EXPERIMENTAL STUDY OF DIGIPASS GO3 AND THE SECURITY OF AUTHENTICATION

Igor Semaev

***Abstract*—Based on the analysis of 6-digit one-time passwords (OTP) generated by DIGIPASS GO3 we were able to reconstruct the synchronization system of the token, the OTP generating algorithm and the verification protocol in details necessary for an attack. The OTPs are more predictable than expected. A forgery attack is described. We argue the attack success probability is much higher than it may be expected if all the digits are independently and uniformly distributed. The implications for the security of authentication are discussed and open questions are formulated.**

***Keywords*— security of authentication; DIGIPASS; attack success probability**

I. INTRODUCTION

THE remote authentication is commonly two-way. It is a combination of static passwords and dynamic one-time passwords (OTP) generated by security tokens, read from one-time password card, or communicated by mobile devices.

Static passwords are at least theoretically may be tapped by malicious software as trojans with keystroke logging for example. Therefore the authentication security is largely based on the security of one-time passwords. The latter is insecure if the next OTP is predicted with non-negligible probability given a number of previous OTPs. For instance, 6-digit password may be predicted with probability 10^{-6} anyway. If it is possible to predict with a larger probability, then there is an internal weakness in the generator. The number of attempted wrong log-ins is usually bounded. In practical terms that makes the implementation of the attack more difficult, but does not make that impossible.

DIGIPASS is a one-time password generator manufactured by VASCO [1]. In what follows we study 6-digit combinations produced by DIGIPASS GO3 distributed by Norwegian Sparebanken Vest. It is also used in DNB NOR, which is one of the largest banks in Norway with 2.3 millions of retail customers and 198000 corporate customers. Besides Norway, according to [1], the token is used by lots of customers all over the world, for instance, in Belgium, Ireland, United Kingdom, Netherlands, Denmark, Saudi Arabia, South Africa etc. They say the device uses strong crypto algorithms as DES, TDES and AES and supports event and time based authentication. The detailed description of the algorithm was not published.

Igor Semaev was with Department of Informatics, University of Bergen, Bergen, Norway. (e-mail: igor@ii.uib.no).

This is an extended abstract of the paper put in the Cryptology ePrint Archive, see [6].

II. PUBLICLY AVAILABLE STATE OF ART

A. *Published One-Time Password Algorithms*

Two One-Time Password(OTP) algorithms are published in [2,3]. The first HMAC-based One-Time Password (HOTP) algorithm specifies event-based OTP algorithm. The other(TOTP) is an extension of the HOTP algorithm to support the time-based moving factor. We briefly describe the latter. The prover (token) and the verifier(authentication server) use the same time-step value X. There must be a unique secret key K for each prover. Therefore,

$$TOTP=HOTP(K,T)=Truncate(HMAC(\text{“crypto”}\},K,T)),$$

where T is an integer number which represents the number of time steps between the initial counter time T_0 and the current Unix time. The server should compare OTP not only with the receiving time-stamps but also the past time-stamps that are within a transition delay window, specified by the protocol. The function "Truncate" is specified in [2] for HMAC-SHA-1 as “crypto".

B. *Authentication in a Norwegian Bank*

There are 3 ways for a customer in Norwegian Sparebanken Vest to authenticate himself and get access to his account for internet banking. Two of three authentication protocols require an OTP from a token as DIGIPASS or from a one-time password card.

III. EXPERIMENTAL STUDY

The customer presses the token to generate an OTP. During the next 40 seconds this OTP is kept on the screen and then disappears. If one then presses the token in time between 40 and 50 sec. the same OTP reappears. If one presses in time > 50 sec. a new OTP may be generated. We study what happens if the token is pressed steadily at various time intervals. The experiments were produced by pressing the token and the stopwatch simultaneously. A very little fluctuation between the real pressing time and what was recorded is possible. All the experiments below may be easily repeated and verified.

