Workshop Proceedings of

ATIS 2010

Melbourne, November 8th, 2010.

First Applications and Techniques in Information Security Workshop

Edited by:
Matthew Warren
ISBN 978 1 74156 146 3

Proceedings of

ATIS 2010

Edited by

Matthew Warren
ISBN 978 1 74156 146 3

Workshop sponsored by:

Australian Mathematical Science Institute.

Published by the School of Information Systems, Deakin University, Melbourne, Victoria, 3125, Australia.

All papers published in the conference proceedings have been blind refereed by at least two of the ATIS 2010 **Organising** committee.

Welcome

The ATIS 2010 workshop is the first workshop and is being held at Deakin University, Burwood Campus, Melbourne, Australia. This workshop looks at the continued development of Information Security and related technologies taking into account the issues of the 21st Century.

Members of the workshop organising committee accepted each paper in the proceedings after a careful review; this took the form of a **blind review** by at least **two** members of the workshop organising committee. The papers were subsequently reviewed and developed where appropriate; taking into accounts the comments of the reviewers. The aim of this conference is to further the work already achieved within Australia and bring together researchers in the field to discuss the latest issues and their implications within the 21st Century.

We commend the authors for their hard work and sharing their results, and the reviewers of the workshop for producing an excellent program.

ATIS 2010 Organising Committee

Contents

Paper 1: The padding scheme for RSA signatures

By L.M. Batten, Deakin University, Australia and C. Wolf, Ruhr-University Bochum, Germany.

The Padding Scheme for RSA Signatures

Lynn Margaret Batten [1]
Deakin University, Australia
lmbatten@deakin.edu.au

Christopher Wolf [2]
Ruhr-University Bochum, Germany
cbw@hgi.rub.de

This paper is dedicated to the memory of Jim Totten, a fellow student of number theory and good friend.

Abstract:
The RSA scheme is used to sign messages; however, in order to avoid forgeries, a message can be padded with a fixed string of data P. De Jonge and Chaum showed in 1985 that forgeries can be constructed if the size of P (measured in bytes) is less than the size of $N/3$, where N is the RSA modulus. Girault and Misarsky then showed in 1997 that forgeries can be constructed if the size of P is less than the size of $N/2$. In 2001, Brier, Clavier, Coron and Naccache showed that forgeries can still be constructed when the size of P is less than two thirds the size of N. In this paper, we demonstrate that this padding scheme is always insecure; however, the complexity of actually finding a forgery is $O(N)$. We then focus specifically on the next unsettled case, where P is less than $3/4$ the size of N and show that finding a forgery is equivalent to solving a set of diophantine equations. While we are not able to solve these equations, this work may lead to a break-through by means of algebraic number theory techniques.

Keywords
RSA, Cryptography, Signing, Diophantine Equation

1 RSA Fixed Padding Signature Schemes

RSA was invented in 1977 by Rivest, Shamir and Adleman [7]. It is still the most widely implemented public key scheme, and is used to provide privacy

[1]Supported by a Discovery Grant of the Australian Research Council. The author wishes to thank COSIC/ESAT at KULeuven for their hospitality, where she was a Visiting Professor.
[2]Partially supported by Concerted Research Action GOA-MEFISTO-666 of the Flemish Government (Belgium)

and authentication for digital data. Signing messages is an RSA application embedded in several standards, such as PKCS#1, v2.0 and v2.1 [8].

To sign a message m in an RSA scheme, the signer exponentiates with her private key d to get m^d and computes this modulo N, the RSA fixed modulus. To retrieve m, a receiver applies the signer's public key e to obtain $(m^d)^e \equiv m(\text{mod } N)$. The fact that applying e releases m to the receiver verifies that the owner of the key pair (d, e) was in fact the sender, as no-one else knows d.

There are many ways to attack such a signature scheme. For example, suppose Oscar is able to convince Alice to send him two different messages, m_1 and m_2, signed with Alice's private key d. Then Oscar has $(m_1)^d(m_2)^d = (m_1m_2)^d$ and can send the new message m_1m_2 to a third party, signed with Alice's key, and pretend it came from Alice. The usual way of dealing with such an attack is to allow only a certain set of *legitimate* messages (mod N) to be accepted.

A *padding scheme* fixes the set of allowed or legitimate messages modulo N to be only those values between 0 and N which have an affine form $a + wm$ for fixed, known a and w modulo N. As an example, let $N = 91$, $w = 6$ and $a = 1$. Then m can be chosen from 0 to $\lfloor 91/6 \rfloor = 15$ producing legitimate messages 1, 7, 13, 19, 25, 31, 37, 43, 49, 55, 61, 67, 73, 79, 85.

Since w^{-1} exists modulo N with very high probability (recall that in general N is a product of two very large primes), we can rewrite $a + wm$ more simply as $P + m(\text{mod } N)$ where $P \equiv aw^{-1}$ is fixed and m is bounded by the size of P. Thus, a *forgery* is a value $(P + m)^d(\text{mod } N)$ where P is fixed and m is a false message injected by an attacker, but the form and signature d appear to be legitimate.

De Jonge and Chaum [3] in Crypto'85 were the first to show that the size of P in bytes needs to be at least one third the size of N as otherwise, a forgery could be easily constructed. In 1997, Girault and Misarsky [4] were able to show that the scheme was still insecure if the size of P is less than half the size of N, again by directly constructing forgeries. Then in [2], Brier, Clavier, Coron and Naccache extended this to two thirds. In 2002, some additional forgery constructions appeared in this last case by Lenstra and Shparlinski [6]. The next case, where P is less than three quarters the size of N, remains to be solved, in terms of a direct construction.

Some recent papers have again considered this problem. Joux, Naccache and Thomé [5] use number field sieving techniques to improve the complexity of finding forgeries in the general case. More precisely, they show that computing $r'th$ roots modulo N is easier than factoring N using current methods, given access to an oracle outputting roots of the form $x + c$, for fixed c. Oracle methods are again employed in a related paper [1] which describes two new attacks

on the now defunct PKCS #1 v1.5.

In section three we prove the general result that, no matter what the size of P, a forgery is always possible within $O(N)$ computations. This in itself is often not enough to render a cryptographic scheme useless. If it is computationally infeasible to generate a forgery, then the scheme may still be usable. In section four, we illustrate this by demonstrating explicitly how construction of a forgery is equivalent to solving a dependent system of diophantine equations for the case where P is less than three quarters the size of N.

The authors wish to thank David Naccache for directing us to the problem, and the referees for their comments.

2 Forgeries

Rephrasing the ideas of Section 1, Oscar would attempt to gather a number of legitimate messages signed by Alice with her private key d : $(P+x_1)^d$, $(P+y_1)^d$ etc. where P is fixed and public, and can combine these as products or quotients to obtain

$$\frac{\prod_{i=1}^{s}(P+x_i)^d}{\prod_{i=1}^{t}(P+y_i)^d} \equiv (P+m)^d (\text{mod } N)$$

for some new message m without knowing d. He will then claim that $P+m$ came from Alice. However, this is only possible if m is in the correct range.

The Girault, Misarsky result [3] indicates that the size of P must be at least one half the size of N in bytes. In terms of comparative size of the numbers, this translates into $P > \sqrt{N}$. Since $P+m$ is a value less than N, we conclude that $m < \sqrt{N}$. The Brier, Clavier, Coron, Naccache result translates into $P > (\sqrt[3]{N})^2$ and so $m < \sqrt[3]{N}$.

In the next section, we show that for all $r \geq 2$, forgeries exist with $1 \leq m \leq \lceil \sqrt[r]{N} \rceil$. However, we do not actually construct them.

3 Forgeries Are Always Possible

As promised, we show in this section that, no matter what the size of the padding P, a forgery always exists in a fixed-pattern padding scheme. The proof is based on the pigeon-hole principle: if all values we generate are distinct, then we have too many.

THEOREM *Let P be a fixed padding for an RSA fixed-padding signature scheme with modulus N. Let r be any integer greater than or equal to two, sat-*

isfying $r - 1 < \lceil \sqrt[r]{N} \rceil$. *Then there is a message* m, $1 \leq m \leq \lceil \sqrt[r]{N} \rceil$, *such that the signature of* $P + m$ *can be forged.*

Proof. Consider the equation

$$P + x_0 \equiv \prod_{i=1}^{s} (P + x_i) \pmod{N} \tag{1}$$

where $s \geq 2, 1 \leq x_i \leq \lceil \sqrt[r]{N} \rceil$ for all $0 \leq i \leq s$, and all x_i, $1 \leq i \leq s$, are fixed and distinct. A value for x_0 in the range $[1, \lceil \sqrt[r]{N} \rceil]$ provides a forgery, either using $P + x_0$ as the forged message, or, if x_0 equals some x_i, $1 \leq i \leq s$, using a factor in the right-hand side as the forged message. (Note that $(P + x_i)^{-1}$ exists with high probability as noted earlier; in fact, only $p + q$ values are not invertible, where $N = pq$.) Clearly, $s \leq \lceil \sqrt[r]{N} \rceil$.

The plan of attack in the proof is to demonstrate that as the x_i range over their interval, then either a number of values of

$$F \equiv \prod_{i=1}^{s} (P + x_i) - P \pmod{N} \tag{2}$$

lie in the range $[1, \lceil \sqrt[r]{N} \rceil]$, giving us a forgery, or, we obtain a contradiction.

Consider two representations of the right-hand side of (2) which are equal

$$\prod_{\substack{x_i \in X \subseteq S \\ 2 \leq |X|}} (P + x_i) - P \equiv \prod_{\substack{y_i \in Y \subseteq S \\ 2 \leq |Y|}} (P + y_i) - P \tag{3}$$

where $S = \{1.2. \ldots s\}$ and whre some x_i is not equal to any y_i.

Equation (3) then results in a forgery as described in Section 2.

We may therefore assume that all values of

$$\prod_{\substack{x_i \in X \subseteq S \\ 2 \leq |X|}} (P + x_i) - P \tag{4}$$

are distinct for all subsets of S not empty and not singletons. There are $2^s - (s + 1)$ such values.

We now show that $2^s - (s + 1) > N - \lceil \sqrt[r]{N} \rceil$ if we choose s such that $s = \log_2(N)$. This will generate a contradiction, since some value of (4) must be in the range $[1, \lceil \sqrt[r]{N} \rceil]$.

If $2^s - (s+1) > N - \lceil \sqrt[r]{N} \rceil$ then certainly, $2^s \geq N - \lceil \sqrt[r]{N} \rceil + 4$ and $s \geq \log_2(N - \lceil \sqrt[r]{N} \rceil + 4)$. Thus, if we choose $\lceil \sqrt[r]{N} \rceil \geq s \geq \log_2(N)$, the strict inequality above is sastisfied. □

COROLLARY *The complexity of finding a forgery using the method of the proof of the Theorem is* $O(N - \lceil \sqrt[r]{N} \rceil) = O(N)$.

Proof. since we calculate (at most) $2^s - (s+1)$ products, as shown in the proof s is about $\log_2(N - \lceil \sqrt[r]{N} \rceil + 4)$. Thus 2^s is of order $O(N - \lceil \sqrt[r]{N} \rceil) = O(N)$. □

As r grows, the complexity approaches N rapidly which may explain why resolving the 3/4 case has proved considerably more difficult than that for 1/2 and 2/3.

EXAMPLE For $N = 1034273, r = 4, \lceil \sqrt[4]{1034273} \rceil = 36$, we have $2^{23} = 8388608$ and $2^{24} = 167772164$. So $2^s > 1034237 + (s+1)$ if $s = 24$. Thus $s = 24$ suffices to ensure a forgery in this case.

4 Constructing Forgeries

While knowing it is possible to construct forgeries is worthwhile in itself, if it is too difficult, or takes too long, to actually construct a forgery, the scheme may still be used with some sense of security. In this section, we reduce the problem of forgery construction to that of solving a dependent system of diophantine equations in the case where the size of P is less than three quarters the size of N. However, in this case, we have no general method of solving the system and leave this as an open problem.

The diophantine equations we are after are produced from the quotient equations in the previous section. A forgery results in an equality of two products simply by cross-multiplying. The number of terms on each side can be equalized simply by adding sufficient terms of the form $P + x_i = 1$.

The papers dealing with the cases one third, one half and two thirds derive their forgeries from such equations. Here we illustrate the situation for the next case, three quarters. We rewrite a message m as a sum or difference $x + y$ etc.

CASE 3/4
Consider $(P + x + y)(P + z + w)(P + v + s) \equiv (P + x - y)(P + z - w)(P + v - s) \pmod{N}$ which we want to solve for $0 < |x + y|, |x - y|, |z + w|, |z - w|, |v + s|, |v - s| < N^{1/4}$. This implies $0 < |x|, |y|, |z|, |w|, |v|, |s| < N^{1/4}$.

Expanding and multiplying by $2^{-1} \bmod N$ (N is odd), we obtain

$$P^2(y+w+s)+P(x(w+s)+z(y+s)+v(y+w))+$$
$$xzs+xwv+yzv+yws \equiv 0 \ (mod\ N) \tag{4}$$

Let $P^2 \equiv Q$, $0 < Q < N$.

By the extended Euclidean algorithm (see reference [4]), there exist t_Q and r_Q such that

$$t_Q Q \equiv r_Q \ (mod\ N),\ |t_Q| < N^{1/4}, 0 < r_Q < 2N^{3/4}.$$

And there exist t_P and r_P such that

$$t_P P \equiv r_P \ (mod\ N),\ |t_P| < N^{1/2}, 0 < r_P < 2N^{1/2}.$$

So $t_Q Q + t_P P \equiv r_Q + r_P \ (mod\ N)$.

Thus (4) becomes, for known r_Q and r_p

$$r_Q + r_P + x(zs+wv) + y(zv+ws) \equiv 0 \ (mod\ N) \tag{4}'$$

We want to obtain y, w, s, x, z and v such that

$$t_Q = y+w+s, \tag{5}$$

$$t_P = x(w+s)+z(y+s)+v(y+w) \tag{6}$$

and such that (4)' holds. Since $r_Q + r_P$ is a known quantity, we can re-write the equation (4)' with constraints as:

determine s, v, w, x, y and z such that

$$x(zs+wv) + y(zv+ws) \equiv A \ (mod\ N) \tag{4}^*$$

$$t_Q = y+w+s, \tag{5}$$

$$t_P = x(w+s)+z(y+s)+v(y+w) \tag{6}$$

where A, t_P and t_Q are known quantities, A is less than N, $0 < |x|$, $|y|$, $|z|$, $|w|$, $|v|$, $|s| < N^{1/4}$, $|t_Q| < N^{1/4}$ and $|t_P| < N^{1/2}$.

Similar systems of equations can be developed for each value of r from the preceding section, but clearly, as r increases, so does the complexity of the equations. We do not know how to solve these equations but hope to inspire these working in the field of diophantine equations to tackle them.

References

[1] A. Bauer, J. - S. Coron, D. Naccache, M. Tibouchi and D. Vergnaud, On the broadcast and validity-checking security of PKCS#1 v1.5 encryption. Proceedings of ACNS 2010, LNCS 6123, pp. 1-18.

[2] E. Brier, C. Clavier, J.-S. Coron and D. Naccache, Cryptanalysis of RSA signatures with fixed-pattern padding. Proceedings of Crypto'01, LNCS vol. 2139, Springer-Verlag 2001, pp. 433-439.

[3] W. De Jonge and D. Chaum, Attacks on some RSA signatures. Proceedings of Crypto'85, LNCS vol. 218, Springer-Verlag 1986, pp. 18-27.

[4] M. Girault and J.-F. Misarsky, Selective forgery of RSA signatures using redundancy, Proceedings of Eurocrypt'97, LNCS vol. 1233, Springer-Verlag 1997, pp. 495-507.

[5] A. Joux, D. Naccache and E. Thomé, When $e'th$ roots become easier than factoring. Proceedings of Asiacrypt 2007, LNCS 4833, pp. 13-28.

