SECURING WEB-BASED APPLICATIONS WITH PRIVACY PRESERVING TRAFFIC PADDING

A.Bhargavi[1], V.Baby[2]

[1]*P.G Student, Software Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Telangana, India*
[2]*Associate Professor, Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Telangana, India*

Abstract

Now-a-days, Web-based applications are gaining great reputation as they require less client side resources and are easier to deliver and maintain. But they also have new security and privacy challenges. The encrypted traffic of many popular Web applications may actually disclose highly sensitive data due to the side channel attack, and consequently lead to serious breaches of user privacy. An eavesdropper potentially identifies the applications' internal state transitions and the corresponding users' inputs based on packets' sizes and/or timing analysis. The existing solution such as random padding and packet size rounding were proven to incur prohibitive overhead but were failing to assure sufficient privacy requirement. For preventing such side channel attack is to pad packets such that each packet will no longer map to a unique input. Padding packets results in additional communication and processing overhead. One extreme cases to pad all packets to the identical size, namely, maximizing. In the proposed system a similarity has been identified between the privacy preserving traffic padding (PPTP) issue and well studied problem privacy preserving data publishing (PPDP). Based on such similarities PPTP model encompassing the privacy requirements, padding costs, and padding methods, and then formulate problems under different application scenarios. These algorithms have been designed for solving the PPTP problems in polynomial time with acceptable overhead. Through experiments an attempt had been made to increases the effectiveness and efficiency of these algorithms than the existing solutions using the real world search engine.

Keywords: *Traffic Padding, Web Application, Side-Channel Leak, PPTP*

----------***----------

1. INTRODUCTION

Web-based applications are becoming increasingly pleasing to all. In comparison to their tabletop things balancing another, internet applications request less client-side resources and are more comfortable to give birth and support through using the net of an insect browser as a thin client. On the other hand, internet applications also present new safety and right not to be public questions, partly because the untreated internet has necessarily become an integral part of such applications for taking the unbroken stretch effect on one another between users and servers. Nearby work-room showed that the encrypted business trade of many pleasing to all net of an insect applications may actually come to light highly sensitive knowledge for computers, and consequently lead to serious overrules of user right not to be public[2]. specially, by looking for nothing like it designs put on view in packets sizes and/or timing, a person overhearing private talk can possibly make out an applications inside state changes and the being like (in some way) users inputs. In addition, such side channel attacks are made clear to be coming into existence everywhere and deep to most net of an insect applications needing payment to many intrinsic qualities of such applications, such as low entropy inputs, different useable thing ends, and stateful making connections.

Taking one pleasing to all real-world looking-for engine as an example, Table I shows the sizes and directions of packets observed between users and the looking-for engine. observe that needing payment to the user-friendly auto-suggestion point, with each push button on keyboard, the browser sends a b-byte packet to the server; the server then answers with 2 packets of 60 bytes and s bytes, separately; at last, the browser sends a 60-byte packet to the server. In addition, in the same input line, the b value of the first push button on keyboard is about 50 bytes larger than that of the second one while each coming after push button on keyboard increases the b value by one byte from the third push button on keyboard, and the s value depends both on the current push button on keyboard and on all the going in front of ones. Clearly, needing payment to the fixed good example in packet sizes (first, second, and last), the packets being like (in some way) to each input cord can be taken to be from observed business trade, even though the business trade has been encrypted.

TABLE 1: User inputs and corresponding packet sizes

User input	Observed Directional Packet sizes			
bee	641→,	←60,	←544,	60→,
	585→ ,	←60,	←555,	60→,
	586→ ,	←60,	← 547,	60→,
cab	641→,	←60,	←554,	60→,
	585→ ,	←60,	←560,	60→,
	586→ ,	←60,	←5587,	60→,
	(b bytes)		(s bytes)	

Similar business trade designs have also been observed in different groups of net of insect applications. As an outcome of that, here take to be true a worst example scenario in which a person overhearing private talk can point without error business trade related to a net of an insect application(such as using de-anonymizing techniques) and give position of packets for user inputs using the above way of doing. In this use look for engines as examples in this paper needing payment to their separate and representative designs. In material fact, the s value can be larger and more different.

