複数認証可能なワンタイムパスワード生成器の提案と評価

足立 周介 † 伊沢 亮一 ‡ 森井 昌克 †

† 神戸大学大学院工学研究科
657-8501 神戸市灘区六甲台町 1-1

‡ 神戸大学大学院自然科学研究科
657-8501 神戸市灘区六甲台町 1-1

{adachi@stu., mmorii@}kobe-u.ac.jp

isawa@stu.kobe-u.ac.jp

あらまし パスワード認証の際のユーザの負担を軽減する目的として複数認証エージェントを用いたワンタイムパスワード認証方式が提案されている．これらの方式はユーザの利便性の向上にはつながるものの，サーバの情報が漏えいした際に，攻撃者に簡単にユーザになりすまされる等，その安全性に関する議論は未だ十分とは言い難い．本稿ではサーバの情報が漏えいしたとしても，攻撃者にユーザまたはエージェントになりすまされない複数認証可能なワンタイムパスワード認証方式を提案する．提案方式ではワンタイムパスワードをユーザ固有の固定情報から作り出すのではなく，その固有情報もまた各セッション毎に更新する．これにより，ユーザは認証の際の利便性と安全性を両立することが可能となる．

Proposal of Single One-Time Password Generator Applicable for Multi-Authentication and its Evaluation

Shusuke Adachi† Ryoichi Isawa ‡ Masakatu Morii†

†Graduate School of Engineering, Kobe University
1-1 Rokkodai, Nada-Ku, Kobe-Shi, 657-8501 Japan

{adachi@stu., mmorii@}kobe-u.ac.jp

‡Graduate School of Science and Technology, Kobe University
1-1 Rokkodai, Nada-Ku, Kobe-Shi, 657-8501 Japan

isawa@stu.kobe-u.ac.jp

Abstract To reduce users' burden, some single one-time password authentication methods applicable for multi-authentication agents has been proposed. These methods are convenience for users but the security is risky when the information server stores is stolen. In this paper, we propose a single one-time password method applicable for multi-authentication agents and is tolerate even if the server's information is leaked. Our method gives high security because it generates the authentication information by using the updated disposable private key at each session. This will increase security and will be convenient for users.

1 Introduction

Recently, the way to authenticate users over the communication is indispensable according to the population of electronic commerce. There are many services that needs to authenticate users over the Internet. Well-known tools for the authentication systems are public-key and symmetric-key cryptosystems and one-way hash function. Since its security level is so high, the public-key cryptosystem can be applied to a wide range of area. The required computational cost is however very high. Considering the users' resources of low performance terminal like smart cards, it is reasonable to apply the symmetric-key cryptosystem and one-way

functions which computational cost is much lower than public-key cryptosystem.

There are stability password and biometric authentication systems. Especially the password system that changes every session called one time password authentication is a safe way and many kinds of methods are proposed[1, 2, 3].

Usually, users access to many kinds of services over the Internet, and they have to hold respective passwords to every services. The preservation of such passwords is tiresome for them. One solution to authenticate the user with only one password is to employ an authentication server to which the servcie provider requests the user's authentication like Open ID[4]. On the other hand, Tuji et al. proposed a method called SAIFU[5] to employ agents between the user and the authentication server. Considering the server's load when many users request for authentication, the latter method can reduce it. But the SAIFU method requires many communication when renewaling the authentication information between agents and the server. Also, SAIFU is supposing a secure channel between agents and the server so the scalability is low. We have proposed a method that can reduce the communication cost and can exclude the secure channel in authentication process[6, 7, 8].

In addition most recently, protection of private information is strongly demanded. In order to minimize the damage caused by leakage of information from servers, systems that tolerate the Stolen-Verifier attack are desired.

We have proposed a one-time password method against the SV attack[9]. Based on this method, we propose a single one-time password usable for multi-authentication agents and is against the SV attack. Our method use a disposable private key to generate the authentication information. The private information users use to registrate will be used only to generate the disposable private key so it will not remain or go on the communication.

2 Definitions

2.1 One-Time Password

In this paper, we define one-time password authentication method as the password that go through the communication channel and the secret information that termial holds changes every authentication session.

2.2 Notations

We define the symbols as bellow in this paper.