[6] A.K. Lenstra and I.E. Shparlinski, Selective forgery of RSA signatures with fixed-pattern padding, In PKC 2002, LNCS vol. 2274, Springer-Verlag 2002, pp. 228-236.

[7] R. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signatures and public key cryptosystems, CACM 21, 1978.

[8] RSA Laboratories, PKCS# 1: RSA cryptography specifications, version 2.0, September 1998, version 2.1, June 2002

Paper 2: Information theoretically secure data transmission in networks

By Siu-Wai Ho, Terence H. Chan, Alex Grant, University of South Australia, Australia.

Information Theoretically Secure Networks

Terence Chan, Siu-Wai Ho and Alex Grant
Institute for Telecommunications Research
University of South Australia

Abstract

There are three main approaches to security: computational, physical, and information-theoretic. Each approach has its own advantages and disadvantages. They are not conflicting but instead complementing each other. In this paper, we will consider the information theoretically secure data transmission problem. First we will consider the set of all admissible rate-capacity tuples for secure data transmission in networks. By using representable functions, we explicitly characterise the set when linear codes are used. Then, we will consider the case when the sources are biased (and are of small block length). We show that there is a tradeoff between the size of secret key and the expected codeword length. Two schemes are also proposed which respectively minimise the key size and the expected codeword length. If a countably-infinite-valued source is compressed and then encrypted by one-time pad, no scheme can achieve perfect secrecy and finite number of channel use simultaneously.

Keywords

Entropy function, Huffman code, one-time pad, secure network coding,

1. INTRODUCTION

There is no doubt that communications network security is vital to modern digital infrastructure [17]. To protect users from malicious attacks, it is critical that data should remain confidential such that they will not be leaked to illegitimate parties via malicious or accidental eavesdropping. Furthermore, a network must be robust so that communications will not be interrupted due to link or node failures and unintentional interference. Reliable and authenticated communication is also absolutely necessary, especially in the presence of malicious tampering (where an adversary intelligently modifies transmitted signals), spoofing (adversary attempts to impersonate a legitimate user) and jamming (adversary injects random noise to disrupt communication).

There are three approaches to security: computational, physical, and information-theoretic. Public key cryptographic systems such as the Rivest-Shamir-Adleman (RSA) system [18] is one example of computationally secure system. In the system, security will be compromised if the adversary can factorise an integer easily. If we assume that the eavesdropper has unbounded computational power, then the RSA system will not be secure. On the other hand, physical security exploits physical properties to achieve security. The simplest example is the door lock where an adversary will need a physical key to open the door. Quantum key distribution [2] is another fancy example which exploits physical properties such as quantum superposition or entanglement [9] to detect any eavesdropping [21].

Information-theoretic approach, pioneered by Shannon [20], determines the maximum transmission rate such that it is impossible for the adversary to break the system, regardless of the computational resources available, or the physical embodiment of the data. Security is achieved by a disparity of information resources available to legitimate and illegitimate users. One example of an information secure system is the one-time pad [20], which requires two communications channels: one for the encrypted message, and another for the key. The usual assumption is that the eavesdropper has no knowledge about what the secret key is. By this lack of knowledge, the eavesdropper will remain ignorant to the source message even if it can eavesdrop the transmitted data packet.

There is much existing work on information theoretic approach to security, including secure data transmission [23], [3], [7], robust data transmission [4], [5], key agreement [14], [15], secret sharing [1], [19], message authentication [16], [22], biometric identification [12] and many others. The focus of this paper is on information-theoretic approaches to secure data transmission. However, our goal is *not* to replace computational or physical security. Rather, we believe that it complements existing approaches, amplifying security in network environments.

In this paper, a tutorial on the setup of secure network coding will be given in Section 2. Base on the settings, the problem of secure data transmission in networks will be studied from an information theoretic approach in Section 3. The admissible rate-capacity region for secure data transmission in networks will be characterized. In network applications, latency is always one of the most important issues. The asymptotic analysis used in the first part of this paper usually ignores the latency issue. Therefore, in Section 4, we will consider the non-asymptotic scenarios. We will see that finding an efficient algorithm/coding scheme to achieve perfect security is more complicated when compared with one-time pad [20]. If we want to achieve perfect secrecy, there is indeed a fundamental tradeoff between the key rate and the number of channel uses. If a countably-infinite-valued source is compressed and then encrypted by one-time pad, no scheme can achieve perfect secrecy and finite number of channel use simultaneously.

2. SECURE DATA TRANSMISSION IN NETWORKS

To illustrate the ideas of secure data transmission, consider a network depicted in Figure 1. In this figure, the source node and the intermediate network nodes are respectively depicted by a filled circle and open circles. Alice is at the source node who aims to transmit a secret message m to her friends Bob and Beth who are located respectively at the two sink nodes (illustrated by open squares) in the network. Assume that the eavesdropper Eve is in the middle of the network who can choose to listen/eavesdrop any single link. The question is how secure data transmission can be made possible?

One possibility is by using the following secure network code. Assume that the secret message m is drawn from the finite field $GF(3)$. First, Alice can pick a key k randomly also from $GF(3)$. Then, the packet being transmitted on each link is depicted in the figure.

It is not hard to see that Eve can obtain no knowledge about the message m by eavesdropping any link. For Bob at the upper sink node, he can receive $m+k$ and $m+2k$ from two of its incoming links. Therefore, Bob can compute the secret message m back by solving a system of two linear equations. Similar is true for Beth at the lower sink node. Therefore, the secure network code does achieve perfect secrecy.

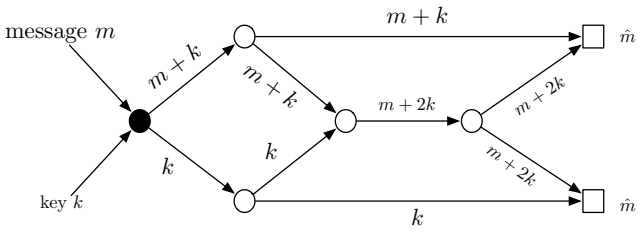

Fig. 1. A secure network code.

In the following, we will extend the example in a general setting where there are more than one source. First, we will model a communication network G by a directed acyclic graph $(\mathcal{V}, \mathcal{E})$. Here, the set of nodes $\mathcal{V}$ and the set of directed edges $\mathcal{E}$ respectively model the set of communication nodes and error free transmission links. A link $e \in \mathcal{E}$ is defined by a tuple $(tail(e), head(e))$ where information can be sent from the node $tail(e)$ to the node $head(e)$ without errors.

Definition 1 (Connection constraint): For a given communication network G, a *connection constraint* M is a tuple $(\mathcal{S}, O, D, \mathcal{W})$ such that

- $\mathcal{S}$ is the set of independent sources;
- $O : \mathcal{S} \mapsto \mathcal{V}$ is the source location mapping. Specifically, $O(s)$ is the node where the s^{th} source is available;
- $D : \mathcal{S} \mapsto 2^{\mathcal{V}}$ is the sink location mapping such that $D(s)$ are the set of sink nodes to which the s^{th} source needs to be transmitted;
- $\mathcal{W} \triangleq \{(\mathcal{A}_r, \mathcal{B}_r), r \in \mathcal{R}\}$ is the *wiretapping pattern* of the network. We assume that there are $|\mathcal{R}|$ eavesdropper in the network, such that the r^{th} eavesdropper can observe message transmitted along links in the set $\mathcal{B}_r \subseteq \mathcal{E}$ and aims to reconstruct the set of sources indexed by $\mathcal{A}_r \subseteq \mathcal{S}$.

The objective of secure data communications is to *transmit data across a network such that the eavesdroppers in the network cannot obtain any information about the set of sources they are interested in*.

It is worth pointing out that secure network coding includes secret sharing as a special case. In secret sharing [1], a secret is shared among a set of users $\mathcal{N}$ where each user i holds a piece of secret. The main objective is to ensure that only a specified

legitimate subgroups of users (indexed by a subset $\mathcal{A}$ of $\mathcal{N}$) are allowed to decode the secret. For all other illegitimate subgroups of users, they will receive no information about the secret. Let Ω be the collection of all legitimate subsets. The set Ω is called the access structure of the problem.

We will now reformulate a secret sharing problem as a secure network coding problem (G, M). In this secure network coding problem, there is only one source (i.e., the secret) (available at the source node u^*). There are $|\mathcal{N}|$ intermediate nodes, each of which represents a user. The transmitted message that an intermediate node (or a user) received from the source corresponds to a piece of secret it holds. There are $|\Omega|$'s sink nodes indexed by $\{v_\alpha, \alpha \in \Omega\}$. The sink node v_α is connected to nodes (or users) $i \in \alpha$ and aims to reconstruct the secret. We also assume that for each $\beta \notin \Omega$, it is associated with an eavesdropper who can wiretap the set of edges $\{e_i, i \in \beta\}$. So the secrecy constraint implies that all illegitimate subgroups of users have no information about what the secret is. Mathematically, the secret sharing problem is formulated as follows.

1) $\mathsf{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{u^*\} \cup \mathcal{N} \cup \{v_\alpha, \alpha \in \Omega\}$ and $\mathcal{E} = \{e_i, f_{i,\alpha}, i \in N, \alpha \in \Omega\}$;
2) for any $i \in \mathcal{N}$, $tail(e_i) = u^*$, $head(e_i) = \{i\}$, $tail(f_{i,\alpha}) = i$ and $head(f_{i,\alpha}) = v_\alpha$;
3) $\mathsf{M} = (\mathcal{S}, O, D, \mathcal{W})$ where (i) $\mathcal{S} = \{1\}$, (ii) $O(1) = \{u^*\}$, (iii) $D(1) = \{v_\alpha, \alpha \in \Omega\}$, and (iv) $\mathcal{W} = \{(1, e_i, i \in \beta) : \beta \subseteq \mathcal{N}$ and $\beta \notin \Omega\}$.

By specifying the secret sharing problems as the above secure network coding problems, results obtained in this paper can be translated accordingly to ones for secret sharing problems.

Example 1: Consider (4,2) secret sharing scheme (the access structure is the set of all subsets of legitimate of size at least two). Then the network coding problem associated with the secret sharing scheme is depicted in Figure 2.

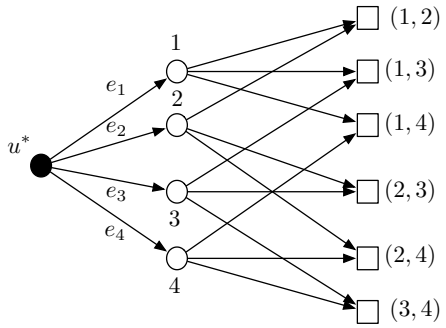

Fig. 2. A (4,2)-threshold secret sharing scheme.

For any network coding problem (G, M), let

$$\text{in}(e) \triangleq \{s \in \mathcal{S} : O(s) = tail(e)\} \cup \{f \in \mathcal{E} : head(f) = tail(e)\} \cup \{tail(e)\} \tag{1}$$

$$\text{in}(u) \triangleq \{s \in \mathcal{S} : O(s) = u\} \cup \{f \in \mathcal{E} : head(f) = u\} \tag{2}$$

for any $e \in \mathcal{E}$ and $u \in \mathcal{V}$. Consider any network coding problem P defined by a network and a connection constraint (G, M). A linear network code Φ (with respect to P) is specified by a set of local encoding matrices $\{M_e, e \in \mathcal{E}\}$ where M_e is a $\sum_{f \in \text{in}(e)} (c_f \times c_e)$ matrix. Specifically, for each $s \in \mathcal{S}$, the s^{th} random source is a length c_s row vector (over the field $GF(q)$), and is denoted by Y_s. Similarly, for each node $v \in \mathcal{V}$, it is associated with a random key Y_v, which is a length c_v row vector. All these vectors are assumed to be independently and uniformly distributed over their sample spaces. For any link $e \in \mathcal{E}$, the message Y_e transmitted on the link e is a length c_e row vector defined recursively by

$$Y_e = [Y_f, f \in \text{in}(e)]\, M_e. \tag{3}$$

The message Y_e is a linear function of the sources, transmitted message and random keys available at the node $tail(e)$. It can be proved recursively that Y_f is a linear function of $[Y_s, s \in \mathcal{S}, Y_u, u \in \mathcal{V}]$ for all $f \in \mathcal{S} \cup \mathcal{V} \cup \mathcal{E}$. In fact, there exists $\sum_{g \in \mathcal{S} \cup \mathcal{V}} c_g \times c_f$ matrices K_f such that $Y_f = [Y_s, s \in \mathcal{S}, Y_u, u \in \mathcal{V}]K_f$. A linear network code can also be specified by these global encoding matrices $\{K_f, f \in \mathcal{S} \cup \mathcal{E} \cup \mathcal{V}\}$.

A linear network code $\Phi = \{K_f, f \in \mathcal{S} \cup \mathcal{E} \cup \mathcal{V}\}$ is called secure and error free if

1) for any $r \in \mathcal{R}$, the random sources $(Y_s, s \in \mathcal{A}_r)$ (which the r^{th} eavesdropper is interested in) is independent of the set of transmitted messages $(Y_e, e \in \mathcal{B}_r)$ (which the eavesdropper can listen to). Consequently, the r^{th} eavesdropper can gain no information about the set of sources that he is interested in.

2) for any $s \in \mathcal{S}$ and $u \in D(s)$, Y_s is a function of $(Y_f, f \in \text{in}(u))$. In other words, the receiver at the sink node u can reconstruct the source vector Y_s (by using only the transmitted messages and sources that are available to it)

Definition 2 (Admissibility): For a secure network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M})$, a tuple $(\lambda, \omega) \triangleq (\lambda(s), s \in \mathcal{S}, \omega(e), e \in \mathcal{E})$ is called *admissible* if there exists a sequence of secure and error free network codes $\Phi^n = \{K_f^n, f \in \mathcal{S} \cup \mathcal{E} \cup \mathcal{V}\}$ and positive normalizing constants δ_n such that for all $e \in \mathcal{E}$ and $s \in \mathcal{S}$,

$$\lim_{n\to\infty} \delta_n r_e^n \leq \omega(e), \tag{4}$$

$$\lim_{n\to\infty} \delta_n r_s^n \geq \lambda(s), \tag{5}$$

where r_f^n is number of columns of the matrix K_f^n.

Let $\mathsf{T}(\mathsf{P})$ be the set of all tuples (λ, ω). For any subset $\mathcal{R}$ of $\mathsf{T}(\mathsf{P})$, we define $\mathsf{CL}(\mathcal{R})$ as the subset of $\mathsf{T}(\mathsf{P})$ containing all tuples (λ, ω) such that there exists a sequence of $(\lambda^n, \omega^n) \in \mathcal{R}$ and positive numbers δ_n satisfying

$$\lim_{n\to\infty} \delta_n \omega^n(e) \leq \omega(e), \tag{6}$$

$$\lim_{n\to\infty} \delta_n \lambda^n(s) \geq \lambda(s). \tag{7}$$

Clearly, if $\mathcal{R}$ is admissible, then $\mathsf{CL}(\mathcal{R})$ is also admissible.

3. CHARACTERISATION OF ADMISSIBLE TUPLES

It is *fundamental* to determine the set of all admissible rate-capacity tuples for arbitrary networks. Unfortunately, [6] proved that solving this general problem is extremely difficult: At least, it is as hard as determining the set of all rank inequalities [10]. In fact, the only way to characterise the set of admissible/achievable tuples is via the use of "representable functions" (whose definitions will be given below). So far, the sets of admissible tuples are determined only in a very limited number of scenarios. One scenario is when there is only one source (i.e., $|\mathcal{S}| = 1$) and subject to no secrecy constraint (i.e., $|\mathcal{R}| = 0$). In this case, the set of achievable tuples is determined by the cut-set bound. If secrecy constraint is additionally imposed, the set of achievable tuples can still be determined if (i) all links have unit capacity and (ii) the eavesdropper is "uniform" in the sense that an eavesdropper can wiretap any t links in the network.

