Information SecurityUsingThreshold Cryptography With Paillier Algorithm

[1]Machha.Narender, [2]G.N.Ramesh [3]P.Ranganath

GJCST Classification D.4.6,K.6.5

,

Abstract-The dynamic and cooperative nature of ad hoc networkspresent challenges in securing these networks. There arerecent research efforts in securing ad hoc networks. Amongstsecurity approaches, there are threshold cryptography andauthentication. In this paper we survey the thresholdcryptography based schemes and the authentication schemesthat have been proposed to secure ad hoc networks. Weconclude this paper and identify the challenges and openresearch areas associated with each of these approaches.The idea of threshold cryptography is to protect information(or computation) by fault-tolerantly distributing it among acluster of cooperating computers. First consider thefundamental problem of threshold cryptography, a problemof secure sharing of a secret. A secret sharing scheme allowsone to distribute a piece of secret information among severalservers in a way that meets the following requirements: (1)no group of corrupt servers (smaller than a given threshold)can figure out what the secret is, even if they cooperate; (2)when it becomes necessary that the secret information bereconstructed, a large enough number of servers (a numberlarger than the above threshold) can always do it.

I. Introduction

ThresholdCryptographyis the art of chopping asecret into little bits. Only by possessing morethan a threshold number of bits of the secret canthe secret be determined. Algorithms exist tobreak any secret up such that at least and exactlyM out of N holders of pieces of the secret mustgive approval (and their partial secret or key) inorder to compute the total secret (e.g. 3 of 5, 3 of12, 5 of 12, etc.). Removing probability has acost, though... a secret must be broken into C(N,M-1) pieces and each holder carries (NM+1)/N parts of the whole key... so '3 of 12' ismore expensive per-node than '5 of 12'. (Thesenumbers come from the pigeonhole principle andconstraints: any piece 'pK' must be found on NM+1 holders so that access to a full M secretholdersguarantees 'pK' will be known, whilstaccess to M-1 computers must guarantee thatthere is at least one piece not found, so 'pK' mustNOT be with the other M-1 computers Theminimum number of component 'pK' elements todo this is (N) Choose (M-1). Individual pKelements can be made artificially large in orderto subvert guessing of one or two missing pieces;the combinatory function needn't be straightforward appended. However, Thecomputation and storage cost of this approach ishigh, and it may do well to combine it with somestraightforward split-and-distribute as listedabove; e.g. splitting the 'require 5 of 7 pieces' to'more than 7' people is the natural extension tosplitting 'require 3 of 3 pieces' to '12 people'. Thecombined effect can avoid
the massive costs ofsplitting and storing, say '5 of 20' parts (C(20,4)unique parts, every node holding ~16/20ths oftotal secret, vs. C(7,4) parts with each nodeholding 3/7ths of total secret). The mainadvantage of mixing in this algorithmic divisionapproach is in achieving better guarantees as toredundancy and survivability whilesimultaneously increasing the number of usersone must access to possess the whole secret. E.g.for the other approach, to require 5 users wouldrequire splitting the key into 5 pieces anddivvying that up among, say, 15 people; it wouldtake access to 5 people to gain the secret, and thesecret could be lost by losing 3 people. Splittingit to 5 of 7 first, then dividing the 7 chunksamong 14 people results in 2 different peoplehaving a copy of any given chunk, and the secretwon't be lost before losing 6 people (losing threewhole chunks).

As a security measure, Threshold Cryptography requires that many systems must be compromised prior to taking control of a secret,inherently including resistance to snooping orabuse by any super users of the computationresource (who would have the ability to do so ifthe secret were wholly on one system). It alsoprovides inherent redundancy of the secret... e.g.if you can guarantee that it takes at least and atmost 5 of 12 secret-holders to build the secret,you can guarantee that a failure of up to 7systems is tolerable without failure. With aprobabilistic split, you can easily calculate apercentage chance that the data is unavailable foreach loss of node... and, with intelligent split ofcomponents, you can guarantee that at least somecount of nodes must be lost before the data hasany chance of being lost.In the case of authorization to access a differentsystem (e.g. to control a power plant), securitycan be increased further by demanding that a fewparts of the approval come from –particularpeoplethat are known to still be accessible... andby changing these people at regular intervals.This makes it much more difficult to gain accesseven by compromising the systems... becauseyou can't easily know which particular systemsought to be compromised.