U_n	nth user.
A_m	mth authentication agent.
ID_{U_n}	U_n's ID.
ID_{A_m}	A_m's ID.
S	authentication server.
$K_{U_n}(i)$	U_n's secret information in ith authentication session that only U_n holds.
$K_{A_m}(j)$	A_m's secret information in jth authentication session that only A_m holds.
$F_{U_n}(i)$	U_n's authentication information in ith authentication session that U_n and S holds.
$F_{A_m}(j)$	A_m's authentication information in jth authentication session that A_m and S holds.
SI_{U_n}	the secret information that $U_n$use to registrate initial value $K_{U_n}(1)$.
SI_{A_m}	the secret information that $A_m$use to registrate initial value $K_{A_m}(1)$.
$V_{U_n}(i)$	the information S holds to verify the received information from U_n.
$V_{A_m}(j)$	the information S holds to verify the received information from A_m.
$R_Z(y)$	the pseudorandom number Z generates in the yth session.
h	one way hash function. $h(message)$ means $message$ is hashed once.
$\oplus$	represents a bitwise exclusive OR(XOR) operation.
$\|$	represents concatenation
$A \longrightarrow B{:}X$	A sends X to B in a common channel.
$A \Longrightarrow B{:}X$	A sends X to B in a secure channel.

2.3 Well-Known Attacks

We define the well-known attacks as below.

Man-In-The-Middle(MIM) attack

An attacker intercepts the data between U_n and A_m or between A_m and S, and alter it to make connections with A_m or S.

Replay attack

Assume that an attacker intercepts the past session's data between U_n and A_m or between A_m and S. Retransmitting these data, an attacker makes connections with A_m or S.

Denial-of-Service(DoS) attack

Assume that an attacker intercepts the past session's data between U_n and A_m or between A_m and S. Using these data, an attacker alters the data between U_n and A_m or between A_m and S and make A_m or S receive it. U_n or A_m cannot request for connections to S from the next session.

Impersonation attack

Assume that an attacker intercepts the past session's data between U_n and A_m or between A_m and S. Using these data, an attacker alters the data between U_n and A_m or between A_m and S, and make connnections with A_m or S.

Stolen-Verifier(SV) attack

Assume that an attacker gets the informataion saved on the server. In addition, an attacker intercepts the data between U_n and A_m or between A_m and S. Using these data, an attacker alters the data between U_n and A_m or between A_m and S, and make connnections with A_m or S.

According to the definitions above, the Impersonation attack includes the MIM attack and the Replay attack. In addition, the SV attack includes the Impersonation attack. The method against the SV attack is against the MIM attack, the Replay attack and the Impersonation attack. In conclusion, the method against the DoS attack and the SV attack is against all well-known attacks above.

3 Conventional Methods

To authenticate users with only one password to several services, there are two methods. One method is to employ an authentication server and the servcie provider requests the user's authentication to it like Open ID[4]. Another method is to employ authentication agents between the user and the authentication server. Considering server's load when many users request for authentication, the latter method can reduce it. There is SAIFU[5] which is using the latter method. But this method requires a lot of communication because the server sends the user's next session authentication information to all agents after every user's authentication session is completed. In addition, this renewaling process is supposed under secure channel so the scalability is low. We have proposed a method that can reduce communication and assumed to use public channel in authentication process[6, 7, 8]. But either methods are not against the SV attack.

Recently, the information saved on the server leaking is a big problem. A method which cannot make connections with the server even if the information saved on the server is stolen, is important and required.

4 Proposed Method

In this section, we propose a method of single one-time password using authentication agents. In addition, our method is against all well-known attacks including SV attack. Also, our proposed method's authentication information not only changes but also the secret information that users and agents hold changes every authentication session. Our method is divided into registration phase and authentication phase. Registration phase is executed only once. Authentication phase is executed every time the user log on.

4.1 Registration phase

In the user registration process, U_n registers their own informations to S. In the agent registration process, A_m registers their own informations to S. Fig. 1 shows User registration process. Fig. 2 shows Agent registration process.

User registration

U_n registers their own informations to S as below.

Step 1 $U_n \longrightarrow S$:$Request$, ID_{U_n}.

Step 2 S generates $R_{S1}(x)$, $R_{S2}(x)$

Step 3 $S \Longrightarrow U_n$:$R_{S1}(x)$, $R_{S2}(x)$.