In this section, we will give a complete characterisation for the set of admissible tuples by using representable functions. Define $\mathcal{H}[\mathcal{N}]$ as the $2^{|\mathcal{N}|}$-dimensional Euclidean space whose coordinates are indexed by subsets of $\mathcal{N}$. Thus $h \in \mathcal{H}[\mathcal{N}]$ has coordinates $(h(\alpha), \alpha \subseteq \mathcal{N})$. For any set of vector spaces $\{Y_i, i \in \mathcal{N}\}$, it induces a set function h such that for any subset α of $\mathcal{N}$,

$$h(\alpha) \triangleq \dim\langle Y_i, i \in \alpha\rangle. \tag{8}$$

Here, we define $h(\alpha) = 0$ whenever α is an empty set. We call such induced functions *representable*. Let $\bar{\Upsilon}^*(\mathcal{N})$ be the minimal closed and convex cone containing the set of all representable functions. Clearly, h is non-negative and submodular (i.e., $h(\alpha \cup \beta) + h(\alpha \cap \beta) \leq h(\alpha) + h(\beta)$ for all $\alpha, \beta \subseteq \mathcal{N}$).

For any function h in $\mathcal{H}[\mathcal{N}]$ and disjoint $\alpha, \beta \subseteq \mathcal{N}$, we define

$$h(\alpha|\beta) \triangleq h(\alpha \cup \beta) - h(\beta), \tag{9}$$

$$I_h(\alpha; \beta) \triangleq h(\alpha) + h(\beta) - h(\alpha \cup \beta). \tag{10}$$

Hence, if h is a representable function induced by subspaces $\{Y_i, i \in \mathcal{N}\}$, then $h(\alpha|\beta) = 0$ if and only if $\langle Y_i, i \in \alpha\rangle$ is a subspace of $\langle Y_i, i \in \beta\rangle$. Also, $I_h(Y_\alpha; Y_\beta) = 0$ if and only if the two vector spaces $\langle Y_i, i \in \mathcal{N}\rangle$ and $\langle Y_i, i \in \alpha\rangle$ intersect trivially.

Let $\mathcal{H} \triangleq \mathcal{H}[\mathcal{S} \cup \mathcal{E} \cup \mathcal{V}]$. For any $h \in \mathcal{H}$, we define proj$[h]$ as a rate-capacity tuple in $\mathsf{T}(\mathsf{P})$ such that for all $s \in \mathcal{S}$ and $e \in \mathcal{E}$,

$$\text{proj}[h](s) = h(s) \tag{11}$$

$$\text{proj}[h](e) = h(e). \tag{12}$$

Similarly, for any subset $\mathcal{R}$ of $\mathcal{H}[\mathcal{S} \cup \mathcal{E}]$, we define proj$[\mathcal{R}]$ as $\{\text{proj}[h] : h \in \mathcal{R}\}$.

Given a secure network coding problem $\mathsf{P} = (\mathsf{G}, \mathsf{M})$. Define the following constraint

$$\mathcal{C}_{\rm indep} \triangleq \left\{ h \in \mathcal{H} : h(\mathcal{S}, \mathcal{V}) = \sum_{s \in \mathcal{S}} h(s) + \sum_{u \in \mathcal{V}} h(u) \right\}, \tag{13}$$

$$\mathcal{C}_{\rm netwk} \triangleq \left\{ h \in \mathcal{H} : h\left(s \mid {\rm in}(e), tail(e)\right) = 0, \forall e \in \mathcal{E} \ \right\}, \tag{14}$$

$$\mathcal{C}_{\rm de} \triangleq \left\{ h \in \mathcal{H} : h\left(s \mid {\rm in}(u)\right) = 0, \ \forall s \in \mathcal{S}, u \in D(s) \ \right\}, \tag{15}$$

$$\mathcal{C}_{\rm sec} \triangleq \left\{ h \in \mathcal{H} : I_h(\mathcal{A}_r; \mathcal{B}_r) = 0, \forall r \in \mathcal{R} \right\}. \tag{16}$$

The physical meanings of the above subsets are as follows. Consider a secure and error free network code defined by the global encoding matrices $\{K_f, f \in \mathcal{S} \cup \mathcal{E} \cup \mathcal{V}\}$. Each matrix induces a subspace spanned by the rows of the matrix. These induced set of subspaces further gives rise to a representable function h. It can be directly verified that $h \in \mathcal{C}_{\rm indep}$ because all the sources and the random keys are independent. Similarly, $h \in \mathcal{C}_{\rm netwk}$ because the symbol transmitted on link e is a function of all the source, link symbols and random key available at the tail of e by (3). $h \in \mathcal{C}_{\rm de}$ follows from that Y_s can be reconstructed at the sink node $u \in D(s)$. Finally, the last constraint corresponds to the secrecy constraint where the r^{th} eavesdropper gains no information about the sources $(Y_s, s \in \mathcal{A}_r)$ even after eavesdropping the links $(Y_e, e \in \mathcal{B}_r)$.

Theorem 1 (Admissibiilty): Let $\bar{\Upsilon}^* \triangleq \bar{\Upsilon}^*(\mathcal{S} \cup \mathcal{E} \cup \mathcal{V})$. A rate-capacity tuple (λ, ω) is admissible if and only if

$$(\lambda, \omega) \in \mathsf{CL}(\mathsf{proj}(\bar{\Upsilon}^* \cap \mathcal{C}_{\rm indep} \cap \mathcal{C}_{\rm netwk} \cap \mathcal{C}_{\rm de} \cap \mathcal{C}_{\rm sec})).$$

4. NON-ASYMPTOTIC SCENARIOS

In the previous sections, we impose no constraint on the size of the input message and assume that the input message is uniformly distributed. These assumptions can usually be satisfied if the source can wait and collect a large block of source symbols before encoding and transmission. In this case, the source can always compress the block of source symbols into a sequence of uniformly distributed binary bits. However, in some scenarios, the assumption may not hold. One example is when data transmission is subject to a low latency constraint where data need to be transmitted as soon as possible. In this case, the input of the source is likely to be non-uniform. In the following, we will illustrate how to transmit a biased source securely and efficiently.

First, let us ignore the secrecy constraint and focus on how to transmit a non-uniform source efficiently. Consider the following example in which the probability distribution of the source M is given as

$$\Pr(M = a) = \frac{1}{2}, \ \Pr(M = b) = \frac{1}{4}, \ \Pr(M = c) = \Pr(M = d) = \frac{1}{8}. \tag{17}$$

We can encode the source M as

$$a \mapsto [0, 0], \ b \mapsto [0, 1], \ c \mapsto [1, 0], \ d \mapsto [1, 1]. \tag{18}$$

For example, if the source is $M = b$, then $[0, 1]$ will be transmitted. On average, two bits are needed to transmit one source symbol. Now, consider another encoding scheme.

$$a \mapsto [0], \ b \mapsto [1, 0], \ c \mapsto [1, 1, 0], \ d \mapsto [1, 1, 1]. \tag{19}$$

The average number of bits required to transmit one source symbol is

$$1 \times \frac{1}{2} + 2 \times \frac{1}{4} + 3 \times \frac{2}{8} = 1.75.$$

Therefore, the second variable-length encoding scheme clearly outperforms the first fixed-length encoding scheme. In fact, the reduction in the number of channel uses is due to the following idea: Encode a source symbol of a higher probability into a codeword of a shorter length.

Example 2 (One-time pad [20]): Consider a scenario where Alice aims to transmit n bits $M = [M_1, \ldots, M_n]$ secretly to Bob over an insecure channel. Assume that Alice and Bob share a private key $K = [K_1, \ldots, K_n]$ which is a length-n binary sequence. To prevent an eavesdropper to learn what the secret message M is, Alice can in fact transmit cipher text $X \triangleq M + K$ over the insecure channel instead. It can be proved that (i) an eavesdropper will have no knowledge about M even if he/she knows X, and (ii) Bob can reconstruct M from X and the private key K.

Example 3 (Insecure one-time pad): Suppose that the secret message M has an underlying probability distribution given by (17) and is compressed as specified in (19). Since the maximal length of the compressed codeword is 3, we will assume that Alice and Bob both share a three-bit secret key $[K_1, K_2, K_3]$. In that case, the ciphertext will be given by

$$a \mapsto [0 + K_1],\ b \mapsto [1 + K_1, 0 + K_2],\ c \mapsto [1 + K_1, 1 + K_2, 0 + K_3],\ d \mapsto [1 + K_1, 1 + K_2, 1 + K_3]. \tag{20}$$

Note that the length of the ciphertext X is not a constant. In fact, if the eavesdropper knows that the length of X is two, then he/she can determine that the secret message M must be b. Clearly, applying one-time pad directly on top of a variable length code will not be secure. In the following, we will propose two schemes – one minimizes the size of key needed to be shared by Alice and Bob, and the other one minimizes the number of channel uses.

Remark: In general, all the data compression algorithms are based on the principle that a frequent symbol is encoded by a shorter codeword. Therefore, the simple approach by combining data compression with one-time pad does not provide perfect secrecy.

Algorithm A: (codebook construction)

1) Construct for the secret message M a prefix-free variable length code which minimises the expected number of channel uses[1]. Let n be the maximal codeword length.
2) For each message m, let the associated codeword be $c(m)$ whose length (denoted by $|c(m)|$) will be no longer than n.
3) For each m, let $d(m)$ be a codeword with length n obtained by concatenating $c(m)$ with $n - |c(m)|$'s padding zeros.

Algorithm A: (encoder)

1) To transmit the secret message m, Alice will first construct $n - |c(m)|$'s random bits denoted by $[R_1, \ldots R_{n-|c(m)|}]$. These random bits are unknown to Bob.
2) Construct a length n binary key $\hat{K}$ defined by $[K_1 K_2 \ldots K_{|c(m)|} R_1 \ldots R_{n-|c(m)|}]$ where $[K_1 K_2 \ldots K_{|c(m)|}]$ is part of the private key shared by Alice and Bob. Note that the required key size for message m is $c(m)$.
3) The ciphertext $X \triangleq [X_1, \ldots, X_n]$ is then obtained by bitwise XOR $d(m)$ and the key $[K_1 K_2 \ldots K_{|c(m)|} R_1 \ldots R_{n-|c(m)|}]$ together.

Algorithm A: (decoder)

Decoding rule is very simple. Pick m such that

$$[X_1, \ldots, X_{c(m)}] = c(m) + [K_1, \ldots, K_{|c(m)|}].$$

It can be verified easily that decoding is error free because the chosen variable length code is a prefix free code.

Proposition 1 (Key minization): Among all the secure data transmission schemes, Algorithm A "almost minimizes" the expected key size.

Proof: By [20], for any secure data transmission scheme, $H(K) \geq H(M)$. Now, if we pick the optimal variable length code for M, the expected length $E[c(M)]$ (which is also the expected required key size) satisfies

$$H(K) = E[c(M)] \leq H(M) + 1.$$

Hence, the required key size is no more than one bit greater than the optimal key size. ■

While algorithm A "optimizes" the required key size, it does not necessarily minimize the number of channel uses. In fact, in the following, we will propose another algorithm which can minimize the number of channel uses.

Algorithm B: (encoder)

1) Let $n = \lceil \log |M| \rceil$.
2) It is easy to compress M into a fixed length n binary code. Like in algorithm A, let $c(m)$ be the length n codeword
3) To transmit the secret message m, Alice will construct the ciphertext $X \triangleq [X_1, \ldots, X_n]$ by bitwise XOR $c(m)$ with a private key $K = [K_1, \ldots, K_n]$ shared by both Alice and Bob.

[1]The minimal length prefix free code can be constructed by Huffman procedure.

Algorithm B: (decoder)

Decoding is very simple. Bob simply needs to bitwise XOR X with the secret key K to obtain $c(M)$. Then m can also be decoded as c is a one-to-one mapping.

Before we continue, we define a specific type of secure data transmission scheme called Compressed and One-time-pad (C-OTP) system. Its block diagram is specified as in Figure 3.

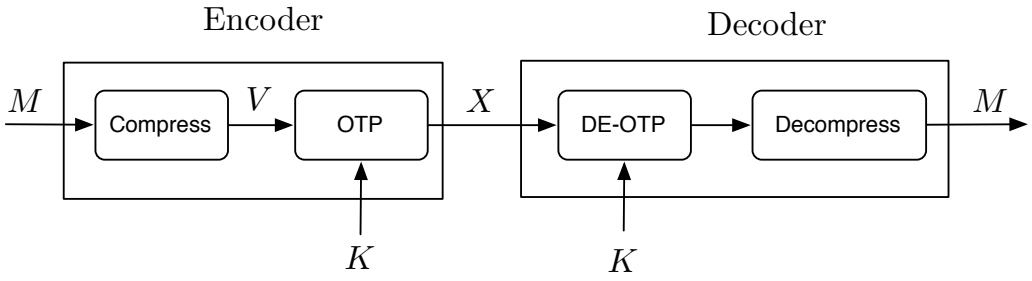

Fig. 3. A C-OTP system

The encoding procedure in a C-OTP system is described as follows. For any input message m, it will generate a compressed sequence $V = (V_1, \ldots, V_{\ell(V)})$ where $\ell(V)$ is the length of V. Note that in this compression stage, the secret key $K = [K_1, \ldots, K_n]$ shared by Alice and Bob will not be used. Furthermore, the length of V is a random variable independent of K. The second one-time pad stage is very straightforward. Specifically, the cipher text X is $[X_1, \ldots, X_{\ell(V)}]$ defined as

$$X_i = V_i + K_i$$

for $i \in \ell(V)$.

Clearly, the secure data transmission system derived from algorithm B is an example of C-OTP system. The following proposition proves that the proposed algorithm B minimizes the amount of channel uses, when compared to all other C-OTP systems.

Proposition 2 (Minimizing Channel Uses): Among all the C-OTP systems, Algorithm B minimizes the number of channel uses.

Proof: Suppose the message M can take values in the set $\{1, ..., |M|\}$ with $P_M(m) > 0$ for $m = 1, \ldots, |M|$ where $|M|$ denotes the size of the support of random variable M. Let V be the output after the compression stage.

Since K is not used in the compression stage, (M, V) is independent of K. Furthermore, in order for the decoder to decode, it is required $H(M|V, K) = 0$. This implies that $H(M|V) = 0$, or equivalently, M is a function of V. Also, as $\ell(X) = \ell(V)$, the eavesdropper will also know $\ell(V)$.

Now, suppose to the contrary that $k < \lceil \log |M| \rceil$ and that $\Pr(\ell(V) = k) > 0$. Note that when $\ell(V) = k$, V can take at most 2^k's values. As $H(M|V) = 0$ (i.e., M is a function of V), the eavesdropper will know that M can take at most 2^k's values.

On the other hand, since $2^k < |M|$, the probability distribution $P_M(M = m)$ cannot be the same as $\Pr(M = m|\ell(V) = k)$. In other words, M and $\ell(V)$ cannot be independent. The secrecy constraint implies that $\Pr(\ell(V) < \lceil \log |M| \rceil) = 0$. Therefore, the expected minimum number of channel use is at least $\lceil \log |M| \rceil$. Clearly, this is achievable by algorithm B and hence, the proposition follows. ■

Remark: (a) If the source takes values from a countably infinite alphabet, i.e., $M = \infty$, then no C-OTP system can simultaneously achieve finite number of channel use and perfect secrecy. (b) From the two algorithms, it is clear that there is a tradeoff between the key size and the number of channel uses. In Figure 4, we illustrate the tradeoff for the case when the secret message M is distributed as in (17).