II. Motivation

The strongest reason for using this mechanismover straightforward encryption is that a secretmight need to be available to users that can onlyprovide a -certificate- authorizing access to a fileor service, and the primary

[1]*Machha.Narender, Assistant Professor, HITS College of Engg, machha.narender@gmail.com*
[2]*G.N.Ramesh, Assistant Professor, Bhoj Reddy Engg College , noya.ramesh@gmail.com.*
[3]P.Ranganath, Assistant Professor, Asifia Engg College, ranganathponnaboyina@gmail.com

encryption isn'tagainst any key with which individuals sharelong-term access (there is no shared key). E.g.one can use ThresholdCryptography to encryptfiles or split keys requiring, say, either 'Secret'clearance with 'Power Grid' specialization, or'Top Secret' clearance, represented as acertificate signed by a government master keynot in expiration, and any individual that canprove to M of N systems that he or she possessesthe necessary clearances will be provided thecapability to actually perform the task. Keydistribution is a difficult problem, doubly sowhen you won't trust that any one keydistribution server hasn't been compromised;ThresholdCryptography is one of the moreelegant answers to that particular problem. Avery useful extension of secret sharing isfunction sharing. Its main idea is that a highlysensitive operation, such as decryption orsigning, can be performed by a group ofcooperating servers in such a way that nominority of servers is able to perform thisoperation by themselves, nor would they be ableto prevent the other servers from performing theoperation when it is required.

In many real-life situations, we don't believe that any given person can be trusted, and we may even suspect that a big fraction of all people are dishonest, yet it is reasonable to assume that the majority of people are trustworthy. Similarly, in on-line transactions, we may doubt that a given server can be trusted, but we hope that the majority of servers are working properly. Based on this assumption, we can create trusted entities. A good example of an application whose security could be greatly improved with a threshold solution is a network Certification Authority, a trusted entity that certifies that a given public key corresponds to a given user. If we trust one server to perform this operation,then it is possible that as a result of just one break-in, no certificate can any longer be trusted.Thus it is a good idea to distribute the functionality of the certification authority between many servers, so that an adversary would need to corrupt half of them before he can forge a certificate on some public key.

Goals: In the threshold setting, we would like toimplement, via efficient protocols, the mostsecure cryptosystems and signature schemes. Wewould also like to make our protocols secure inthe strongest possible model of faults. Thefollowing are some of the various considerationswe make when modeling computer faults

A. The Size Of The Threshold

What fractionof the servers can be corrupted by theadversary without any harm to theservice (e.g. signature or decryption)that these servers implement?

B. Efficiency Considerations

How muchcommunication, storage, andcomputation dothese fault-tolerantprotocols require?

C. Model Of Communication

How realisticare the requirements we place on it? Dowe require synchronous or partiallysynchronous communication,authenticated broadcast and secure linksbetween servers?

D. Type Of Adversary We Tolerate

Howdoes the adversary choose whichplayers to corrupt? Can a serversecurely erase its local data so that itcannot be retrieved by the adversaryonce the server is infiltrated?

III. PAILLIER CRYPTOSYSTEM ALGORITHM

A. Key Generation

i. Choose two large prime numbers p and q randomly and independently of each other such that $\gcd(pq, (p-1)(q-1)) = 1$. This property is assured if both primes are of equivalent length, i.e., $p, q \in 1||\{0,1\}^{s-1}$ for security parameter s.

Compute $n = pq$ and $\lambda = \mathrm{lcm}(p-1, q-1)$

ii. Select random integer g where $g \in \mathbb{Z}^*_{n^2}$

iii. Ensure n divides the order of g bychecking the existence of the followingmodular multiplicative inverse $\mu = (L(g^\lambda \mod n^2))^{-1} \mod n$

Where function L is defined as $L(u) = \frac{u-1}{n}$.

Note that the notation $\frac{a}{b}$ does not denote the modular multiplication of a times the modular multiplicative inverse of *b* but rather the quotient of a divided by b,i.e., the largest integer value $v \geq 0$ to satisfy the relation $a \geq vb$.

a. *The Public (Encryption) Key Is (N, G).*

b. *The private (decryption) key is (λ, μ).*

If using p, q of equivalent length, a simpler variant of the above key generation steps would be to set $g = n+1, \lambda = \varphi(n)$, And $\mu = \varphi(n)^{-1} \mod n$ where $\varphi(n) = (p-1)(q-1)$.

IV. ENCRYPTION

i. Let m be a message to be encryptedwhere $m \in \mathbb{Z}_n$

ii. Select random r where $r \in \mathbb{Z}^*_n$

iii. Compute cipher text as $c = g^m \cdot r^n \mod n^2$

V. DECRYPTION

i. Cipher text $c \in \mathbb{Z}^*_{n^2}$

ii. Compute message $m = L(c^\lambda \mod n^2) \cdot \mu \mod n$

As the original paper points out, decryption is"essentially one exponentiation modulo n^2."