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

The data below were produced with DIGIPASS GO3 (79-8456258-6). Some of the experiments were repeated for another DIGIPASS GO3 (28-4804970-8) with similar results.

TABLE 1

6-digit OTPs at time step 50 sec.				
240445	773743	220372	747108	141253
302168	818140	388126	747108	251821
498773	943630	388126	829372	370755
591327	943630	475446	904089	455412
642055	091211	534182	012530	455412
642055	147884	621339	141253	

We start by generating 6-digit combinations with the time step (period) of t=50 sec., see the above table. It contains 29 combinations. The first OTP 240445 indicates starting time 0 sec. , the next 302168 was generated in 50.35 sec., the next 498773 was generated in another 50.41 sec. and so on. Surprisingly, the token happens to repeat the combination. For instance, 642055 repeats in 50 sec. again etc. Even worse, one finds the 6-digit combination is predictable with probability 1 under some condition. Let the customer generate a sequence of 5 different OTP at time step 50 sec. Then the next OTP, after another 50 sec., is always a repetition of the last combination in the sequence. For instance, assume one observes the sequence 388126, 475446, 5344182, 621339,747108 as in the above table, then the next combination is again 747108.

That does not seem affect the security of the authentication as the server (verifier) does not accept repeated combination as correct, though they are legally produced. The sequence of OTPs in the table may be split into intervals, each interval ends with a repeated OTP. For 50 sec. time step the length of the intervals is 4 or 5. For t=51 sec. the length of the intervals is 5 or 6. For t=52 sec. the length of the intervals is 5 or 6 again, but the pattern is different. For t=53 sec. the length of the intervals is 6 or 7 and so on. Finally, for t=63 sec. the length of an interval may be larger than 67.

When pressing steadily at time step t between 50 and 63 sec. the first digit of the combination increases by 1 modulo 10 or it is the same and the whole combination repeats. In particular, the probability that the left most digit repeats is steadily decreasing while the time step is increasing. For t around 64 sec. the left most digit increases by 1 with probability very close to 1. When the pressing steadily at time step t between 64 and 127 sec. the first digit of the combination increases by 1 or 2 modulo 10. For t within 128 and 191 sec. the first digit of the combination increases by 2 or 3 modulo 10, etc. One now sees that the left most digit of the OTP is predictable with high probability if one knows time elapsed since the previous OTP was generated.

TABLE 2

	Distribution of decimal digits in the OTP last 5 positions									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
b	102	92	94	109	102	105	53	59	48	50
c	96	108	96	107	115	80	61	52	45	54
d	101	121	109	94	98	103	51	49	47	41
e	108	100	108	93	110	97	53	56	45	44
f	97	110	110	108	100	83	51	57	54	44

Let's have a look at the rest 5 digits. Let abcdef be a 6-digit combinations produced by a DIGIPASS GO3. According to the analysis in the next section, the first digit a is used for synchronization and it is predictable. We study the distribution of the rest digits b, c, d, e, f taken separately. 814 OTPs were generated in this study. We count the number of times a decimal digit appears in each of the last five positions in those OTPs. The data are collected in Table 2, where the positions are denoted by b, c, d, e, f. The distributions are not uniform. The digits 0, 1, 2, 3, 4, 5 appear twice more often on the average than 6, 7, 8, 9. Our explanation is the following. The last 5 digits are produced from a 20-bit string of pseudo-random data taken from the encryption function output: one decimal digit per each subsequent 4-bit string. The latter represents a number from 0, 1, …, 15 and therefore a decimal digit after reduction modulo 10. We may assume 4-bit strings are distributed uniformly, so the distribution of decimal digits is as 0, 1, 2, 3, 4, 5 have probability 1/8 and 6, 7, 8, 9 have probability 1/16. That fits well with the experimental data in the above Table 2. More data are shown in [6].

IV. ANALYSIS AND RECONSTRUCTION

In this section the collected data are analyzed. We reconstruct the algorithm implemented by the token, its synchronization and the verification protocol in details important for an attack presented below.