Instead of perfect secrecy, it is also important to consider other reliability criteria [11]. Again, suppose the source generates a sequence of secret messages M. We may consider the number of errors in adversary's estimate $\hat{M}$ provided that the adversary can have infinite computational power. We may require that the number of errors is equal to the case that the adversary makes only blind guess according to $\Pr(M = m)$. This requirement is weaker than $I(M; X) = 0$. In this case, it is possible to decide an algorithm requires less channel use or less key length. Again, this direction will be further explored in [13].

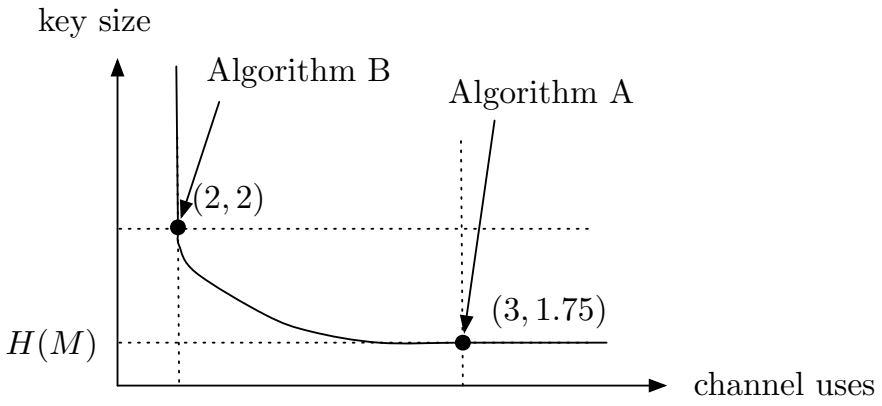

Fig. 4. Tradeoff between key size and number of channel uses

5. CONCLUSION

This paper considers an information-theoretic approach secure data transmission problem. First, we consider the problem where it is subject to no latency constraint. In other words, the source can collect a large block of source data before transmission. In this case, we will assume that the source is in fact uniformly distributed. Then we consider a slightly different setup where data transmission is subject to a latency constraint and hence, it is not feasible to assume that source is uniformly distributed.

In the first problem, we use representable functions to explicitly characterise the set of all admissible tuples. We also elaborate how a secret sharing problem can be formulated as a secure data transmission problem. In the second problem, we raise the concern that an eavesdropper may be able to infer information about the transmitted source by knowing only the size of the transmitted codeword length. We propose schemes on how to extend the one-time pad encryption scheme to avoid leak of information when the sources are not uniformly distributed. We demonstrate that there is a tradeoff between the expected transmitted codeword length and the size of the key that is needed in order to achieve perfect secrecy.

REFERENCES

[1] A. Beimel, N. Livne, and C. Padro, "Matroids can be far from ideal secret sharing," http://www.cs.bgu.ac.il/beimel/Papers/BLP.pdf
[2] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *IEEE Int. Conf. Computers, Systems, and Sig. Proc.*, Bangalore, India, 1984, pp. 175–179.
[3] N. Cai and R. W. Yeung, "A security condition for multi-source linear network coding," in *IEEE Int. Symp. Inform. Theory*, Nice, France, June 2007.
[4] N. Cai and R. W. Yeung, "Network error correction, I: Basic concepts and upper bounds," *Commun. Inf. Syst.*, vol. 6, no. 1, pp. 19–36, 2006.
[5] N. Cai and R. W. Yeung, "Network error correction, II: Lower bounds," *Commun. Inf. Syst.*, vol. 6, no. 1, pp. 37–54, 2006.
[6] T. H. Chan and A. Grant, "Dualities between entropy functions and network codes," *IEEE Trans. Inform. Theory*, pp. 4470 – 4487, Oct. 2008.
[7] T. H. Chan and A. Grant, "Capacity bounds for secure network coding," in *Australian Commun. Theory Workshop*, Feb. 2008.
[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory, 2nd Ed.*, New York: Wiley-Interscience, 2006.
[9] A. K. Ekert, "Quantum cryptography based on Bell's theorem," *Phys. Rev. Lett.*, vol. 67, pp. 661–3, 1991.
[10] L. Guille, T. H. Chan and A. Grant, "The minimal set of Ingleton inequalities," *Proc. 2008 IEEE Int. Symp. on Inform. Theory*, (Toronto, Canada), July 2008.
[11] S.-W. Ho, "On the Interplay between Shannon's Information Measures and Reliability Criteria," in *Proc. 2009 IEEE Int. Symposium Inform. Theory (ISIT 2009)*, Seoul, Korea, June 28-July 3, 2009.
[12] L. Lai, S.-W. Ho and H. V. Poor, "Privacy-Security Tradeoffs in Reusable Biometric Security Systems," in *Proc. The 35th Int. Conf. on Acoustics, Speech, and Signal Processing* (ICASSP), (Texas, USA), Mar. 2010.
[13] S.-W. Ho and T. H. Chan "The Interplay between Key Rate and Channel Capacity." To be submitted to *IEEE Int. Symp. Inform. Theory 2011*.
[14] U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Trans. Inform. Theory*, vol. 39, no. 3, pp. 733–742, Jan 1993.
[15] U. Maurer and S. Wolf, "Unconditionally secure key agreement and the intrinsic conditional information," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 499-514, Mar. 1999.
[16] U. Maurer, "Authentication theory and hypothesis testing," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1350-6, July 2000.
[17] J. Pieprzyk, T. Hardjono, and J. Seberry, *Fundamentals of Computer Security*, Springer Verlag, 2003.
[18] R. L. Rivest, A. Shamir and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–6, Feb. 1978.
[19] A. Shamir, "How to share a secret," Commun. of the ACM 22 (1979) 612–613.
[20] C. Shannon, "Communication theory of secrecy systems," *Bell Sys. Tech. J.*, vol. 28, pp. 656–715, 1949.
[21] P. Shor and J. Preskill, "Simple proof of security of the BB84 quantum key distribution protocol," *Phys. Rev. Lett.*, vol. 85, no. 2, pp. 441–444, 2000.
[22] G. Simmons, "Authentication theory/coding theory," in *Advances in Cryptology – CRYPTO '84*, Santa Barbara, USA, Jan 1985, pp. 411-431.
[23] A. Wyner, "The wire-tap channel," *Bell Sys. Tech. J.*, vol. 54, no. 8, pp. 1355–1387, 1975.
[24] R. W. Yeung, S.-Y. R. Li, N. Cai and Z. Zhang, *Network Coding Theory*, Now Publishers, 2006.
[25] R. W. Yeung, *Information Theory and Network Coding*, Springer, 2008.

Paper 3: Identity Protection against Data Linkage in m-Health

By Vinod Mirchandani, Peter Bertok and Jian Zhong, RMIT, Australia.

Identity Protection against Data Linkage in mHealth

Jian Zhong, Vinod Mirchandani and Peter Bertok
RMIT University

Abstract

The emergence of mobile devices and networks with enhanced and multi-functional capabilities, has spurred the popularity of mobile health (m-Health) technology. Often in m-Health application scenarios, data transfers between medical staff's mobile devices involves substantial amount of private and sensitive information, such as: patients' medical record, medical history, names, addresses, medical insurance number. Due to this involvement of private and sensitive data in the information exchange, it becomes imperative to address the issue of prevention of identity discovery by data linkage. In this regard, many approaches have been proposed in the literature, which have essentially focussed on using strong encryption of the data transferred. Whereas, in this paper, we have proposed a novel solution based on a data linkage algorithm.

Keywords

Identity protection, data linkage, mobile health

INTRODUCTION

Protecting privacy of individuals is a challenging task in a digitalized world. The amount of individual information collected by various data holders is continuously increasing. This paper focuses on two distinct types of information, namely: (i) personally identifying information (PI) such as name, address, gender, date of birth and possibly some identifying number, and (ii) private data, such as: health or clinical information, financial details or insurance records (O'Keefe 2004). Practical application in this paper concentrates on the example of health databases held by clinical staff in a mobile environment, since health data are often containing these two types of data and regarded by individuals as amongst their most sensitive information.

When health data is shared between clinical staff and researchers, the attributes that directly can discover the patient for example i.e. Name and PassportID are removed, while other personal identifying information, such as Gender or Age, which could lead to the possible identification of individuals, are usually attached. Removing these personal identifying information may lead to false medical diagnosis. Unfortunately, a malicious collector may use record linkage techniques (Winkler 1994) between these attributes and externally available information to discover the identity of individuals from the released data. For example, in the mid 1990s, the Massachusetts Group Insurance Commission (GIC) decided to release "anonymised" medical data on state employees that showed their every single hospital visit to help the study by researchers. But it was soon found that 87 percent of all Americans could be uniquely identified using only three pieces of information: ZIP code, date of birth, and gender (Anderson 2009).

Although these data can be encrypted by strong security algorithms, it is very hard for it to be implemented on mobile device not only due to a large size of a medical record (including images) but also limited battery power and computation cost. Medical data shared in such environments has the risk of being maliciously collected and linked to discover individual identities.

Problem Statement

The problem investigated in this paper, is how to reduce the probability of an individual's identity discovery when large amount of private data is shared in a mobile environment. Most existing solutions focus on data modification. While this requires that the data is modified before its release, but it may consequently reduce data accuracy.

Also, in a mobile environment, for example, between medical staff and researchers, large amounts of personal data is shared and strong security methods are not applicable due to the mobile environment constraints stated below. Therefore, issues that must be considered are: (i) large amount of medical data, which is shared between medical staff and researchers; (ii) personal identifying information (PI) must be attached and should be accurate; (iii) constraints of mobile environments cannot be ignored, such as: limited battery power and computation capability.

Outline of the Solution

Our aim is to reduce the probability of disclosure in such an environment, by employing random IDs and a random grouping mechanism. The outline of the solution is:

(i) At first, a number of random keys are generated by both the sender and receiver using the *Elliptic curve Diffie–Hellman* (ECDH) algorithm. Each identity has a group of keys, which are known only by the sender and the receiver.

(ii) Then, each ID is perturbed by a group of keys generated in (i).

(iii) The group of perturbed IDs are processed by a Hash function and a group of random IDs are derived.

(iv) These random IDs are attached to the related identity's personal information (PI), which classifies all individuals' PI by the information type, such as date of birth, gender etc.

(v) Finally, all PIs in each information types are re-ordered, and the original data patterns are regrouped and send out.

Contributions

This paper proposes a method that makes it computationally hard to identify individual's identity in mobile environment without compromising data accuracy. The contributions of our research work documented in this paper are: (i) the proposed algorithm overcomes the drawbacks of existing solutions, and effectively reduces the probability of identity discovery by attaching random ID to selected personal identifying data, and regrouping these PI; (ii) the proposed algorithm does not require modification of the data to keep privacy. As such, all data is kept intact.

The rest of the paper is organised as follows. In Section 2, we introduce some existing solutions, as well as, definitions and notations needed to understand our subsequent discussions and methods. Section 3 details the proposed method and Section 4 gives the implementation and test results, which is followed by the discussion of the proposed method in Section 5. Conclusions are provided in Section 6.

BACKGROUND

In this section, we first discuss existing solutions, and then provide definitions and notations, and finally explain the security algorithms adopted by the proposed method.

Related Work

There are several methods, also called disclosure control or masking methods, presented in research papers, such as simulation (Adam and Wortmann 1989), noise addition (Kim 1986), microaggregation (Domingo-Ferrer and Mateo-Sanz 2002), randomization and perturbation methods (Muralidhar Sarathy 1999, Kooiman et al.1997, Evfimievski et al.2002) etc.

These algorithms were used to modify the initial data in order to achieve individual privacy and preserve data usefulness. However applying these methods requires a proper quantity of data to be modified, which should not be too much to lose important information or too less to protect privacy. A solution proposed in the literature to protect the identity of individuals in a highly sensitive data, is to enforce a property that must hold for the masked data called k-anonymity (Sweeney 2002, Samarati 2001).

The principle of *k-anonymity* is for a masked data *D*, in which every combination of specified attribute values in *D* occurs *k* or more times. Based on this principle, in a masked data that satisfies *k-anonymity*, the probability to identify correctly an individual is at most 1/*k*. This mechanism relies on the diversity of personal identifying information (PI). The higher the diversity, the better the effect. While usually it is very hard to guarantee that all personal identifying information satisfies *k-anonymity*. Therefore, by adopting this mechanism, data may need to be changed before sharing. Also, by increasing *k* the level of protection increases, along with changes to the initial data.

However, the data owner releases the set of attributes (PIs as mentioned in the paper) that do not directly discover an identity, but are used to link to other data sources, which may lead to the disclosure of an individual. In this paper, we use similar definitions and notations (in the Definition and Notation sub-section) and overcome this drawback of the existing solutions.

Definitions and Notations

Assume a data set *D* contains three data fields which are (i) subject identity (ID), such as: name, username, account name, account number, (ii) subject personal information (PI) such as: date of birth, gender, address and (iii) private data (PD), such as: medical history, employment history, life record, information item *i* of subject *X*. Here granular personal information item denotes small pieces of personal information. Information *i* ($0 < i < N$) can refer to date of birth, or gender, or address etc. Also, it could be even finer grained such as referring to the year of birth, the month of birth, the day of birth, gender, street number, street name, suburb, first digit of post code (indicating territory), second digit of post code (indicating suburb), the remaining part of the post code (indicating area) etc.

According to (Anderson 2009), we assume that there is a probability *p* that the identity of a subject can be recovered by a certain amount of PI, which is represented by (1)

$$p_n^X = P(\sum_{i=0}^{n} PI_i^X) \tag{1}$$

where, *P* denotes the operation of analysis conjunction of the collected data PI. Equation (1) denotes the probability of reviewing identity of a subject *X* with collection of *n* attributes of PI that depends on the maliciously collected granular PIs.

Further to equation (1), we have:

$$\begin{aligned} p_n^X(ID^X) &= P[(\sum_{i_1=0}^{n_1} PI_{i_1}^{X_1}) \bigcup (\sum_{i_2=0}^{n_2} PI_{i_2}^{X_2}) \bigcup \cdots \\ &= P[\bigcup_{z=0}^{m} (\sum_{i_z=0}^{n_z} PI_{i_z}^{X_z})] \end{aligned} \tag{2}$$

which, denotes discovery of a subject *X*'s identity ID^X by reviewing a series of PI. Each individual PI may not be able to point out a subject *X* but after a collection and linkage, the ID will be found by analysis and linkage function *P*.

We assume an ID-discoverable group of information is represented as:

$$\begin{aligned} &DISC(ID, |\Omega(PI)|) = p \\ &\Omega(PI^X) = \{PI_i^X : i \in n\} \end{aligned} \tag{3}$$

Where Ω(PI) denotes a subset of the set PIs; *n* denotes

the number of PI items; |Ω(PI)| denotesΩ(PI) with minimum of PI items, which can point out a subject with probability *p*.

Security Algorithms

Elliptic curve Diffie–Hellman (ECDH) (Elaine et al. 2007, Microsoft Technet 2009), *Hash* functions (Mulvey 2010) and *Salt* encryption algorithm (Leong and Tham 1991) are used in the proposed method. Explanations are given below:

Elliptic curve Diffie–Hellman (ECDH) is adopted to generate security *Salts* for both data receiver and sender in the proposed method. Rather than a strong security encryption algorithm, ECDH can be used as a security key generation mechanism. The advantage of ECDH is that it does not rely on security connection, and both sender and receiver can calculate the same secure key even in an insecure communication.

In the proposed mechanism, Salt is used to perturb the attached identities in order to reduce the possibilities of data linkage. The advantage of using Salt is that it can effectively perturb the target string without a high computation cost.

A key ingredient in the construction of the proposed method is a secure hash function operating on an arbitrary bit string and whose values are encrypted by Salt. The reason we chose hash function is that original data cannot be derived from its output. Also, it does not rely on a high computation capability.

PROPOSED METHOD

In this section, we propose a random ID and random grouping algorithm to address the issue in the previous section. The overall model is shown in Figure 1 and composed of three components, namely salt generation (*sGen*), random ID (*rID*) and random grouping (*rGP*).