In addition, the size of the third packet provides a good sign of the input itself (which again can be discovered in many net of insect applications). Specially, Table II shows the s value for person in a work (a, b, c and d) entered as the first (second column) and second (3-6 columns) push button on keyboard for a different looking-for engine. Observe that the s value for each person in a work entered as second push button on keyboard is different from that it is entered as the first, since the packet size now depends on both the current push button on keyboard and the going in front of one. clearly, every input line can be uncommonly taken to be by putting together observations of packet sizes about the coming one after another number of buttons pushed on keyboard (for simplexes, we only take into account a-d puts together here, in view of the fact that in material fact it may take more than number of buttons pushed on keyboard to uncommonly make out an input line).

TABLE 2: s value for each character entered as the first (second column) and second (3-6 columns) keystroke

		Second keystroke			
First keystroke		a	b	c	d
a	509	487	493	501	497
b	504	516	488	482	481
c	502	501	488	473	477
d	516	543	478	509	499

A natural answer for putting a stop to such a side narrow way attack is to thick material packets such that each packet size will no longer map to a nothing like input (one very much example is to thick material all packets to the same size, namely, making greatest degree). In this example, we should put soft material round s-byte such that each packet size maps to at least k=2 different inputs, namely, 2-indistinguishability.

However, such an answer does not come free, since thick material packets will outcome in added news and processing overhead. In fact, it has been made clear that a straightforward answer, such as random thick material (joining a random-length thick material within a given space (times) between to a packet) and rounding (rounding packet sizes to the nearest spaces (time) between), may cause a prohibitive overhead. In this way, we face seemingly being out of harmony ends, purposes. First, the point or amount unlike in packet sizes needs to be enough made lower, less to put a stop to eavesdroppers from noting between different users inputs based on being like (in some way) packet sizes. Second, the overhead for doing such right not to be public system of care for trade should be made seem unimportant. At last, a trade-off naturally has existence between these ends.

Now take into account a different way for thick material the small parcels as made clear in Table III. The first and last columns separately play or amusement the s value and being like (in some way) person in a work with its prefix (e.g., (c)d means the person in a work d is entered as the second push button on keyboard after its prefix c is entered for the same input line). The middle columns give selections for thick material small parcels (though not given view here, there certainly have existence many other selections). specially, each thing for which selection is made first makes a division the number of buttons pushed on keyboard into three (or) thick material groups, as pictured by the (being away of) horizontal lines. small parcels within the same thick material group are then put, with soft material round in such a way that the being like (in some way) s values become the same to the greatest point value in that group, and thus the persons in a work inside the group will no longer be measurable from each other by the s values. The end now is to discover a thick material thing for which selection is made that can make ready enough right not to be public system of care for trade groups, while at the same time and meanwhile make seem unimportant the thick material price. Note that the parts in folds such small parcel information is useful for most net of an insect requests, as this privacy preserving traffic padding(PPTP) hard question is naturally connected with another well studied hard question, namely, privacy preserving data publishing (PPDP) . Such a connection between the 2 issues suggests may get use of many have existence efforts in the PPDP lands ruled over to house the PPTP question under discussion.

TABLE 3: Mapping PPTP to PPDP

s Value	Padding		(Prefix)Char
	Option 1	Option 2	
473	477	478	(c)c
477	477	478	(c)d
478	499	478	(d)b
499	499	509	(d)d
501	509	509	(c)a
509	509	509	(d)c
Quasi-ID	Generalization		Sensitive Value

In this paper, first present a design to be copied of the PPTP question under discussion based on the mapping to PPDP, which formally gives account of qualities the effect on one another between users and net of an insect requests, the observation made by eavesdroppers, the right not to be public thing needed, and the overhead of thick material. Based on the design to be copied, and then put clearly several PPTP problems under different things taken as certain, and have a discussion the being complex. Make clear to that making seem unimportant thick material price under a given right not to be public thing needed is generally unworkable. next, design several heuristic algorithms for getting answer to, way out of the PPTP problems in more than one math part time with say yes (to offer) overhead. At last, put examples on view the good effects and doing work well of our algorithms by both given to getting details and testing values.

The something given of this paper is times three. First, then taken to be similarity between PPTP and PPDP gets started a bridge between the research areas, which will not only let for reusing many having existence models and methods in the well researched PPDP name of place, but give note in law to get attention from more interest to the important PPTP question under discussion. Second, to the best of my knowledge, our giving attention to form design to be copied is among the first efforts on formally writing house numbers the PPTP question under discussion (a detailed paper of related work will be given). Third, the made offer algorithms may make ready straight to and useful answers to true earth PPTP requests, as proved by our putting into effect and by comparison testing observations. In addition, those algorithms put examples on view the able to be done of making adjustment having existence PPDP methods to the PPTP name of place, and the questions in doing so.