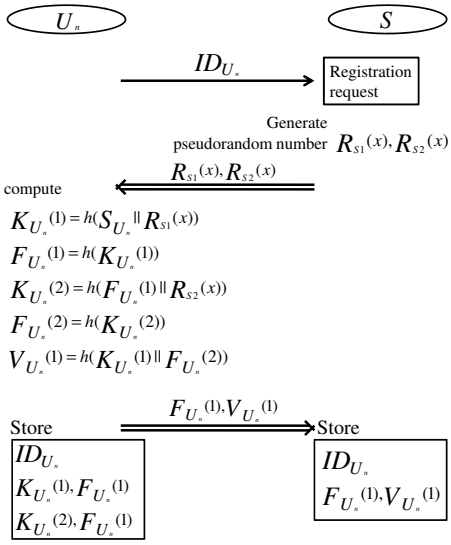

Fig. 1: User registration process.

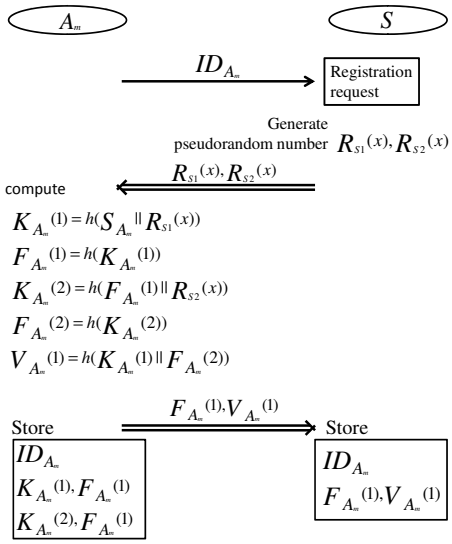

Fig. 2: Agent registration process.

Step 4 U_n calculates
$K_{U_n}(1) = h(SI_{U_n} \| R_{S1}(x))$
$F_{U_n}(1) = h(K_{U_n}(1))$
$K_{U_n}(2) = h(F_{U_n}(1) \| R_{S2}(x))$
$F_{U_n}(2) = h(K_{U_n}(2))$
$V_{U_n}(1) = h(K_{U_n}(1) \| F_{U_n}(2))$
and stores
ID_{U_n}, $K_{U_n}(1)$, $F_{U_n}(1)$, $K_{U_n}(2)$, $F_{U_n}(2)$

Step 5 $U_n \Longrightarrow S$:$F_{U_n}(1)$, $V_{U_n}(1)$.

Step 6 S stores ID_{U_n}, $F_{U_n}(1)$, $V_{U_n}(1)$.

Agent registration
A_m registers their own informations to S as below.

Step 1 $A_m \longrightarrow S$:$Request$, ID_{A_m}.

Step 2 S generate $R_{S1}(x)$, $R_{S2}(x)$

Step 3 $S \Longrightarrow A_m$:$R_{S1}(x)$, $R_{S2}(x)$.

Step 4 A_m calculates
$K_{A_m}(1) = h(SI_{A_m} \| R_{S1}(x))$
$F_{A_m}(1) = h(K_{A_m}(1))$
$K_{A_m}(2) = h(F_{A_m}(1) \| R_{S2}(x))$
$F_{A_m}(2) = h(K_{A_m}(2))$
$V_{A_m}(1) = h(K_{A_m}(1) \| F_{A_m}(2))$
and stores
ID_{A_m}, $K_{A_m}(1)$, $F_{A_m}(1)$, $K_{A_m}(2)$, $F_{A_m}(2)$

Step 5 $A_m \Longrightarrow S$:$F_{A_m}(1)$, $V_{A_m}(1)$

Step 6 S stores ID_{A_m}, $F_{A_m}(1)$, $V_{A_m}(1)$

4.2 Authentication phase

We explain the authentication phase. Fig. 3 shows ith authentication session of U_n and jth authentication session of A_m. The protocol of the authentication phase is as follows.

Step 1 U_n calculates $h(ID_{U_n} \| F_{U_n}(i))$.

Step 2 $U_n \longrightarrow A_m$:ID_{U_n}, $h(ID_{U_n} \| F_{U_n}(i))$.

Step 3 A_m calculates
$h(ID_{A_m} \| F_{A_m(j)} \| h(ID_{U_n} \| F_{U_n}(i)))$ using received $h(ID_{U_n} \| F_{U_n}(i))$.