The process in Figure 1 describes a logical flow of the operation of the random grouping algorithm. Steps depicted in the diagram are explained as follows:

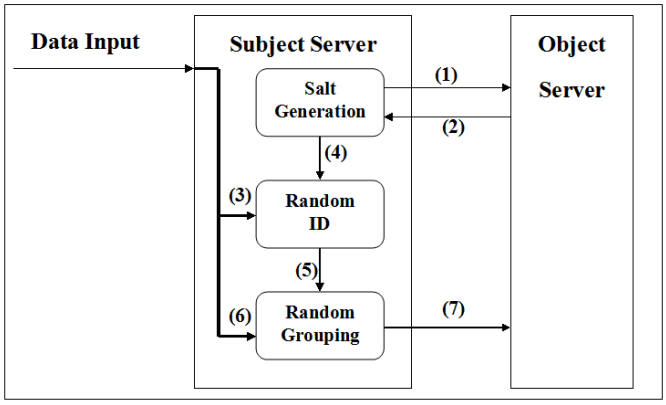

Figure 1: Proposed Model

(i) After the subject server collects data from a number of subjects, it first establishes an Elliptic curve Diffie–Hellman (ECDH) algorithm connection between data sender and the receiver.

(ii) After step (i), a secret ECDH key is derived on both sides.

(iii) Subject ID is loaded by Random ID component.

(iv) The secret ECDH key is loaded by Random ID component and random identities are calculated.

(v) The random IDs are passed to Random Grouping component.

(vi) The rest of data is passed to the Random Grouping component and information belonging to different subjects is a randomized sort.

(vii) Results in step (vi) are sent to an object server.

The processes are detailed from three components mentioned in Figure 1. The participants in the protocol are subject *X*, several groups of PI (belong to subject *X*, but the relationship between each pair of PIs is kept secret), and data receiver *DR*. The protocol is detailed out in three aspects, namely salt generation, random ID algorithm and random grouping algorithm.

Salt Generation

The component, salt generation (*sGen*), is the preliminary procedure of the proposed protocol, which provides a fundamental parameter to support following two components (*rID* and *rGP*). The salt derivation based on *Elliptic curve Diffie–Hellman* (ECDH) algorithm is described in 2.2 and 2.3 sub-sections. In this component, each subject will receive a group of salts, which are only known by the server.

Random ID

Subject's identity and salts are passed to rID component to process random ID function. There are two steps for computing random Ids, which are listed as steps (a) and (b).

(a) Subject X calculate $salt_i^X (ID^X)$

(b) Subject X calculates the hash result of $salt_i^X (ID^X)$, represented by $Hash[salt_i^X (ID^X)]$

In step (b), Hash function gives a non-reversible characteristic to the outputs, which ensures the privacy of subject's ID used in a public network.

According to steps (a) and (b), a set of different hashed IDs are derived from original IDX and each of them are expressed randomly.

Random Grouping

Subject's PIs and rIDs are passed to *rGP* component to complete random grouping. There are mainly five steps to achieve *rGP* algorithm, which are detailed as follows:

(1) Break PI into fine-grained granular data

(2) Attach random ID to each granular data.

(3) Put the same type granular data in a "black box". For example, in black box 1, the granular data type is post code. Then all post-codes from different people are put in box 1.

(4) Randomly change the orders of the granular data in each black box.

(5) Output each different type of granular data from the boxes and reassemble them into the original pattern. For example, we get one granular data "post code" from the box 1, "gender" from box 2, "age" from box 3, hospital ID from box 4 and medical record from box 5. All of these data makes one "person" but it cannot be identified as all of its attributes are from different real person. Without knowing the order of these data, the attacker has a very low possibility to guess a person's identity.

Optimisation Algorithm

Because of extra information (EI) being brought in, the communication cost increases. To keep the extra information at a low level, the optimization algorithm for our method is proposed and detailed as follows: Let the number of person be *n*, each person has *m* subsets on an average each subsets has *z* PI elements. We assume the requirement is that *ip* should lower than *p*, and then the optimization of the proposed method is represented in Figure 2.

```
function optimization
   apply proposed algorithm on n and m
   for all data processing do
      if m*n < =p then
         end
      else if (m*n)^2 <=p then
         only one of z is applied proposed algorithm
         end
      end if
      if (m*n)^(z+1) <=p then
         end
      else
         all elements are applied proposed algorithm
      end if
   end for
```

Figure 2: Optimization Algorithm

IMPLEMENTATION AND TEST RESULTS

In the implementation, each person is assigned an ID, Personal information (PI) (namely date of birth, gender, age, address, post code), and Private Data (PD) (such as all medical related data, blood type, medical history etc.). Full demonstration is described in an extended thesis.

Results

In the evaluation and analysis, identification probability (*ip*) is the probability of pointing out a person by linking several PIs (O'Keefe 2004), extra information (EI) is the information beyond ID, PIs and PDs, such as attached security keys, authentication data sets etc, accuracy (acc) is a measurement of data integrity and accuracy. Detailed evaluation and analysis of the proposed solution are described in an extended thesis.

In Figure 3, the number of persons (subject) is set to 10; each person has 4 subsets data on an average; each subset has 4 PI elements on an average. The figure shows that as the identification probability decreases, the data accuracy of existing solution drops steadily. On the contrary, as the proposed method does not rely on compromising data accuracy to meet the low identification probability, the line remains the same at 100% level. Also, by adopting the proposed method, the identification probability is always lower than 18% in the simulation.

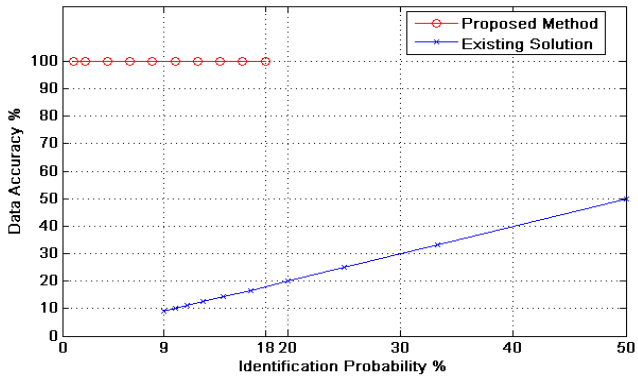

Figure 3: Data accuracy and identification probability

Figure 4 shows that when people's subset number is fixed to 4 and PI elements in it are set to 4, as the number of persons (subjects) increases, the identification probability of the proposed method declines, while for the existing solution, the identification probability remains at a high level of 12.5%.

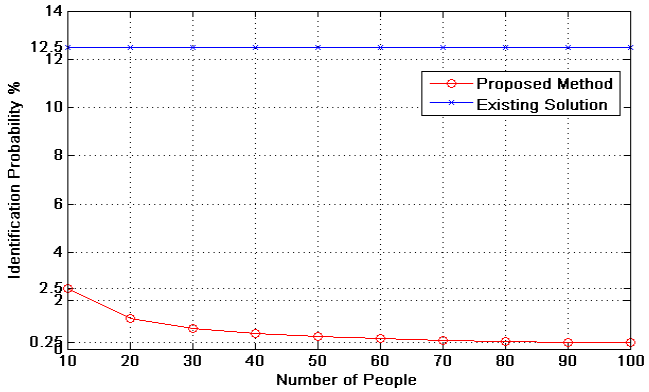

Figure 4: Identification probability and number of people

Figure 5 shows that when the number of people (subject) is set to 10, PI elements in each subset are set to 4 and optimization algorithm applied, then as the number of subset increases, identification probability drops from 2.5% to 0.25% in the proposed method while it remains the same in the existing solutions.

Figure 6 shows that for the same conditions as in Figure 5, as the number of subset for each person increases, the extra information (EI) increases till the number of subset is 9 and decreases dramatically by activating optimization algorithm in the proposed method. The result is similar to that when the number of persons (subjects) increases and number of subsets remains the same.

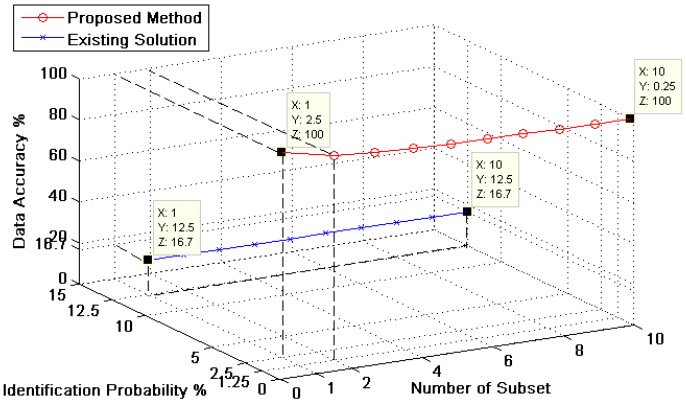

Figure 5: Data accuracy, identification probability and number of subset

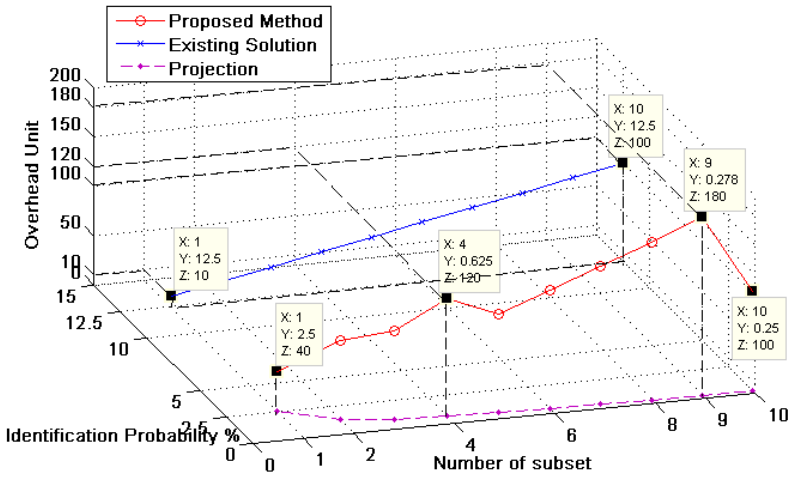

Figure 6: Overhead, Identification probability and number of subset

DISCUSSION

In this section, the evaluation results between the proposed method and existing solutions are discussed from the perspective of (i) identification probability, (ii) data accuracy (iii) overhead (Extra information). With the proposed method, the identification probability is practically eliminated.

Our simulation shows that existing solutions rely on either personal identifying information (PI) diversity or reduction on data accuracy, which fail to protect against data linkage in mobile environment. This motivates us to propose this random ID and random grouping mechanism to improve the level of protection.

The advantages of employing our method are:

(i) Individuals' identity is computationally very hard to be discovered, especially as the number of individual and/or subset increases, the identification probability drops dramatically. From the test results, the probability of discovering a person's identity is easy to be controlled at less than 1%, which is 5 to 50 times better than the existing solutions. Trends are similar when the number of subset and number of element in each subset increase.

(ii) The effect of (i) does not rely on a high computation cost. Data shared via mobile devices are capable enough to keep privacy without compromising constraints such as: limited battery power and computation capability.

(iii) All data shared is intact.

(iv) Does not rely on diverse personal identifying information (PI).

Due to the algorithm employed in the proposed method, extra information (EI) is attached to selected personal identifying information (PI) based on the optimization algorithm. In contrast to the existing solutions, although data size in the proposed method increases slightly, with the optimization algorithm, the balance between a high efficiency of privacy protection and communication is achieved. It results in 5 to more than 50 times improvement in identification probability over existing solutions, whereas the data size is only less than twice of existing solutions on an average.

In the proposed method, elimination of identification probability depends on the arithmetic product of a number of individuals and number of subset - A larger product results in higher effectiveness. Less number of individuals and subsets will not benefit significantly from out method.

CONCLUSION

A mechanism for identity protection in a mobile environment was proposed in this paper. It addresses the issues of identity disclosure by linked data and lost accuracy.

The main results are that the proposed method can practically eliminate the probability of individuals' identity discovery without compromising the data accuracy. It can be implemented in a mobile environment, realize a high quality of privacy protection and cooperate with the mobile environment's limitations.

REFERENCES (alphabetical order)

Adam N.R and Wortmann J.C. 1989. “Security Control Methods for Statistical Databases: A Comparative Study,” ACM Computing Surveys, Vol.21, No.4

Anderson N. 2009. “Anonymized data really isn't—and here's why not,” Published on September 8, 2009. At http://arstechnica.com/tech-policy/news/2009/09/your-secrets-live-online-in-databases-of-ruin.ars

Domingo-Ferrer J. and Mateo-Sanz J.. 2002. “Practical Data-Oriented Microaggregation for Statistical Disclosure Control,” IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No.1, 2002, pp. 189-201

Elaine Barker, Don Johnson, and Miles Smid. 2007. “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised),” In NIST Special Publication 800-56A, March, 2007

Evfimievski A., Srikant R., Agrawal R. and Gehrke J.. 2002. “Privacy Preserving Mining of Association Rules,” Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 217-228

Kim J.J. 1986. “A Method for Limiting Disclosure in Microdata Based on Random Noise and Transformation,” American Statistical Association, Proceedings of the Section on Survey Research Methods, pp. 303-308

Kooiman, P., Willemborg L. and Gouweleeuw J.. 1997. “PRAM: A Method for Disclosure Limitation for Microdata,”,Report, Department of Statistical Methods, Statistical Netherlands, Voorburg.

Leong Philip and Tham C. 1991. “UNIX Password Encryption Considered Insecure,”. In USENIX Winter '91, pp. 269-276, Dallas, TX.

Microsoft Technet. "Overview of the ECDH Algorithm (CNG Example)". Available at http://technet.microsoft.com/zh-cn/library/cc488016.aspx, accessed at 24/03/2009

Mulvey Bret . Accessed April 30 2010. “Hash Functions”. http://bretm.home.comcast.net/~bretm/hash/

Muralidhar K. and Sarathy R.. 1999. “Security of Random Data Perturbation Methods,” ACM Transaction on Database Systems, Vol.24, No. 4, 1999, pp. 487-493

O'Keefe, C. M., Yung, M., Gu, L., and Baxter, R. 2004. “Privacy-preserving data linkage protocols,” Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society, New York, NY, pp. 94-102.

Samarati P. 2001. “Protecting Respondents Identities in Microdata Release,” IEEE Transactions on Knowledge and Data Engineering, Vol. 13, No. 6, 2001, pp. 1010-1027

Sweeney L. 2002. “Achieving k-anonymity Privacy Protection Using Generalization and Suppression”, International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems, Vol. 10, No. 5, 2002, pp. 571-588

Winkler W.E. 1994. “Advanced Methods for Record Linkage”, Proceedings of the section on Survey Research Methods, American Statistical Association, pp. 467-472

COPYRIGHT

Paper 4: Security of Hou-Tan Electronic Cash Scheme

By Xun Yi, Victoria University, Australia.

Security of Hou-Tan Electronic Cash Scheme

Xun Yi

School of Engineering and Science
Victoria University
Melbourne, VIC 8001, Australia
Xun.Yi@vu.edu.au

Abstract. In 2005, Hou and Tan proposed a new electronic cash model to offer trade-off between credit card and traditional off-line electronic cash systems. Based on the new model, they proposed a new electronic cash scheme, which eliminates the withdrawal phase of current electronic cash schemes and enables a customer to generate electronic cash by himself. In this paper, we show that their proposed electronic cash scheme is insecure against a forgery attack, in which an attacker is able to forge electronic coins and spend them at any shop without being detected.

Keywords: electronic cash, group signature, non-repudiation.

1 Introduction

The expansion of the e-commerce has increased the demand for a new payment method suiting the electronic environment better than the current systems based on identifying the customer. Electronic cash is a potential alternative because it can be implemented to provide anonymity. The first electronic cash scheme was introduced by Chaum [7]. Since then, electronic cash schemes are extensively studies in [10, 15, 12, 3, 4, 17, 14, 18, 1, 6].