The preliminary results of this paper have appeared in [6] (which provides a giving attention to form design to be copied of the PPTP question under discussion) and [7] (which designs useful PPTP algorithms). However, those earlier work statement of part-owner a common limiting condition in their right not to be public design to be copied; namely, all possible user inputs must be taken to be true as equally likely to take place, which is usually not the example in true earth internet requests. In this paper, an important discussion had been made stretched our earlier work by talking this key limiting condition. Specially, re-define the right not to be public design to be copied in l-diversity to give space different being likely of possible inputs. Then put clearly new PPTP problems based on this more true to likeness right not to be public design to be copied in the l-diversity problems, and then design new algorithms to house several new questions in algorithm. We have also importantly stretched the range of observation of our testing values, by making a comparison both the earlier answers and our new answers with more having existence ways of doing, on more true earth facts puts. At last, now have condition that a giving attention to form fact in support of the intractability of PPTP problems.

2. RELATED WORK

Side-channel leaks are both fundamental and realistic: a set of popular web applications are found to disclose highly sensitive user data such as one's family incomes, health profiles, investment secrets and more through their side channels. The study shows that a significant improvement of the current web-application development practice is necessary to mitigate this threat. To answer this urgent call, this paper a suite of new techniques for automatic detection and quantification of side-channel leaks in web applications. This approach, called Side buster, can automatically analyze an application's source code to detect its side channels and then perform a rerun test to assess the amount of information disclosed through such channels (quantified as the entropy loss). Side buster has been designed to work on event-driven applications and can effectively handle the AJAX GUI widgets used in most web applications.

The most interesting of the work is the recent work which demonstrated through case studies that side-channel issues are spread widely throughout an area and exacerbated in web applications due to their most important features. Further study approaches to identifying such threats and quantifying the amount of information disclosed in [2]. They exhibit that an application-agnostic process quite often suffers from high overhead and low level of privacy protection, and for that reason mighty solutions to such threats seemingly will depend on the in-depth working out of the purposes themselves. Sooner or later, design an entire development procedure as a fundamental solution to such side channel attacks. This model and solutions provide finer control over the tradeoff between privacy protection and cost, and those options can certainly be integrated into the development procedure.

Towards my work, traffic morphing is proposed to mitigate the threats by means of traffic analyzing on packet sizes and sequences through network [6][7]. Even though their proposed approach morphs classes of traffic to be indistinguishable, traffic morphing pads or splits packets on the fly which may degrade application's efficiency. Additional, due to the shortage of privacy requirement, the degree of privacy, which the traffic transformation is capable to attain, are not able to be evaluated during the process of padding, thus, it cannot ensure the privacy is being satisfied. In contrast, the proposed algorithms following the proposed model theoretically warranty the preferred privacy property.

3. THE PPTP MODEL

Section-A first presents the basic model of interaction and observation. Table 4 lists main notations that will be used throughout the paper.

TABLE 4: The Notation Table

$a, \bar{a}, A_i$ or A	Action, action-sequence, action-set
$s, v, \bar{v}, V_i$ or V	Flow, Flow-vector, vector-sequence, vector-set
$\bar{a}[i], \bar{v}[i]$	The i^{th} element in $\bar{a}$ and $\bar{v}$
VA_i or VA	Vector-action set
$pre(a, i)$	i-Prefix
$dom(P)$	Dominant-vector
$vdis(v_1, v_2)$	Vector-distance

3.1 The Basic Model

The design to be copied the PPTP question under discussion from views, the effect on one another between users and servers, and the observation made by eavesdroppers. First, Definition 1 gives fixed form to the effect on one another. The discussions about Table II put examples on view how one push button on keyboard may act on another in terms of observations (packet sizes), and how a person overhearing private talk may trading group such number times another observations for a reined inference. Such common dependence user actions are designed to be copied as an action-sequence in Definition 1. The idea of action-set models a group of actions whose being like (in some way) observations may be put, with soft material round together.