VI. HOMOMORPHIC PROPERTIES

A notable feature ofthe Paillier cryptosystem isits homomorphic properties. As the encryptionfunction is additively homomorphic, thefollowing identities can be described:

A. Homomorphic Addition Of Plaintexts

The product of two cipher texts will decrypt to the sum of their corresponding plaintexts,

$$D(E(m_1,r_1)\cdot E(m_2,r_2) \mod n^2) = m_1 + m_2 \mod n.$$

The product of a cipher text with A plaintext raising g will decrypt to the sum of the corresponding plaintexts,

$$D(E(m_1,r_1)\cdot g^{m_2} \mod n^2) = m_1 + m_2 \mod n.$$

B. Homomorphic Multiplication OfPlaintexts

An encrypted plaintext raised to the power of another plaintext will decrypt to the product of the two plaintexts,

$$D(E(m_1,r_1)^{m_2} \mod n^2) = m_1 m_2 \mod n,$$
$$D(E(m_2,r_2)^{m_1} \mod n^2) = m_1 m_2 \mod n.$$

More generally, an encrypted plaintext raised to a constant k will decrypt to the product of the plaintext and the constant,

$$D(E(m_1,r_1)^k \mod n^2) = k m_1 \mod n.$$

However, given the Paillier encryptions of two messages there is no known way to compute an encryption of the product of these messages without knowing the private key.

VII. Semantic Security

The original cryptosystem as shown above doesprovide semantic security against chosenplaintextattacks (IND-CPA). The ability tosuccessfully distinguish the challenge cipher textessentially amounts to the ability to decidecomposite residuosity. The so-called decisionalcomposite residuosity assumption (DCRA) isbelieved to be intractable.Because of the aforementioned homomorphicproperties however, the system is malleable, andtherefore does not enjoy the highest echelon ofsemantic security that protects against adaptivechosen-cipher text attacks (IND-CCA2). Usuallyin cryptography the notion of malleability is notseen as an "advantage," but under certainapplications such as secure electronic voting andthreshold cryptosystems, this property mayindeed be necessary.Paillier and Point cheval however went on topropose an improved cryptosystem thatincorporates the combined hashing of message mwith random r. Similar in intent to the Cramer-Shoup cryptosystem, the hashing prevents anattacker, given only c, from being able to changem in a meaningful way. Through this adaptationthe improved scheme can be shown to be INDCCA2secure in the random oracle model.

VIII. Applications

A. Electronic voting

Semantic security is not the only consideration.There are situations under which malleabilitymay be desirable. The above homomorphicproperties can be utilized by secure electronicvoting systems. Consider a simple binary ("for" or "against") vote. Let m voters cast a vote of either *1* (for) or *0* (against). Each voter encryptstheir choice before casting their vote. Theelection official takes the product of the *m*encrypted votes and then decrypts the result andobtains the value*n,* which is the sum of all thevotes. The election official then knows that npeople voted for and *m-n* people voted against.The role of the random *r* ensures that twoequivalent votes will encrypt to the same valueonly with negligible likelihood, hence ensuringvoter privacy.

B. Electronic Cash

Another feature named in paper is the notion ofself-blinding. This is the ability to change onecipher text into another without changing thecontent of its decryption. This has application tothe development of electronic cash, an effortoriginally spear-headed by David Chaum.Imagine paying for an item online without thevendor needing to know your credit cardnumber, and hence your identity. The goal inboth electronic cash and electronic voting is toensure the e-coin (likewise e-vote) is valid, whileat the same time not disclosing the identity of theperson with whom it is currently associated.

IX. Conclusion

A new threshold Signing scheme is proposed in this project that when combined with Shared Paillier secret keys generation will leads us to a complete solution for the Threshold Paillier problem. The complete solution has also been implemented successfully in this project.

X. References

1) Adam Barnett and Nigel P.Smart. Mental Poker Revisited. Cryptography and Coding 2003, Springer-Verlag LNCS 2898.
2) M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for Designing e_client protocols. Proc. 1st ACM Conference on Computer and Communications Security, 1993, 62–73, 1993.
3) Benaloh. Secret sharing homomorphisms:keeping shares of a secret. Advances in Cryptography - CRYPTO '86. Lecture notes in Computer Science,vol. 263. Springer- Verlag, New York, LNCS 263.
4) Ben-Or, M. Goldwasser and Wigderson.Completeness theorems for noncryptographicfault-tolerant distributed computation.Proceeding of the 20th Annual ACMSymposium on Theory of Computing.Chicago, 11, May 2-4. ACM,
5) D.Boneh and M. Franklin. E_client generation of shared RSA keys. Advances in Cryptography – CRYPTO '97, Springer-Verlag LNCS 1233, 425–439, 1997.
6) Dan Boneh and Matthew Franklin. E_cientGeneration of shared RSA Keys. J. ACM,48, 702-722, 2001.
7) Boyd. Digital Multisignatures. Cryptographyand Coding 1989. Instituteof Mathematics and its application, IMA. 241–246,Clarendon Press, 1989.