In general, an electronic cash scheme involves three kinds of players: (1) a payer or consumer; (2) a payee, such as a merchant; and (3) a bank with whom both the payer and payee have accounts. The sequence of events in an electronic cash scheme includes: (1) withdrawal, in which the payer withdraws an electronic coin from her Bank account; (2) payment, in which the payer transfers the electronic coins to the payee; and (3) deposit, in which the payee transfers the electronic coin he has received to his Bank account.

The main idea of electronic cash is that, even though a bank is responsible for issuing electronic coins, and for later accepting them for deposit, the withdrawal and the payment protocols are designed in such a way that it is impossible to identify when a particular coin was spent, i.e., the withdrawal protocol does not reveal any information to the bank that would later enable it to trace how a coin was spent.

As an electronic coin is represented by data, and it is easy to duplicate data, an electronic cash scheme requires a mechanism that prevents a user from

spending the same coin twice (double-spending). There are two scenarios. In the on-line scenario [8, 10], the bank is on-line in each transaction to ensure that no coin is spent twice, and each merchant must consult the bank before accepting a payment. In the off-line scenario [9], the merchant accepts a payment autonomously, and later submits the payment to the bank; the merchant is guaranteed that such a payment will be either honored by the bank, or will lead to the identification (and therefore punishment) of the doublespender.

In 2005, Hou and Tan [16] proposed a new electronic cash model to offer trade-off between credit card and traditional off-line electronic cash systems. Their model is based on a group signature scheme, such as [11, 5], allowing a member of a group to anonymously sign a message on behalf of the group. In their model, a customer opens an account with a bank and obtains a membership secret key at first. To make payment to a shop, the customer can sign a transaction message with the membership secret key while the shop can verify the signature with the group public key published by the bank. Later, the shop deposits the signed payment message to the clearing house where the customer can be identified and the payment can be settled. Based on the model, Hou and Tan proposed a new electronic cash scheme, which eliminates the withdrawal phase of current electronic cash schemes and enable a customer to generate electronic cash by himself.

Hou-Tan electronic cash scheme is more convenient for customers to use than other existing electronic cash schemes. However, this paper shows that Hou-Tan electronic cash scheme is insecure against a forgery attack, in which an attacker is able to forge electronic coins and spend them at any shop without being detected, in this paper.

The rest of this paper is organized as follows. Section 2 introduces group signature on which Hou-Tan electronic cash scheme is built. Section 3 describes Hou-Tan scheme and Section 4 presents an forgery attack to their scheme. Conclusion is drawn in the last section.

2 Preliminary

The notion of group signature is a central anonymity primitive that allows users to have anonymous non-repudiable credentials. The primitive was introduced by Chaum and Van Heyst [11] and it involves a group of users, each holding a membership certificate that allows a user to issue a publicly verifiable signature which hides the identity of the signer within the group. The public-verification procedure employs only the public-key of the group.

Essential to a group signature scheme is a group manager, who is in charge of adding group members and has the ability to reveal the original signer in the event of disputes. In some systems the responsibilities of adding members and revoking signature anonymity are separated and given to a membership manager and revocation manager respectively. By now, many group signature schemes have been proposed, e.g., [11, 5, 13, 2]. In general, a group signature scheme should meet the following basic requirements:

1. Soundness and Completeness: Valid signatures by group members always verify correctly, and invalid signatures always fail verification.
2. Unforgeable: Only members of the group can create valid group signatures.
3. Anonymity: Given a message and its signature, the identity of the individual signer cannot be determined without the group manager's secret key.
4. Traceability: Given any valid signature, the group manager should be able to trace which user issued the signature.
5. Unlinkability: Given two messages and their signatures, the public cannot tell if the signatures were from the same signer or not.
6. No Framing: Even if all other group members (and the managers) collude, they cannot forge a signature for a non-participating group member.
7. Unforgeable tracing verification: The revocation manager cannot falsely accuse a signer of creating a signature he did not create.

These features of group signatures make them attractive for many specialized applications, such as voting and bidding. They can, for example, be used in invitations to submit tenders [13]. All companies submitting a tender form a group and each company signs its tender anonymously using the group signature. Once the preferred tender is selected, the winner can be traced while the other bidders remain anonymous. More generally, group signatures can be used to conceal organizational structures, e.g,, when a company or a government agency issues a signed statement.

3 Hou-Tan Electronic Cash Scheme

Hou-Tan electronic cash scheme [16] is built on the group signature scheme suggested by Camenish and Michels [5]. It involves four parties (palying different roles for the group signature scheme) as follows:

1. The bank is the group registration manager, which maintains the accounts of all customers and registers new customers;
2. The clearing house, which servers as the group revocation manager, clears the transactions between the shop and the customer;
3. The customer, which is the group member, can make payment by signing the transaction message using her membership secret key;
4. The shop can verify the signature using the group public key published by the bank.

Hou-Tan electronic cash scheme, which eliminates the withdrawal phase of current electronic cash schemes and enable a customer to generate electronic cash by himself, as shown in Fig. 1, can be summarized as follows:

1. Setup: The bank (group registration manager) randomly chooses two large primes p, q of the form $p = 2p' + 1$ and $q = 2q' + 1$, where p' and q' are primes, and computes $n = p \cdot q$. Next, it constructs a subgroup G of Z_n^*, obtained by a generator g with a large order. Then, it randomly chooses

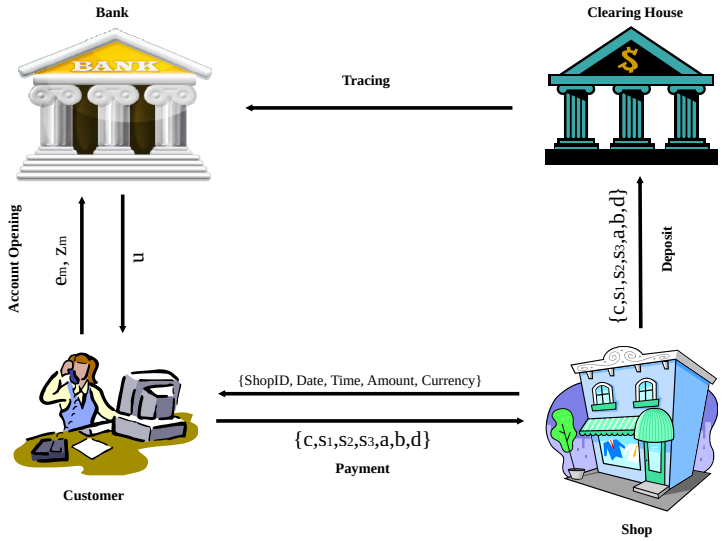

Fig. 1. Hou-Tan Electronic Cash Scheme

two elements $z, h \in G$ with large orders, and publishes n, g, z, h, but keeps p, q secret. Based on (n, g), the clearing house (group revocation manager) randomly chooses its secret key x and publishes its public key $y = g^x (mod\ n)$ along with an one-way hash function $H(\cdot)$.

2. Account Opening: At first, the customer identifies himself to the bank by means of official documents. Next, the customer randomly chooses two large primes e and e_n, computes $e_m = ee_n$ and $z_m = z^{e_n} (mod\ n)$, and commits e_m and z_m along with his signature on $\{e_m, z_m\}$ to the bank. The bank computes $u = {z_m}^{1/e_m} (mod\ n)$ with the knowledge of p and q. Then it sends u to the customer and stores (u, e_m, z_m) with the customer's identity. The customer checks that $z = u^e (mod\ n)$ and keeps (u, e) as her membership secret key.

   Note that if $e_m = ee_n$, $z_m = z^{e_n} (mod\ n)$ and $u = {z_m}^{1/e_m} (mod\ n)$, then

$$u^e = ({z_m}^{1/e_m})^e = (z^{e_n})^{e/e_m} = z^{ee_n/e_m} = z (mod\ n)$$

3. Payment: When the customer makes a payment at a shop, she generates an electronic coin as follows.
   (a) The shop generates and sends to the customer a payment message $m = H(ShopID, Data, Time, Amount, Currency)$

(b) The customer chooses integers w, r_1, r_2, r_3 and computes

$$a = g^w \ (mod\ n) \tag{1}$$
$$b = uy^w \ (mod\ n) \tag{2}$$
$$d = g^e h^w \ (mod\ n) \tag{3}$$
$$t_1 = b^{r_1} y^{-r_2} \ (mod\ n) \tag{4}$$
$$t_2 = a^{r_1} g^{-r_2} \ (mod\ n) \tag{5}$$
$$t_3 = g^{r_3} \ (mod\ n) \tag{6}$$
$$t_4 = g^{r_1} h^{r_3} \ (mod\ n) \tag{7}$$
$$c = H(g, h, y, z, a, b, d, t_1, t_2, t_3, t_4, m) \tag{8}$$
$$s_1 = r_1 - c(e - 2^\ell) \tag{9}$$
$$s_2 = r_2 - cew \tag{10}$$
$$s_3 = r_3 - cw \tag{11}$$

where ℓ is a constant published by the bank.
The generated electronic coin is $(c, s_1, s_2, s_3, a, b, d)$, which is sent to the shop.

The shop verifies whether

$$c = H(g, h, y, z, a, b, d, z^c b^{s_1 - c2^\ell} y^{-s_2}, a^{s_1 - c2^\ell} g^{-s_2}, a^c g^{s_3}, d^c g^{s_1 - c2^\ell} h^{s_3}, m) \tag{12}$$

If so, the shop accepts the coin.

4. Deposit: Later on, the shop sends the electronic coin to the clearing house, where the validity of the electronic coin is checked on the basis of (12) at first and then the identity code $u = b/a^x (mod\ n)$ is determined, where a, b are from the electronic coin and x is the secret key of the clearing house. A report summarizing the deducted and credited amount for each identity code and ShopID is sent to the bank periodically by the clearing house. Based on the identity code u, the bank determines the real identity of the customer at first and then deducts the amount of money from the customer's account, and credits the amount to the shop.

4 Attack on Hou-Tan Electronic Cash Scheme

In this section, we show that Hou-Tan electronic cash scheme is insecure against a forgery attack, where an attack forges a valid coin and spends it at a shop.

In Hou-Tan electronic cash scheme, a membership key is a pair (u, e) such that $z = u^e (mod\ n)$. Since z is public, a trivial membership key $(z, 1)$ is known to everyone because $z = u^e (mod\ n)$ when $u = z$ and $e = 1$. With $(z, 1)$, an attacker can forge an electronic coin and spend it at a shop as follows:

Without need to open account with the bank, the attacker, pretending to be a customer, orders something from a online shop. After receiving the payment message m from the shop, the attacker randomly chooses integers w', r'_1, r'_2, r'_3

and computes

$$a' = g^{w'} \ (mod\ n) \tag{13}$$
$$b' = zy^{w'} \ (mod\ n) \tag{14}$$
$$d' = gh^{w'} \ (mod\ n) \tag{15}$$
$$t_1' = b^{r_1'} y^{-r_2'} \ (mod\ n) \tag{16}$$
$$t_2' = a^{r_1'} g^{-r_2'} \ (mod\ n) \tag{17}$$
$$t_3' = g^{r_3'} \ (mod\ n) \tag{18}$$
$$t_4' = g^{r_1'} h^{r_3'} \ (mod\ n) \tag{19}$$
$$c' = H(g, h, y, z, a', b', d', t_1', t_2', t_3', t_4', m) \tag{20}$$
$$s_1' = r_1' - c'(1 - 2^{\ell}) \tag{21}$$
$$s_2' = r_2' - c'w' \tag{22}$$
$$s_3' = r_3' - c'w' \tag{23}$$

Then the attacker sends the forged electronic coin $(c', s_1', s_2', s_3', a', b', d')$ to the shop, which verifies whether (12) holds or not. Because

$$\begin{aligned} z^{c'} b'^{s_1' - c' 2^{\ell}} y^{-s_2'} &= z^{c'} (zy^{w'})^{r_1' - c'(1-2^{\ell}) - c'2^{\ell}} y^{-r_2' + c'w'} \\ &= z^{c'} z^{r_1' - c'} y^{w'(r_1' - c')} y^{-r_2' + c'w'} \\ &= (zy^{w'})^{r_1'} y^{-r_2'} \\ &= b'^{r_1'} y^{-r_2'} \\ &= t_1' \ (mod\ n) \end{aligned}$$

$$\begin{aligned} a'^{s_1' - c' 2^{\ell}} g^{-s_2'} &= (g^{w'})^{r_1' - c'(1-2^{\ell}) - c'2^{\ell}} g^{-r_2' + c'w'} \\ &= g^{w'(r_1' - c')} g^{-r_2' + c'w'} \\ &= (g^{w'})^{r_1'} g^{-r_2'} \\ &= a'^{r_1'} g^{-r_2'} \\ &= t_2' \ (mod\ n) \end{aligned}$$

$$\begin{aligned} a'^{c'} g^{s_3'} &= (g^{w'})^{c'} g^{r_3' - c'w'} \\ &= g^{r_3'} \\ &= t_3' \ (mod\ n) \end{aligned}$$

$$\begin{aligned} d'^{c'} g^{s_1' - c' 2^{\ell}} h^{s_3'} &= (gh^{w'})^{c'} g^{r_1' - c'(1-2^{\ell}) - c'2^{\ell}} h^{r_3' - c'w'} \\ &= g^{c'} h^{c'w'} g^{r_1' - c'} h^{r_3' - c'w'} \\ &= g^{r_1'} h^{r_3'} \\ &= t_4' \ (mod\ n) \end{aligned}$$

so

$$H(g, h, y, z, a', b', d', z^{c'} b'^{s'_1 - c' 2^\ell} y^{-s'_2}, a'^{s'_1 - c' 2^\ell} g^{-s'_2}, a'^{c'} g^{s'_3}, d^{c'} g^{s'_1 - c' 2^\ell} h^{s'_3}, m)$$
$$= H(g, h, y, z, a', b', d', t'_1, t'_2, t'_3, t'_4, m) = c'$$

Therefore, (12) does hold, and $(c', s'_1, s'_2, s'_3, a', b', d')$ is a valid coin, and the shop will accept it. Because the shop does not know the secret key x of the clearing house and sends the electronic coin to the clearing house later, it is unable to detect whether the identity code $u = b/a^x (mod\ n)$ of the customer is z or not during the payment so as to avoid being cheated.

From (13)-(23), we can see the computation and communication complexities of our attack are the same as those for a customer to generate an electronic coin and make a payment.

5 Conclusion

In 2005, a new electronic cash scheme was proposed to allow a customer to generate electronic coins by himself. In this paper, we have shown that this scheme is insecure against a forgery attack, by which an attacker is able to forge electronic coins and spend them at any shop without being detected. This forgery attack is efficient and feasible. To improve Hou-Tan electronic cash scheme, Boneh et al.'s short group signature [2] can be used to construct more secure and efficient electronic cash schemes. Our future work will perform this attack on similar electronic cash schemes.

References

1. M. Bellare and A. Palacio, "GQ and Schnorr identification schemes: proofs of security against impersonation under active and concurrent attacks", *Proc. Crypto'02*, pages 162-177, 2002.
2. D. Boneh, X. Boyen and H. Shacham, "Short group signatures", *Proc. Crypto'04*, pages 41-55, 2004.
3. S. Brands, "Untraceable off-line cash in wallets with observers", *Proc. Crypto'93*, pages 302-318, 1993.
4. J. L. Camenisch, J. M. Piveteau, and M. A. Stadler, "Blind signatures based on the discrete logaritm problem", *Proc. Eurocrypt'94*, pages 428-432, 1994.
5. J. Camenish and M. Michels, "A group signature scheme based on an RSA-variant", *Proc. Asiacrypt'98*, pages 160-174, 1998.
6. J. Camenisch, S. Hohenberger and A. Lysyanskaya, "Compact e-cash", *Proc. Eurocrypt'05*, pages 302-321, 2005.
7. D. Chaum, "Blind signature for untraceable payment", *Proc. Crypto'83*, pages 199-203, 1983.
8. D. Chaum. "Security without identification: transaction systems to make big brother obsolete", *Communications of the ACM*, 28(10):1030-1044, 1985.
9. D. Chaum, A. Fiat, and M. Naor, "Untraceable electronic cash", *Proc. Crypto'88*, pages 319-327, 1988.