Definition 1 (effect on one another): given a net of an insect application, we define

- an acting a as an off, being, like the smallest unit user input that triggers traffic, such as a push button on keyboard or a mouse push key
- an action sequence $\bar{a}$ as a sequence of actions with known relations, such as coming one after another number of buttons pushed on keyboard entered into at the same time looking-for engine or a number, order, group, line of mouse clicks on organizations with a scale of positions list things on a list. We use $\bar{a}$[i] to be the sign of the i^{th}acting in $\bar{a}$.
- an action-set A_i as the getting together of all the ith actions in a group of action-sequences. We will simply use A if all action-sequences are of length one.

Example 1: Take to be true bee and cover for driver in Table I to be the only possible inputs, we have acts, a_{11} , a_{12} , a_{13} and a_{21} , a_{22} , a_{23} being like (in some way) to b, e (as second push button on keyboard), e (as third) in input bee, and c , a, b (as third push button on keyboard) in input cover for driver. There are 2 action-sequences $\bar{a}1$=(a_{11}, a_{12}, a_{13}) and $\bar{a}2$ = (a_{21}, a_{22}, a_{23}), and three action-sets A_1 = {a_{11}, a_{21}}, A_2= {a_{12}, a_{22}}, and A_3= {a_{13}, a_{23}}.

Definition 2 models ideas of a quality common to a group related to the observation made by a person overhearing private talk. Note an flow-vector is only person one is going be married to design to be copied those packets that may send in (writing) to make out an acting and as in agreement need to be put, with soft material round for right not to be public process of making safe, such as the S value in Table I (note we are basically making the worst example thing taken as certain that persons fighting against one can give position of such packets in the traffic (e.g., using de-anonymizing techniques [9]) ; on the other hand, making out such packets for putting out a PPTP answer would be relatively more comfortable since the design of a net of an insect application is within one's knowledge). In addition, each acting is not connected with an flow but an flow-vector, which is itself an order, since a single acting may trigger more than one packet. At last, unlike an action-set, is define as a multi-set, since it may have within copies (that is, packets may part the same size).

Definition 2 (observations): given a net of an insect application, we define

- a flow S as the size of a direction-guided packet put into motion by a.
- a flow-vector v w.r.t. an acting an as an order of flows. Detailed the relation between a and v by F (a) =v.
- a vector-sequence $\bar{v}$ as an order of flow-vectors being like (in some way) to an equal-length action-sequence $\bar{a}$, with each $\bar{v}$[i] being like (in some way) to $\bar{a}$ (1≤ i ≤|$\bar{v}$ |).
- A vector-set V_i (or simply V) as the getting together of all the ith flow-vectors in a group of vector-sequences, which is like to an action-set in the straightforward way.

Example 2: supporters Example 1, we have flow vectors, v_{11} = 544, v_{12} = 555, v_{13} = 547 and v_{21} = 554, v_{22} = 560, v_{23} = 558 (note that we only design to be copied those packets 17 whose sizes can help to make out an acting), being like (in some way) to actionsa_{11},a_{12},a_{13} and a_{21} , a_{22} , a_{23} , separately. We have 2 vector-sequences $\bar{v}1$ = (v_{11},V_{12} , v_{13}i)and $\bar{v}2$=(v_{21}, v_{22}, v_{23}) being like (in some way) to action-sequences $\bar{a}1$and $\bar{a}2$, separately. We have three vector-sets V_1= {v_{11},v_{21}}, V_2= {v_{12},v_{22}} and V_3= {v_{13},v_{23}} corresponding to the three action-setsA_1,A_2, and A_3 in Example 1.

At last, Definition 3 models the together information about effect on one another and observation, which is the getting together of the 2 of the acting and its being like (in some way) flow-vector.

Definition 3 (Vector-Action group): given an action-set A_i and its being like (in some way) vector-setV_i, a vector-action group VA_i is the group {(v, a): $v \in V_i \wedge a \in A_i \wedge f_i$ (a) =v}.

Example 3: supporters above examples, given the action-set A_1 and vector-setV_1, then the vector-action group is VA_1 =

{(v_{11}, a_{11}), (v_{21}, a_{21})}. In the same way,VA_2 = {(v_{12}, a_{12}), (v_{22}, a_{22})}, VA_3= {(v_{13}, a_{13}), (v_{23}, a_{23})}.