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

The smallest time interval used in the measurements and the computations by DIGIPASS and the server is one second. For t within 64a and <64(a+1) sec. the first digit of the combination increases by a or a+1 modulo 10. Each such interval incorporates 64 sec. Assume an OTP was generated and the next OTP was generated in t sec. The first digit of the combination may increase by a modulo 10 only if t within 64(a-1) and <64(a+1) sec. time interval. In the beginning of this time interval the first digit mostly increases by a-1. In the end it mostly increases by a+1 mod 10. In the middle it mostly increases by a mod 10. By symmetry, the probability of getting an increase by a modulo 10 within the above interval is 1/2. Those observations were supported by experiments with random time steps of around 10 min.

A. OTP generating algorithm

We now reconstruct OTP generating algorithm implemented by the token. Let a 6-digit OTP generated at time ti be ai,Xi, where ai is its left most digit and Xi, are its rest 5 digits. Also let T0 be some initial moment of time. Then OTP ai,Xi is generated by computing Ai= [(ti-T0)/64], and ai=Ai mod 10, and Xi=EK(Ai), where EK is an encryption based function which depends on a secret key K. By [..] a floor function is here denoted.

This algorithm is similar to one described in Section II and it fits well with the properties of DIGIPASS GO3 found in Section III. In particular, this reconstruction well explains OTPs repetition if they are generated subsequently at time ti and ti+1, where ti+1-ti is less than 64 sec.

B. Server(verifier) action in authentication.

We reconstruct the server(verifier) action in authentication. To log in the customer first introduces his identifier and static passwords into a pop up window on the monitor of his computer, presses the DIGIPASS, reads an OTP and introduces that into another pop up window. He then hits the return key on the computer keyboard. To verify the server is to solve the following three problems:

Handle the delay between generating the OTP and when the server gets it for authentication. Check if the OTP was produced by the token assigned to that customer. Assume the customer generates several OTPs without log in. At some later point one more OTP is generated by him to log in. The server should be able to authenticate the customer in that case.

Let t be the time of generating an OTP and t' the time it comes to the server for authentication. There should be an acceptable delay time interval t'-t< T. We found T is 480 sec. by the following experiment: an OTP was generated and then introduced into the system with a delay t'-t, and the server reaction was then observed. Interestingly, after the OTP was introduced with the delay 479 sec. and accepted, the synchronization between the token and the server got lost. A new token had to be used.

C. Verification protocol

We now reconstruct verification protocol. Let an OTP a,X, where a is the left most digit and X is the rest 5-digit combination, was generated at time t and came for verification at time t'. The verifier finds natural B=[(t'-T0)/64]. So as t'-t<480, then A=[(t-T0)/64] should be in the interval B-9,B-8,..,B. As A=a mod 10, the verifier computes A and hence X'=EK(A). The OTP a,X is accepted if X'=X. This reconstruction fits the above OTP generating algorithm and experimental data.

V.ATTACKS

We may assume customer's static passwords are known to the adversary, see the Introduction for an explanation. In what follows a basic algorithm to forge the dynamic password generated by a DIGIPASS GO3 is presented and the probability of its success is calculated.

A. Basic attack

To attack the account of one chosen customer a malicious server takes random or fixed decimal digit a, then random or fixed decimal digits b, c, d, e, f from 0,1,2,3,4,5 and submits a forged OTP "abcdef". From the description of the verification protocol the OTP is accepted if "bcdef" is identical to $E_K(A)$. The latter may be considered as a random 5-digit combination, where the digits 0, 1, 2, 3, 4, 5 appear independently each with probability 8^{-1}, and the digits 6, 7, 8, 9 appear with probability 16^{-1}, see Section III. So the attack success probability is 8^{-5}. That is much high than expected 10^{-6}.