10. D. Chaum, "Online cash checks", *Proc. Eurocrypt'89*, pages 289-3293, 1989.
11. D. Chaum and E. van Heyst, "Group signatures", *Proc. Eurocrypt'91*, pages 257265, 1991.
12. D. Chaum and T. P. Pedersen, "Transferred cash grows in size", *Proc. Eurocrypt'92*, pages 390-407, 1992.
13. L. Chen and T. P. Pedersen, "New group signature schemes", *Proc. Eurocrypt'94*, pages 171181, 1995.
14. Y. Frankel, Y. Tsiounis, and M. Yung, "Indirect discourse proofs", *Proc. Asiacrypt'96*, pages 286-300. 1996.
15. M. Franklin and M. Yung, "Towards provably secure efficient electronic cash", *Proc. ICALP'93*, 1993.
16. X. Hou and C. Tan, "A new electronic cash model", *Proc. IEEE International Conference on Information Technology: Coding and Computing (ITCC'05)*, pages 374-379, 2005.
17. M. Stadler, J. M. Piveteau, and J. Camenisch, "Fair blind signatures", *Proc. Eurocrypt'95*, 1995.
18. Y. S. Tsiounis, *Efficient Electonic Cash: New Notions and Techniques*, PhD thesis, Northeastern University, Boston, Massachusetts, 1997.

Paper 5: An application of consensus clustering for DDoS attacks detection

By Lifang Zi, John Yearwood, Andrei Kelarev, University of Ballarat, Australia.

An Application of Consensus Clustering for DDoS Attacks Detection

Lifang Zi, John Yearwood, Andrei Kelarev

Centre for Informatics and Applied Optimization
Graduate School of ITMS, University of Ballarat
P.O. Box 663, Ballarat, Victoria 3353, Australia
{l.zi,j.yearwood,a.kelarev}@ballarat.edu.au

Abstract

The detection of Distributed Denial of Service (DDoS) attacks is very important for maintaining the security of networks and the Internet. This paper introduces a novel iterative consensus process based on Hybrid Bipartite Graph Formulation (HBGF) consensus function for DDoS attacks detection. First, the features are extracted during feature extraction process based on the analysis of network traffic. Second, several clustering algorithms are applied in combination with the Silhouette index to obtain a collection of independent initial clusterings. Third, the HBGF consensus function and Silhouette index are used to find an appropriate consensus clustering of the initial clusterings. Fourth, this new consensus clustering is added to the pool of initial clusterings replacing another clustering with the worst Silhouette index. Fifth, the process continues iteratively until the Silhouette index of the resulting consensus clusterings stabilizes. This iterative consensus clustering process can improve the quality of the clusters. The experimental results demonstrate that our iterative consensus process is effective and can be used in practice for detecting the separate phases of DDoS attacks.

Keywords: DDoS attacks detection, consensus clustering, iterative consensus process, Silhouette index

1 INTRODUCTION AND PRELIMINARIES

DDoS attacks are one of the major threats to the security of networks and the Internet. A DDoS attack is a coordinated attack on the availability of services of a given target system or network that is launched indirectly through many compromised computing systems [26]. In this paper, we propose a novel iterative consensus process for DDoS attacks detection based on efficient HBGF consensus function. This section contains brief background information on the architecture of DDoS attacks relevant to the discussion of DARPA data set considered in the present paper. The reader is referred to [37] for more broad preliminaries on defence against DDoS attacks.

The DDoS attacks usually have two large parts, also called phases, and involve three classes of agents, see [7]. In the first phase of a DDoS attack, the attacker infiltrates multiple computer systems and installs the DDoS tools, which are scripts capable of generating large volume of traffic under command from the attacker. This phase is called *pre-attack*. The compromised host machines are called *slaves*. The role of a compromised machine can be a handler or an agent. The second phase is the actual DDoS attack. Following a command from the attacker, the handlers generate attack traffic from numerous *agents* installed in the local systems to bring down the target [26]. The target is also called the *victim*.

DDoS traffic often has different packet characteristics compared to normal traffic, which makes it possible to identify and even prevent attacks, see [8]. Considering post-attack forensics, the stored traffic data can be analysed after an attack to help identify the attackers. This technique is called *traceback*. Many traceback schemes have been proposed, see [1, 2, 4, 24, 27, 35, 38] for more details.

In this paper, we present a novel iterative consensus process for DDoS attacks detection. It can be used to identify the phases of DDoS attacks accurately and efficiently. Compared with other statistical approaches to DDoS detection, our method has two advantages. First, there is no need to know the data distribution in advance, since consensus clustering is applied to cluster data. Second, it can iteratively adjust and improve the consensus clustering to detect different patterns of DDoS attacks and achieve a stable clustering result. Our method can be applied in real networks for intrusion detection because of stability of the resulting final consensus clustering.

The remainder of this paper is organized as follows. Section 2 gives an overview of previous research on DDoS attacks detection. Section 3 describes all stages of our novel iterative consensus clustering process in detail. It begins with feature extraction described in Subsection 3.1. Several independent initial clusterings are then obtained as outlined in Subsection 3.2. A consensus clustering algorithm used in the process and the flow-chart of the iterations of the whole process are presented in Subsection 3.3. Examples of experimental results are included in Section 4. The conclusion is given in Section 5.

2 PREVIOUS WORK

Various approaches to the defence against DDoS attacks have been very actively investigated by many authors and numerous valuable results have been obtained. We believe that all previous ideas considered in the literature have made valuable contributions and will remain essential for future developments. Indeed, it is very important to have a broad spectrum of methods, which can be applied to defending against the attacks, first of all since the character of attacks and software used to launch them will continue to evolve. The need in a variety of approaches is also confirmed by a number of well-known theoretical results, which prove that there never exists one particular method which can achieve best performance for all data sets. Theorems of this sort have become known as 'no free lunch' theorems (see, for example, [33]).

The present paper is not a survey article, and so without trying to be complete we have included only a few examples of recent articles on the topic, where more comprehensive bibliographies can be found. In particular, recently many authors have investigated the detection of DDoS attacks using novel efficient methods based on chaos theory [5], intelligent decision prototypes [6], entropy [40] and a generalized entropy metric [21], as well as several concepts of information theory [41]. Advanced methods for the filtering of DDoS packets and their classification have been considered in [35] and [36], respectively.

Earlier, several statistical methods have also been applied to the detection of DDoS attacks, see [3, 18, 20]. These methods usually analyse some parameters of the network traffic in order to identify statistical patterns of the traffic. Filtering can also be used for defence against DDoS attacks [14, 17]. It often includes scanning of IP packet headers and checking to see if they meet certain criteria [26]. A methodology for automatically extracting probable precursors of DDoS attacks using MIB (Management Information Base) Traffic Variables was introduced in [3], but the method could not solve the problem when the victim and attacker were on different networks. Neural networks have also been used for detection of attacks in [11] and [29]. A hierarchical clustering method for proactive DDoS detection has been also considered in [20]. Our paper introduces a novel stable consensus clustering approach, which can be applied for the detection of various phases of DDoS attacks.

3 ITERATIVE PROCESS BASED ON CONSENSUS FUNCTION

This section describes all stages of our novel iterative consensus clustering process for finding a stable reliable consensus clustering of the network traffic data and determining the cluster structure of DDoS attacks. The whole process is illustrated in Figure 1. It starts with feature extraction outlined in Subsection 3.1. Several independent initial clustering are then found as explained in Subsection 3.2. HBGF consensus functions and iterations of the process are presented in Subsection 3.3.

3.1 Feature Extraction and Preprocessing

A DDoS attack usually involves the selection of slaves, communication and the attack. We can observe traffic parameters changing during various phases of the attack to detect and identify phases.

In the first step, the attacker sends ICMP Echo Request to find slaves. This is also called an *IPsweep* [3]. In this procedure, many ICMP packets are generated. Therefore, the occurrence rate of ICMP packets may be abnormally high. For the communication and compromise between different slaves, increased volume of a specific traffic type such as UDP, TCP SYN and ICMP packets can be sent for message exchange. Therefore,

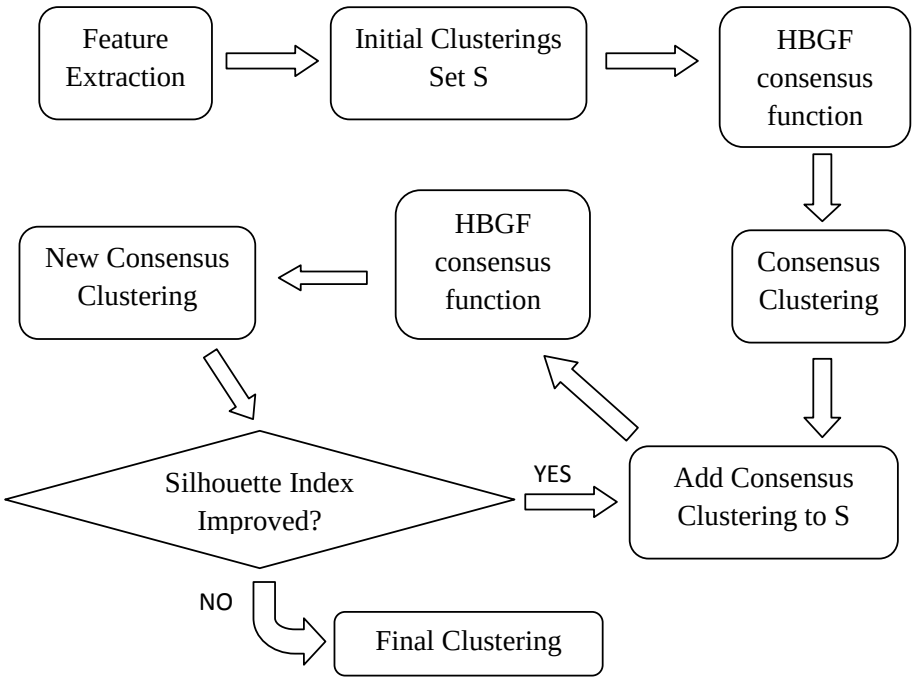

Figure 1: Iterative consensus process based on HBGF consensus function

the occurrence rates of these types of packets can indicate the preparation for launching a DDoS attack.

The distribution of source IP address, destination IP address, source port and destination port can also provide worthwhile information. In order to measure the degree of divergence, we used the entropy, since it is a standard notion of information theory frequently considered in this context (see [8] and [20]) and also because it it has been shown to be quite efficient for applications in this area, see [21] and [40].

The entropy value gives a description of the corresponding distribution of a random variable. The larger the entropy, the more dispersive the variable is. Entropy can be computed on a sample of consecutive packets. If an information source has n independent symbols and transmits each symbol i with probability P_i, then the entropy H is defined as $H = -\sum_{i=1}^{n} p_i \log_2 p_i$, see [25].

In the IPsweep phase, the attacker spreads packets to find slaves. The entropy value of source IP address becomes small and that of destination IP address increases. On the contrary, in the attack phase, attack packets have diverse source IP addresses and a target destination IP address. The entropy value of source IP address increases and that of destination IP address converges to a very small value. Similarly, the entropy values of source and destination port numbers may increase, since some types of DDoS attacks use random port numbers in the attack. A DDoS attack may use a specific type of packets, for example ICMP flood attack. Hence, the entropy value of packet type can be used. If the entropy value of packet type is very small, it may be an indication of a DDoS attack.

In DDoS attacks, a large number of packets are generated. Therefore, the number of packets in a certain time interval is worth observing as well. Following [20], we use nine features to analyse packets:

(1) entropy of destination IP address;

(2) entropy of destination port number;

(3) entropy of source IP address;

(4) entropy of source port number;

(5) entropy of packet type;

(6) occurrence rate of ICMP packets;

(7) occurrence rate of TCP SYN packets;

(8) occurrence rate of UDP packets;

(9) number of packets.

All of these variables are calculated over a 1s interval. Before using these features as input to the clustering algorithms, each variable x was normalized as $x' = \frac{x-\overline{x}}{\sigma}$, where $\overline{x}$ and σ are the mean and standard deviation. Normalisation eliminates differences between scales used to measure the variables [20].

3.2 Initial Clusterings

Looking at the features described in Section 3.1, we used four clustering algorithms implemented in WEKA, SimpleKMeans, Cobweb, EM and FarthestFirst, and obtained an ensemble of independent initial clusterings $C = \{C^{(1)}, C^{(2)}, \ldots, C^{(k)}\}$, where, for each clustering $C^{(i)}$, the whole data set D is a disjoint union of the classes so that $C^{(i)} = \{C_1^{(i)}, C_2^{(i)}, \ldots, C_{k_i}^{(i)}\}$ and $D = C_1^{(i)} \dot\cup C_2^{(i)} \dot\cup \ldots \dot\cup C_{k_i}^{(i)}$, for all $i = 1, \ldots, k$.

SimpleKMeans is the classical k-mean clustering algorithm described, for example, in [15], Section 3.3.2, and [32], Section 4.8, see also [31]. This algorithm randomly chooses k packets as centroids of clusters at the initialization stage. Every other packet is allocated to the cluster of its nearest centroid. After that each iteration finds new centroids of all current clusters as a mean of all members of the cluster. This is equivalent to finding the point such that the sum of all distances from the new centroid to all other sequences in the cluster is minimal. Then the algorithm reallocates all points to the clusters of the new centroids. It proceeds iteratively until the centroids stabilize. We used SimpleKMeans with the default Euclidean distance.

The outcomes of the k-means algorithm often depend on the initial selection of the very first centroids. The outcome of the SimpleKMeans in the WEKA implementation depends on the value of the input parameter "seed". To overcome the dependence of the outcome on the random choice of this parameter we ran it with several random selections of the "seed", as explained below.

Cobweb is the WEKA implementation of the Cobweb and Classit clustering algorithms described in [10] and [12], respectively. EM is the expectation maximisation algorithm in WEKA, and FarthestFirst is a WEKA implementation of the clustering algorithm described in [13]. Cobweb, EM, FarthestFirst and SimpleKMeans produce clusterings given a fixed number of clusters as an input parameter.

In order to determine the appropriate number of clusters we used the Silhouette index. The *Silhouette index* of a clustering is a robust measure of the quality of the clustering introduced in [23]. The Silhouette index $\mathrm{SI}(x)$ of each observation x is defined as follows. If x is the only point in its cluster, then $\mathrm{SI}(x) = 0$. Denote by $a(x)$ the average distance between x and all other points of its cluster. For any other cluster C, let $d(x, C)$ be the average distance between x and all points of C. The minimum $b(x) = \min\{d(x, C) : x \notin C\}$ is the distance from x to its nearest cluster C to which x does not belong. Finally, put

$$\mathrm{SI}(x) = \frac{b(x) - a(x)}{\max\{a(x), b(x)\}} \tag{1}$$

The Silhouette index of the whole clustering is found as the average index over all observations. The Silhouette index always belongs to $[-1, 1]$. The partition with highest Silhouette index is regarded as optimal.