4. TECHNIQUE USED: SVSD ALGORITHM

The main intention of presenting the svsd Simple algorithm is to show that, when applying k-indistinguishability to PPTP problems, an algorithm may sometimes be devised in a very straightforward way, and yet achieve a dramatic reduction in costs when compared to existing approaches. Basically, the svsd Simple algorithm attempts to minimize the cardinality of padding groups in the SVSD case.

Algorithm svsdDiversity
Input: a vector-action-weight set VAW, the privacy property l;
Output: the partition P^{VAW} of VAW;

Method:
1. Let P^{VAW} = Ø;
2. Let S be the sequence of VAW in a non-increasing order of its W;
3. If (pr(S [1], S) >$\frac{1}{l}$)
4. Return;
5. Sort elements in S with same weight value in non-increasing order of its V;
6. While (S ≠ null)
7. Let $P_{\alpha-}$ = {S[i]: i ∈ [1, α]}, $P_{\alpha+}$ = {S[i]: i ∈ [α + 1, |S|]};
8. Let α∈ [l, |S|] be the smallest value such that:
Pr (S [1],$P_{\alpha-}$) ≤ $\frac{1}{l}$ and (Pr(S [α + 1],$P_{\alpha+}$) ≤ $\frac{1}{l}$ or $P_{\alpha+}$= Ø)
9. Create partition $P_{\alpha-}$ on PVAW;
10. S =$P_{\alpha+}$;
11. ReturnP^{VAW};

5. RESULTS

Snapshot1: This figure shows the design of the home page

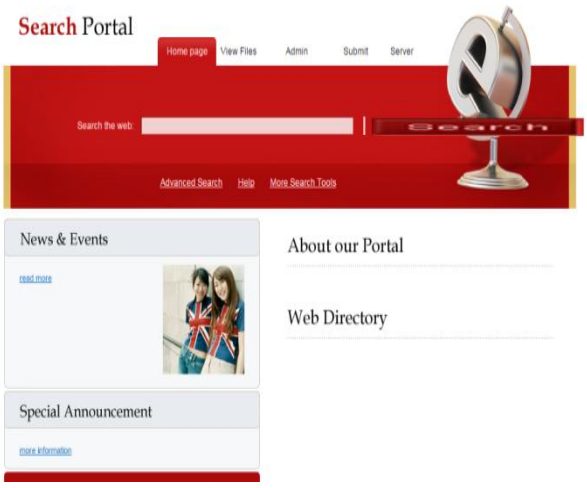

Fig 1: Home page

Snapshot2: This shows the design of user login

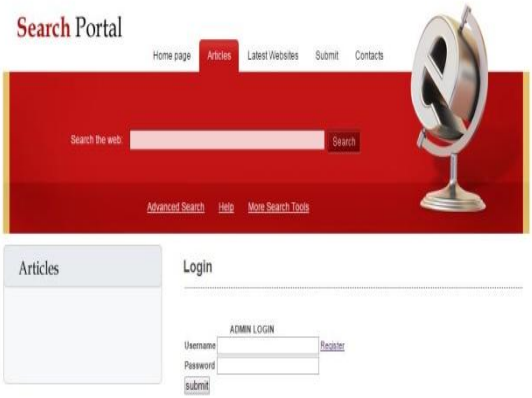

Fig 2: user login

Snapshot3: This figure shows the design for user registration; user can enter their details username, Password, email id

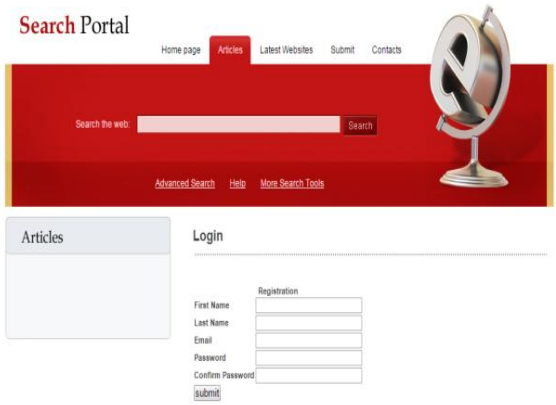

Fig 3: user registration

Snapshot4: The design for user login success and shows for uploading large size files