Step 4 $A_m \longrightarrow S$:ID_{U_n}, ID_{A_m}, $h(ID_{A_m} \| F_{A_m}(j) \| h(ID_{U_n} \| F_{U_n}(i)))$.

Step 5 S verifies ID_{U_n}, ID_{A_m} from received $h(ID_{A_m} \| F_{A_m(j)} \| h(ID_{U_n} \| F_{U_n}(i)))$ and stored$F_{U_n}(i)$, $F_{A_m}(j)$.Then S calculates $h(F_{A_m}(j) \| F_{U_n}(i) \| V_{U_n}(i))$.

Step 6 $S \longrightarrow A_m$:$F_{A_m}(j) \oplus F_{U_n}(i)$, $F_{A_m}(j) \oplus V_{U_n}(i)$, $h(F_{A_m}(j) \| F_{U_n}(i) \| V_{U_n}(i))$.

Step 7 A_m computes $F_{U_n}(i)$ from received $F_{A_m}(j) \oplus F_{U_n}(i)$. Then computes $V_{U_n}(i)$ from recevied $F_{A_m}(i) \oplus V_{U_n}(i)$.Then verify it from received $h(F_{A_m}(j) \| F_{U_n}(i) \| V_{U_n}(i))$.

Step 8 $A_m \longrightarrow U_n$:$F_{U_n}(i) \oplus V_{U_n}(i)$

Step 9 U_n authenticate A_m from received $F_{U_n}(i) \oplus V_{U_n}(i)$. Then U_n generates $R_{U_n}(i)$. Then calculates $K_{U_n}(i+2) = R_{U_n}(i)$, $F_{U_n}(i+2) = h(K_{U_n}(i+2))$, $V_{U_n}(i+1) = h(K_{U_n}(i+1) \| F_{U_n}(i+2))$ and $h(V_{U_n}(i+1))$.

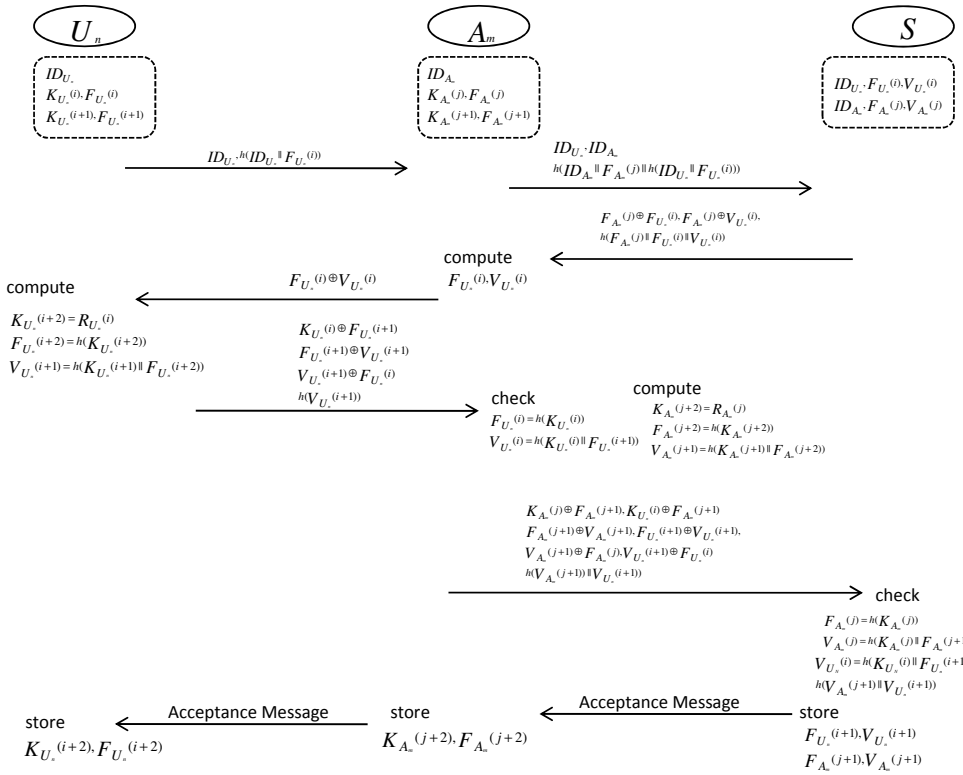

Fig. 3: authentication proccess.

