fixed length, called *hash-values*.

If a goal-oriented *functional* classification is considered, we found the following two types of hash functions:

- Modification detection codes (MDCs). Also known as *manipulation detection codes*, and less commonly as *message integrity codes* (MICs), the purpose of an MDC is (informally) to provide a representative image or *hash* of a message, satisfying additional properties. The end goal is to facilitate, in conjunction with additional mechanisms, data integrity assurances as required by specific applications. MDCs are a subclass of *unkeyed* hash functions; we can mention two specific classes of MDCs:
  - *one-way hash functions* (OWHFs): for these, finding an input which hashes to a pre-specified hash-value is difficult;
  - *collision resistant hash functions* (CRHFs): for these, finding any two inputs having the same hash-value is difficult.
- Message authentication codes (MACs). The purpose of a MAC is (informally) to facilitate, without the use of any additional mechanisms, assurances regarding both the source of a message and its integrity. MACs have two functionally distinct parameters, a message input and a secret key; they are a subclass of *keyed* hash functions.

Different basic uses of hash functions are depicted in Figs. 5 and 6. As example of well known hash algorithm we can mention:

- MD2 / MD4 / MD5: produce a 128-bit hash value. These are specified as Internet standards (RFC1320, RFC1186, RFC1321).
- SHA (Secure Hash Algorithm): produces 160-bit hash values. It was designed by NIST & NSA in 1993 and revised in 1995.

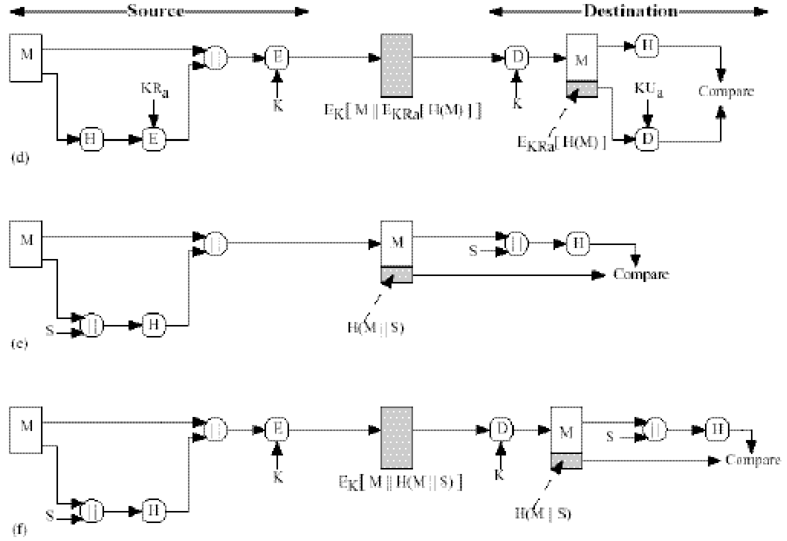

Fig. 6 Basic uses of hash functions (cont.)

7. MIXED SYSTEMS & SPECIALIZED MACHINES

Other solution to workload could be similar to the depicted in Fig. 6(d). The source of information authenticate a hash of the message M with its private key KR$_a$, later the message M and an appended digital signature are enciphered with a secret key K. At destination, message secrecy is revealed using the secret key K. Confidence of message is assured when the appended digital signature is verified with senders public key. The system as described is a mixed solution, symmetric key for secrecy and public key for authentication. Compared with full public key system of Fig. 4, the processing workload decrease because symmetric key approach is less hard to compute than a public key based, better performance is obtained if specialized hardware is used as mentioned below. Although the mixed scheme, use public key cryptosystem to senders authentication, the digital signature is based in a shorter fixed length message. The processing time to signing is shorter to.

Specialized processors or hardware implementations of cryptographic algorithms are also used for improving performance. Candidates for NISTs Advanced Encryption Standard (AES) were restricted to hardware realization of their proposed algorithm. Academic research based in the imposed restriction and FPGA hardware implementation are made [8][9].

8. CONCLUSIONS

We have presented different well known cryptographic algorithms. Carefully applied in electronic voting systems, cryptography can enhance secrecy, authentication and confidence of messages communications. Many remote voting systems include in its proposed protocols some of the mentioned algorithms.

No evaluation of any electronic voting system is made, neither a new one is presented. The main objective is to remark the importance and benefits of the use of ciphering elements, and remember that any distributed electronic voting system has a critical message communication process to do where cryptographic algorithm must be applied.

Computing workload and processing time can be estimated as function of the security level to be obtained. The electronic voting protocol and the cryptographic algorithms implementation (software or hardware) can be optimized if carefully selected. Confidence in communicating results is possible.

10. REFERENCES

[1]. A. D. Rubin, “Security Considerations for Remote Electronic Voting over the Internet ”, **Communications of the ACM**, 45(12), Dec 2002, pp.39-44.

[2]. Caltech/MIT Voting Technology Project, “Where We Have Been, Where We Are Going”, **Project Update**, http://www.vote.caltech.edu/Reports/5_fast_facts.pdf, Jan 2003.

[3]. California Internet Voting Task Force, “A Report on the feasibility of Internet Voting”, **Feasibility Study**, http://www.ss.ca.gov/executive/ivote/4_final_report.doc, Jan 2000.

[4]. R. Joaquim et al., “REVS – A Robust Electronic Voting System”, IADIS International **Journal of WWW/Internet**. Vol. 1, N. 2. Dec 2003, pp.47-63. Also available in http://www.social-informatics.net/evotingtechnical.html.

[5]. T. Kohno et al., “Analysis of an Electronic Voting System”, Proceedings of **2004 IEEE Symposium on Security and Privacy**. Oakland, USA, May 2004, pp.27-40.

[6]. A. J. Menezes et al., **Handbook of Applied Cryptography**, 5fth Ed. CRC Press, 2001. Also available in http://www.cacr.math.uwaterloo.ca/hac/.

[7]. M. Herschberg, “Secure Electronic Voting Using the World Wide Web”, MsC Thesis, MIT, 1997.

[8]. M. Liberatori, J. C. Bonadero, "Minimum Area, Low Cost FPGA Implementation of AES", Actas of **X Congreso Argentino de Ciencias de la Computación – WPDP.** Buenos Aires, ARGENTINA, Oct 2004, pp. 1305-1310.

[9]. J. C Bonadero et al., “Expansión de la clave en Rijndael: Diseño y optimización en VHDL”, Proceedings of **XI Workshop IBERCHIP**, Bahía, BRASIL, March 2005, pp. 150-153. Also available in http://www.iberchip.org/iberchip2005/sesions/a5es.htm