For each initial clustering algorithm and each value of the "seed", we ran it several times increasing the number of clusters, as recommended in [23]. The clustering with the best Silhouette index was included in the set of initial clusterings to be processed by consensus clustering algorithm at the next stage. The same procedure of determining the number of clusters was applied for other initial clustering algorithms too. All these initial clustering algorithms can process our data without any additional data transformations or encoding. The outcomes of all of these clustering algorithms often depend on the initial random selections made during the start of their iterations. A standard approach is to run them for several random selections of input parameters, as in [16]. In WEKA, the outcomes of these algorithms depend on their input parameter

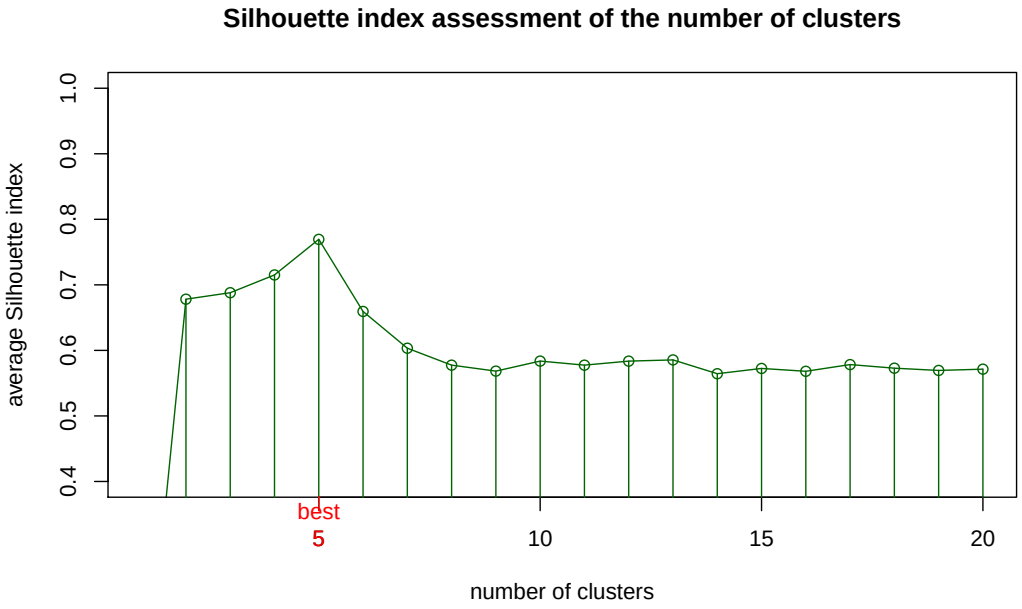

Figure 2: Iterative consensus process based on HBGF consensus function

"seed". For each value of the number k of clusters from 2 to 20, we ran every initial clustering algorithm 10 times, and then chose the number k, which produced the best average Silhouette index. The clustering with this value of k and the best Silhouette index was then included into the pool of initial clustering. This provided sufficient input for the consensus clustering algorithms considered in the next section. Thus, we used Silhouette index and clustering algorithms Cobweb, EM, FarthestFirst and SimpleKMeans, which could process our sample directly and produced sufficient input for the next stage of our approach.

3.3 HBGF Consensus Function and Iterations

Given an ensemble of several independent clusterings on the same data set, the HBGF consensus function proposed in [9] was applied to form a new common consensus clustering. We used HBGF consensus function, because it is known to have several advantages over the other algorithms, Instance-Based Graph Formulation and Cluster-Based Graph Formulation considered in [9]. It was originally introduced in [28] using different terminology, see also [39].

Hybrid Bipartite Graph Formulation, HBGF, is a consensus function proposed in [9] and based on a bipartite graph. It has two sets of vertices: clusters and elements of the data set. A cluster C and an element d are connected by an edge in this bipartite graph if and only if d belongs to C. An appropriate graph partitioning algorithm is then applied to the whole bipartite graph. The final clustering is determined by the way it partitions all elements of the data set. We refer to [9, 28, 30] for more details.

We used METIS graph partitioning software described in [19]. The weights of edges in the input files of METIS must all be strictly greater than zero, which means that it can handle only complete weighted graphs. In order to apply it to a bipartite graphs, we had to set the weights of all edges not present in the graph to 1 and to rescale the weight of all other edges by multiplying them with a constant to make them larger than 10,000. This ensured that METIS removed all nonexistent edges from the graph and then continued analysing the resulting bipartite graph.

The Silhouette index was used again in order to determine the appropriate number of clusters for HBGF consensus function too, as in Section 3.2. We ran HBGF consensus algorithm several times increasing the number of clusters. The HBGF consensus clustering with the best Silhouette index was then added to the pool of initial clusterings, as illustrated in Figure 2. The clustering with the worst Silhouette index was removed in order to avoid increasing the computational cost of further applications of the HBGF consensus

algorithm.

This iterative process continued as illustrated in Figure 1 until the average Silhouette indices of the resulting HBGF consensus clusterings stabilized. The last HBGF consensus clustering obtained in the last iteration is regarded as an output of the whole iterative consensus process.

4 EXPERIMENTAL RESULTS

4.1 Description of the DARPA Data Set

In our experiments, we used the 2000 Intrusion Detection Scenario Specific Data Set [22] of the Defense Advanced Research Projects Agency, DARPA. This attack scenario is carried out over multiple sessions and has five steps, which are usually also called phases:

Phase 1: IPsweep of the AFB (Air Force Base) from a remote site.

Phase 2: Probe of live IP's to look for the sadmind daemon running on Solaris hosts.

Phase 3: Breaking via a known sadmind vulnerability on these hosts.

Phase 4: Installation of a trojan mstream DDoS software on these hosts.

Phase 5: Launching the DDoS attack.

In this attack scenario, the attacker can only launch a DDoS attack via the DMZ network. The packets collected at the sniffer in the DMZ network are kept in the DMZ Tcpdump file. We used the DMZ Tcpdump file as our testing data set.

In phase 1, the attacker sends ICMP Echo Requests and listens for ICMP Echo Replies to determine, which hosts are alive. Besides, most of packets passing by the network in phase 1 are ICMP packets. In phase 2, each of the hosts discovered in phase 1 are probed by sadmind exploit program that generates UDP packets to determine the hosts, which have vulnerabilities. Phase 3 and phase 4 occur when the attacker intrudes agent hosts and installs DDoS software. Therefore, no significant changes in network traffic result at these stages. do not appear. In phase 5, packets collected in the DMZ network are not attack packets, but response packets to the spoofed IP addresses of the attack packets [20].

4.2 Results and Analysis

The average value of each variable in each cluster is given in Table 1. Cluster 1 and Cluster 2 are normal phases. These two clusters have no significant features to show that they are specific phases of the attack. In Cluster 3, the occurrence rate of UDP and ICMP packets are the highest. Therefore, it is most likely that Cluster 3 is associated with the pre-attack phase, which includes the first two phases of the attack scenario.

Cluster 4 corresponds to the attack phase itself. It has very low entropy values of source IP address. On the contrary, the entropy values of destination IP address, source port number, destination port number are very high. In this attack scenario, the agents use randomly spoofed source IP address, source port number and destination port number. At the same time, the destination IP address is the target. With respect to attack phase, the entropy values of source IP address, source port number and destination port number should be much bigger than the entropy value of destination IP address. However, packets collected in the DMZ network are the response packets to the attack packets, so we get the opposite result. Another obvious feature is that the number of packets in Cluster 4 is quite large. A DDoS attack usually uses a lot of packets to block the victim's network.

In Cluster 5, the occurrence rate of TCP SYN is higher than in other clusters, but the number of packets is not big enough to conclude that this is a flooding attack.

Table 1: Average values of variables in the clusters of final consensus clustering

Variable	Cluster 1 normal	Cluster 2 normal	Cluster 3 pre-attack	Cluster 4 attack	Cluster 5 normal
Entropy of destination IP	1.98	0.69	1.95	11.96	1.88
Entropy of destination port	1.75	0.56	2.18	11.98	2.57
Entropy of source IP	2.02	0.67	1.92	0.04	1.91
Entropy of source port	1.82	0.65	2.12	12.20	2.42
Entropy of packet type	0.31	0.06	1.09	0.00	0.13
Occurrence rate of ICMP	0.00	0.00	0.25	0.00	0.00
Occurrence rate of TCP SYN	0.00	0.00	0.03	0.00	0.05
Occurrence rate of UDP	0.00	0.00	0.36	0.00	0.01
Packet number	42	15	35	6375	96

Our iterative consensus process determined five clusters in the data set. The average values of all variables for each cluster are given in Table 1.

5 CONCLUSION

In this paper, we have investigated a novel iterative consensus process based on HBGF consensus function for the identification of DDoS attacks. To evaluate this method, we experimented with a sample of data from the 2000 DARPA Intrusion Detection Scenario Specific Data Set. The Silhouette index was applied to determine the appropriate number of clusters in consensus clusterings. These experimental results show that our novel method can produce stable clusterings useful for DDoS attacks detection. The authors are grateful to three referees for comments which have helped to improve the text of this article.

References

[1] A. Belenky and N. Ansari. IP traceback with deterministic packet marking. *IEEE Communications Letters*, 7(4):162–164, 2003.

[2] H. Burch and B. Cheswick. Tracing anonymous packets to their approximate source. In: *Proceedings of the Fourteenth USENIX System Administration Conference* (LISA 2000), 2000.

[3] J.B.D. Cabrera, L. Lewis, X.Z. Qin, W.K. Lee, R.K. Prasanth, B. Ravichandran, R.K. Mehra. Proactive detection of Distributed Denial of Service attacks using MIB traffic variables – a feasibility study. In: *Proc. Seventh IEEE/IFIP Internat. Symposium on Integrated Network Management*, pp. 1–14, 2001.

[4] A. Chonka, W. Zhou and Y. Xiang. Protecting web services with service oriented traceback architecture. In: *Proc. IEEE 8th Internat. Conf. Computer and Information Technology*, CIT 2008, pp. 706–711, 2008.

[5] A. Chonka and W. Zhou. Chaos theory based detection against network mimicking DDoS attacks. *IEEE Communications Letters*, 13(9):717–719, 2009.

[6] A. Chonka, W. Zhou, J. Singh and Y. Xiang. Detecting and tracing DDoS attacks by intelligent decision prototype. In: *Proc. 6th Annual IEEE Internat. Conf. Pervasive Computing and Communications*, pp. 578–583, 2008.

[7] P.J. Criscuolo, Distributed Denial of Service: Trin00, Tribe Flood Network 2000 and Stacheldraht CIAC-2319, Dept. Energy Computer Incident Advisory (CIAC), Lawrence Livermore National Laboratory, Rev.1, UCRL-ID-136939, 2000.

[8] L. Feinstein, D. Schnackenberg, R. Balupari and D. Kindred. Statistical approaches to DDoS attack detection and response. In: *Proceedings of the DARPA Information Survivability Conference and Exposition*, 1:303–314, 2003.

[9] X.Z. Fern and C.E. Brodley. Cluster ensembles for high dimensional clustering: an empirical study. In: *Proceedings of the of International Conference on Machine Learning*, ACM, Banff, Alberta, Canada, pp. 36–43, 2004.

[10] D. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2):139–172, 1987.

[11] D. Gavrilis and E. Dermatas. Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features. *Computer Networks*, 48(2):235–245, 2005.

[12] J.H. Gennari, P. Langley and D. Fisher. Models of incremental concept formation. *Artificial Intelligence* 40:11–61, 1990.

[13] S. Hochbaum. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985.

[14] Y. Huang and J. Pullen. Countering denial-of-service attacks using congestion triggered packet sampling and filtering. In: *Proceedings of the Tenth International Conference Computer Communications and Networks*, 490–494, 2001.

[15] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. London: Prentice Hall, 1988.

[16] A.K. Jain, M.N. Murty and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[17] C. Jin, H. Wang and K. Shin. Hop-count filtering: an effective defense against spoofed DDoS traffic. In: *Proceedings of the Tenth ACM Conference on Computer and Communications Security*, 30–41, 2003.

[18] S. Jin and D. Yeung. A covariance analysis model for DDoS attack detection. In: *Proceedings of the IEEE International Conference on Communications*, 4:1882–1886, 2004.

[19] G. Karypis and V. Kumar. METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, Technical report, University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Centre, Minneapolis, 1998.

[20] K. Lee, J. Kim, K. Kwon, Y. Han and S. Kim. DDoS attack detection method using cluster analysis. *Expert Systems with Applications*, 34(3):1659–1665, 2008.

[21] K. Li, W. Zhou, S. Yu and B. Dai. Effective DDoS attacks detection using generalized entropy metric. *Lecture Notes in Computer Science*, 5574:266–280, 2009.

[22] MIT Lincoln Laboratory. *DARPA Intrusion Detection Scenario Specific Datasets*. 2000. Available at http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/2000/LLS_DDOS_1.0.html.

[23] P.J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comp. Appl. Math.* 20:53–65, 1987.

[24] S. Savage, D. Wetherall, A. Karlin and T. Anderson. Network support for IP traceback. *ACM/IEEE Transactions on Networking*, 9(3):226–237, 2001.

[25] C.E. Shannon and W. Weaver, A mathematical theory of communication, *Bell System Technical Journal*, 27:379–423, 1948.

[26] S.M. Specht and R.B. Lee. Distributed Denial of Service: taxonomies of attacks, tools and countermeasures. In: *Proceedings of the of 17th International Conference on Parallel and Distributed Computing Systems*, 543–550, 2004.

[27] R. Stone. CenterTrack: an IP overlay network for tracking DoS floods, In: *Proceedings of the USENIX Security Symposium*, 2000.

[28] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *J. Machine Learning Research*, 3:583–617, 2002.

[29] W. Streilein, R. Cunningham and S. Webster. Improved detection of low-profile probe and denial-of-service attacks. In: *Proceedings of the 2001 Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection*, 2001.

[30] A. Topchy, A.K. Jain and W. Punch. Combining multiple weak clusterings. In: *Proceedings of the Third IEEE International Conference on Data Mining*, 331–338, 2003.

[31] D.S. Turaga, M. Vlachos and O. Verscheure. On k-means cluster preservation using quantization schemes. In: *Proceedings of the Nineth IEEE International Conference on Data Mining*, ICDM09, Miami, Florida, USA, 533–542, 2009.

[32] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier/Morgan Kaufman; Amsterdam, 2005.

[33] D.H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.

[34] Y. Xiang and W. Zhou. A defense system against DDoS attacks by large-scale IP traceback. In: *Proc. 3rd Internat. Conf. Information Technology and Applications* (4-7 July 2005, Sydney, Australia). IEEE Computer Society, Los Alamitos, Calif., pp. 431–436, 2005.

[35] Y. Xiang and W. Zhou. Intelligent DDoS packet filtering in high-speed networks. *Lecture Notes in Computer Science*, 3758:395–406, 2005.

[36] Y. Xiang and W. Zhou. Classifying DDoS packets in high-speed networks. *Internat. J. Computer Science & Network Security*, 6(2B):107–115, 2006.

[37] Y. Xiang and W. Zhou. Defending against distributed denial of service. In: M. Quigley (eds), *"Encyclopedia of Information Ethics and Security"*, pp. 121–129. Hershey, PA: IGI Global, 2008.

[38] Y. Xiang, W. Zhou and M. Guo. Flexible deterministic packet marking: an IP traceback system to find the real source of attacks, *IEEE Transactions on Parallel and Distributed Systems*, 20(4):567–580, 2009.

[39] J. Yearwood, D. Webb, L. Ma, P. Vamplew, B. Ofoghi and A. Kelarev. Applying clustering and ensemble clustering approaches to phishing profiling. In: *Data Mining and Analytics 2009*, Proceedings of the Eighth Australasian Data Mining Conference: AusDM 2009, (1-4 December 2009, Melbourne, Australia) Conferences in Research and Pricatice in Information Technology, 101:25–34, 2009.

[40] S. Yu and W. Zhou. Entropy-based collaborative detection of DDOS attacks on community networks. In: *Proc. 6th Annual IEEE Internat. Conf. Pervasive Computing and Communications*, pp. 566–571, 2008.

[41] S. Yu, W. Zhou and R. Doss. Information theory based detection against network behavior mimicking DDoS attacks. *IEEE Communication Letters*, 12(4):319–321, 2008.