Step 10 $U_n \longrightarrow A_m$:$K_{U_n}(i) \oplus F_{U_n}(i+1)$, $F_{U_n}(i+1) \oplus V_{U_n}(i+1)$, $V_{U_n}(i+1) \oplus F_{U_n}(i)$ $h(V_{U_n}(i+1))$.

Step 11 A_m computes $V_{U_n}(i+1)$ from $V_{U_n}(i+1) \oplus F_{U_n}(i)$, $F_{U_n}(i+1)$ from $F_{U_n}(i+1) \oplus V_{U_n}(i+1)$ and $K_{U_n}(i)$ from $K_{U_n}(i) \oplus F_{U_n}(i+1)$.Then verify if $F_{U_n}(i) = h(K_{U_n}(i))$, $V_{U_n}(i) = h(K_{U_n}(i) \parallel F_{U_n}(i+1))$. Also verify $h(V_{U_n}(i+1))$. Then A_m authenticate U_n. A_m generates $R_{A_m}(j)$. Then calculates $K_{A_m}(j+2) = R_{A_m}(j)$, $F_{A_m}(j+2) = h(K_A(j+2))$, $V_{A_m}(j+1) = h(K_{A_m}(j+1) \parallel F_{A_m}(j+2))$ and $h(V_{A_m}(j+1) \parallel V_{U_n}(i+1))$.

Step 12 $A_m \longrightarrow S$:$K_{A_m}(j) \oplus F_{A_m}(j+1)$, $F_{A_m}(j+1) \oplus V_{A_m}(j+1)$, $V_{A_m}(j+1) \oplus F_{A_m}(j)$, $K_{U_n}(i) \oplus F_{U_n}(i+1)$, $F_{U_n}(i+1) \oplus V_{U_n}(i+1)$, $V_{U_n}(i+1) \oplus F_{U_n}(i)$, $h(V_{A_m}(j+1) \parallel V_{U_n}(i+1))$.

Step 13 S computes $V_{A_m}(j+1)$ from $V_{A_m}(j+1) \oplus F_{A_m}(j)$, $F_{A_m}(j+1)$ from $F_{A_m}(j+1) \oplus V_{A_m}(j+1)$ and $K_{A_m}(j)$ from $K_{A_m}(j) \oplus F_{A_m}(j+1)$. Next verify if $F_{A_m}(j) = h(K_{A_m}(j))$, $V_{A_m}(j) = h(K_{A_m}(j) \parallel F_{A_m}(j+1))$. Then S authenticates A_m and understands that U_n and A_m succeeded mutual authentication. Also, S computes $V_{U_n}(i+1)$ from $V_{U_n}(i+1) \oplus F_{U_n}(i)$, $F_{U_n}(i+1)$ from $F_{U_n}(i+1) \oplus V_{U_n}(i+1)$ and $K_{U_n}(i)$ from $K_{U_n}(i) \oplus F_{U_n}(i+1)$. Next check if $V_{U_n}(i) = h(K_{U_n}(i) \parallel F_{U_n}(i+1))$. Also verify $h(V_{A_m}(j+1) \parallel V_{U_n}(i+1))$. S reduces the stored information and stores the next session's authentication information $F_{U_n}(i+1)$, $V_{U_n}(i+1)$, $F_{A_m}(j+1)$ and $V_{A_m}(j+1)$.

Step 14 $S \longrightarrow A_m$:*Acceptance Message.*

Step 15 A_m renews the stored $K_{A_m}(j)$, $F_{A_m}(j)$, $K_{A_m}(j+1)$ and $F_{A_m}(j+1)$ to $K_{A_m}(j+1)$, $F_{A_m}(j+1)$, $K_{A_m}(j+2)$ and $F_{A_m}(j+2)$.

Step 16 $A_m \longrightarrow U_n$:*Acceptance Message.*

Step 17 U_n renews the stored $K_{U_n}(i)$, $F_{U_n}(i)$ $K_{U_n}(i+1)$ and $F_{U_n}(i+1)$ to $K_{U_n}(i+1)$, $F_{U_n}(i+1)$, $K_{U_n}(i+2)$ and $F_{U_n}(i+2)$.