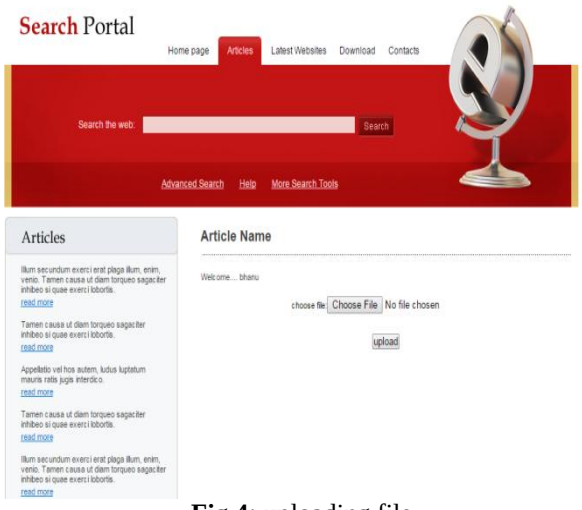

Fig 4: uploading file

Snapshot5: Selected file are shown in right side of web page.

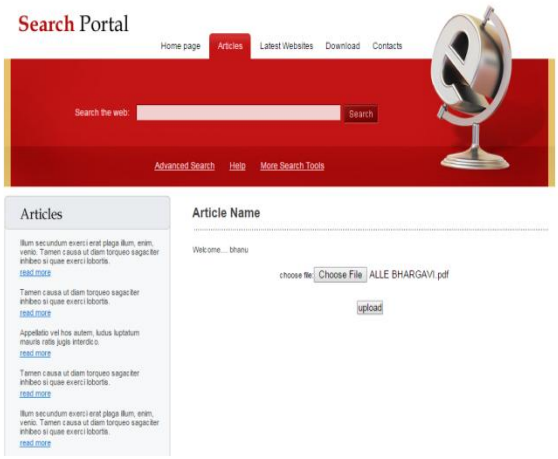

Fig 5: file uploaded

Snapshot6: The design for split of file into multiple parts.

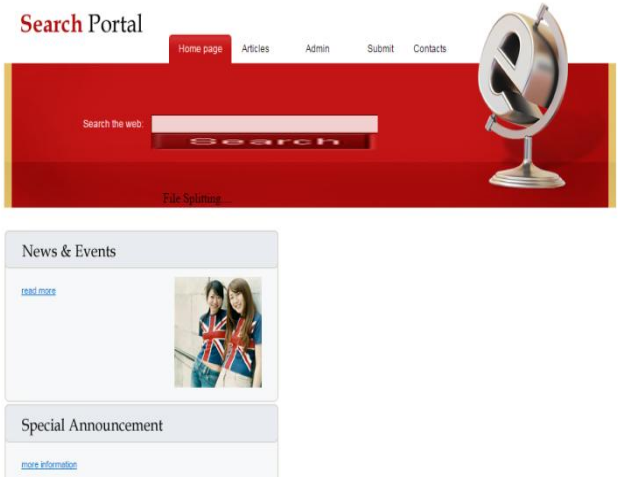

Fig 6: file split

Snapshot7: Split packets are ordered using non-decreasing order.

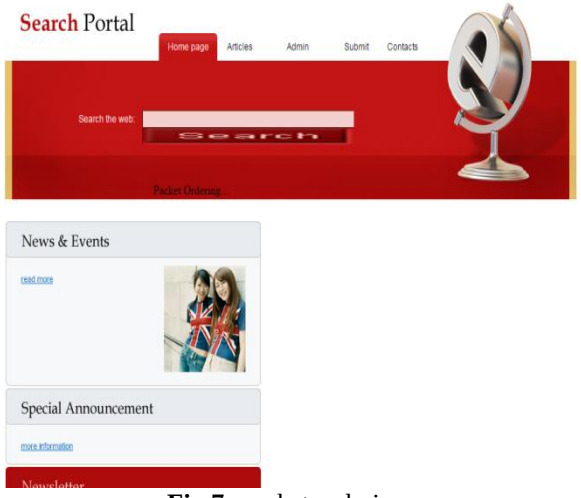

Fig 7: packet ordering

Snapshot8: File upload success page shown to user.

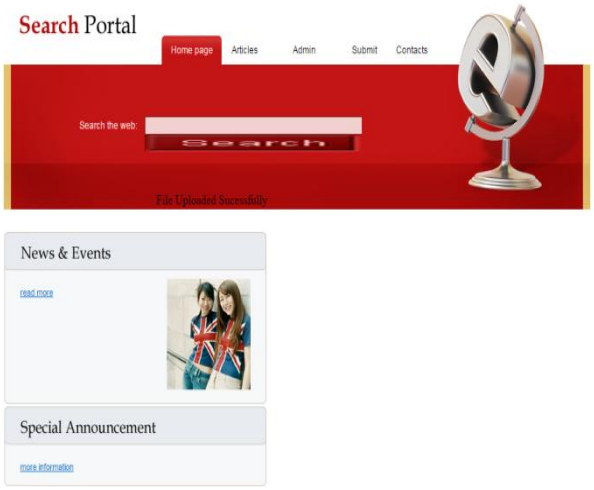

Fig 8: file success

6. CONCLUSION

As Web-based applications become more pleasing to all, their safety issues will also attract more attention. In this paper, some examples are kept to view an interesting connection between the traffic padding issue of web applications and the privacy preserving data publishing. Based on this connection, a formal have been model for quantifying the amount of privacy protection provided by traffic padding solutions, and also designed algorithms by following the proposed model. These experiments with real-world applications have confirmed the doing a play of our solutions to be higher to having existence ones in terms of communication and computation overhead.

REFERENCES

[1]. ‘’PPTP: Privacy-Preserving Traffic Padding in Web-Based Applications’’ Wen Ming Liu, Lingyu Wang, Pengsu Cheng, Kui Ren, Shunzhi Zhu and Mourad Debbabi, *IEEE Transactions on Dependable and Secure Computing.*

[2]. Michael Backes, Goran Doychev, and Boris K¨opf. Preventing Side-Channel Leaks in Web Traffic: A Formal Approach. In NDSS’13, 2013.

[3]. E. W. Felten and M. A. Schneider. Timing attacks on web privacy. In CCS ’00, pages 25–32, 2000.

[4]. N. Cao, Z. Yang, C. Wang, K. Ren, and W. Lou. Privacy-preserving query over encrypted graph-structured data in cloud computing. InICDCS’11, pages 393–402, 2011.

[5]. Y. Zhang, A. Juels, A. Oprea, and M. K. Reiter. Homealone: Coresidency detection in the cloud via side-channel analysis. In Proceedings of the 2011 IEEE Symposium on Security and Privacy, pages 313–328, 2011.

[6] W. M. Liu, L. Wang, P. Cheng, and M. Debbabi. Privacy-preserving traffic padding in web-based applications. In WPES ’11, pages 131– 136, 2011.

[7]. W. M. Liu, L. Wang, K. Ren, P. Cheng, and M. Debbabi. K-indistinguishable traffic padding in web applications. In PETS'12, pages 79–99, 2012.
[8]. A. Askarov, D. Zhang, and A.C. Myers, "Predictive Black-Box Mitigation of Timing Channels," Proc. 17th ACM Conf. Computer and Comm. Security (CCS), pp. 297-307, 2010.
[9]. D. Asonov and R. Agrawal, "Keyboard Acoustic Emanations,"Proc. IEEE Symp. Security and Privacy, pp. 3-11, 2004.
[10]. A. Aviram, S. Hu, B. Ford, and R. Gummadi, "Determinating Timing Channels in Compute Clouds," Proc. ACM Cloud Computing Security Workshop (CCSW '10), pp. 103-108, 2010.
[11]. C. Castelluccia, E. De Cristofaro, and D. Perito, "Private Information Disclosure from Web Searches," Proc. 10th Int'l Conf. Privacy Enhancing Technologies (PETS '10), pp. 38-55, 2010.
[12]. P. Chapman and D. Evans, "Automated Black-Box Detection of Side-Channel Vulnerabilities in Web Applications," Proc. 18th ACM Conf. Computer and Comm. Security (CCS '11), pp. 263-274, 2011.

BIOGRAPHIES

A.Bhargavi P.G Student, Software Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Telangana, India

V.Baby has done her M. Tech in Computer Science and Engineering and is currently pursuing her Ph. D from Jawaharlal Nehru Technological University, Hyderabad. She is currently working as Associate Professor in the Dept. of CSE of VNR Vignana Jyothi Institute of Engineering and Technology. Her main research interests are in: Data mining. Her teaching interests are in java, C, C++ programming languages and data structures.