5 Security Evaluation

In this section, we will describe the security of attacks to our proposal method. As we de-

scribed in section 2, if the method is against the SV attack it is also against the MIM attack, the Replay attack and the Impression attack. So we will describe that our proposal method is against the DoS attack and the SV attack.

DoS attack

If the server don't accept the altered communication data, the method is against the DoS attack. In the ith session, the data flowed in the communication channel are Step 2, 4, 6, 8, 10, 12. At this time, S holds data F_{U_n}, V_{U_n}, F_{A_m}, V_{A_m} to verify the recieved data. These data don't go on the communication channel and the attacker can't get these data and can't alter the communication data. So the attacker cannot make the server accept altered data and it means this proposal method is against the DoS attack.

SV attack

If the attacker cannot spoof as U_n and A_m from getting the information saved on the server and using the communication data flowed in the past session, the method is against the SV attack. If the attacker can get $K_{U_n}(i)$ and $K_{A_m}(j)$ before the ith session, attacker can spoof as U_n or A_m. At Step10, the attacker can get $K_{U_n}(i)$ from using $F_{U_n}(i)$ saved on the server. But at Step13, the server authenticates A_m first by $K_{A_m}(j)$ and then authenticate $U_n(i)$ by $K_{U_n}(i)$. The attacker can't get $K_{A_m}(j)$ so the attacker can't be authenticated.

After U_n's $i-1$th session, the attacker can get $F_{U_n}(i)$ from the information saved on the server. At ith session, using F_{U_n}(i) the attacker can alter $V'_{U_n}(i+1) = h(K'_{U_n}(i+1) \parallel F'_{U_n}(i+2))$ in the communication channel Step10 and make the server accept it. But in the $i+1$th session, the server can verify $V'_{U_n}(i+1)$ to the true $V_{U_n}(i+1)$ which can be computed by $K_{U_n}(i+1)$ and $F_{U_n}(i+2)$, the information that only the real user holds before this session begins. That means the attacker can alter the information using the information saved on the server and the information flowed on the communication data but, cannot spoof as the user. The same thing can be said to agents. Therefore, this proposal method is against the SV attack.

6 Conclusion

In this paper, we have proposed a single one-time password generator applicable for multi-authentication that can tolerate the SV attack. In our method, we use a disposable secret key that will be renewed after each session so that the user's or the agent's secret information will not be stored on the terminal and will not go over the communication channel. We clarified that our method was robust against the DoS attack and SV attack.

Acknowledgements

We thank Dr. Minoru Kuribayashi and Dr. Ryoichi Teramura for useful comments.

References

[1] L. Lamport, "Password authentication with insecure communication," Commun. ACM, vol.24, no.11, pp.770–772, Nov. 1981.

[2] N. Haller, "The S/Key(TM) one-time password system," Proc. Internet Society Symposium on Network and Distributed System Security, pp.151–158, Feb. 1994.

[3] T. Tsuji, T. Kamioka, and A. Shimizu, "Simple And Secure password authentication protocol,ver.2(SAS-2)," IEICE Technical Report, OIS2002-30, vol.102, no.314, pp.7–11, Sep. 2002.

[4] "OpenID," `http://www.openid.ne.jp/`

[5] T. Tsuji and A. Shimizu, "Secure Agreement Identification for Flexible Users(SAIFU)" IEICE Technical Report, OIS2004-16, vol.104, no.238, pp.13–17,2004.

[6] K. Uo, T. Ohigashi, Y. Shiraishi and M. Morii, "A Single One-Time Password Method Usable by Multi-Authentication Agents" IPSJ SIG Technical Report vol.2005, no.41, pp.53–58, May 2005.

[7] K. Uo, T. Ohigashi, Y. Shiraishi and M. Morii, "Security Analysis for a Single One-Time Password Method Usable by Multi-Authentication Agents, " IEICE Technical Report OIS2005, Sept. 2005.

[8] K. Uo, T. Ohigashi, Y. Shiraishi and M. Morii, "A Single One-Time Password Method for Multi-Authentication Server and its Evaluation, " CSS2005, Oct. 2005.

[9] S. Hashimoto,R. Isawa and M. Morii, "Notes on A One-Time Password Method against the Stolen-Verifier Attack, " IEICE Technical Report OIS2009, June. 2009.