B. One customer is targeted

The attack is based on a worst case assumption that static passwords are compromised and the malicious server is able to start a forgery after each correct authentication. To this end the malicious server should follow all communication between the customer's computer and the verifier, the bank. That does not seem a problem if the computer got infected by a trojan or some other malicious software, [5]. The trojan may be able to signal out to the malicious serve each time the customer got authenticated by the verifier. After that the malicious server may start a forgery by submitting a number of forged OTPs.

We assume that the customer needs 10 authentications per month on the average, for instance, to start a session and pay 9

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

bills. That makes 120 correct authentications per year on the average.

Also let r denote the number of allowed incorrect attempts to get authenticated after a correct one. In most banks r=3, which means that after 3 incorrect attempts, the account got blocked. A malicious server may submit r forged OTPs after each correct authentication. Therefore, the number of possible forged attempts is 120r per year. For the above basic attack the probability of at least one success per year is $p=1-(1-8^{-5})^{120r}$. For instance, p=0.0109 for r=3.

C. *Many customers are targeted*

If the attack works against one customer, that should work against many customers under the same conditions. Assume each of N customers needs 120 authentications per year on the average. Then the average number of successful forged authentications, that is the number of compromised customer accounts, is Np. For $N=10^4$ (a medium-size bank) and r=3, the number of compromised accounts is more than 100 per year.

D. *Analysis of the VASCO feedback*

After a first version of this report [6] was put on the Cryptology e-Print archive web-site, the author got a feedback from VASCO. They say that according to unpublished regulations an acceptable level of success probability of one forgery is 1/3000 in Norway. That is higher than one forgery success probability 8-5 argued in this report. However, if one uses the success probability from the regulations instead, a much higher success probability p=1-(1-1/3000)120r of at least one forgery per year per customer is achieved. That is 0.113 for r=3. Therefore, under this condition, for N=104 customers the number of compromised accounts is more than 103 per year on the average. We strongly believe that for better protecting customer's accounts the regulations are to be updated by reducing the acceptable level of the forgery success probability.

VI. CONCLUSION

In this paper we report a number of weaknesses found in the security token DIGIPASS GO3 which generates one-time 6-digit combinations for authentication in internet banking besides other applications. We were able to reconstruct its internal algorithm in some details and verification protocol. Also a basic attack against authentication based on this token is here described and its success probability is computed. An attack against many customer accounts is presented. We show that more than 1% of customer accounts per year may be compromised. A prerequisite for the attack is that an adversary is able to submit a number of forged OTPs after each correct authentication by the customer. To this end the adversary is to follow the communication between the customer's computer and the verifier (the bank).

Malicious software is widely employed by various adversaries. That provides them with a possibility to observe communication between customers and their banks if customer's computers are not properly protected. It looks there are no universal protection against numerous trojans, viruses and worms, and new kinds of them constantly appear. In particular, the adversaries may be now able to steal static passwords. Also by using malicious software diverse type automated forgery attacks may be mounted. One such attack is here described. In these circumstances dynamic one-time passwords seems the only protection against digital crime and mass surveillance. They are to be stronger nowadays.

REFERENCES

[1] DIGIPASS GO3-Ultra-portable, strong Authentication for highest convenience and user acceptability. URL: www.vasco.com/images/DIGIPASS-GO3-DS201007-v1_tcm42-47200.pdf Cited: September 24, 2017.

[2] D. M'Raihi, M. Bellare,F. Hoornaert,D. Naccache, and O. Ranen, "HOTP: An HMAC-Based One-Time Password Algorithm". URL: www.ietf.org/rfc/rfc4226.txt Cited: September 24, 2017.

[3] D. M'Raihi, S. Machani, M. Pei, J. Rydell, "TOTP: Time-Based One-Time Password Algorithm". URL: www.ietf.org/rfc/rfc6238.txt Cited: September 24, 2017.

[4] M. Adham, A. Azodi, Y. Desmedt and I. Karaolis, "How to Attack Two-Factor Authentication Internet Banking". in FC 2013, LNCS 7859, Berlin: Springer, 2013, pp. 322-328.

[5] Security 1:1 – Part 2 – Trojans and other security threat. URL: www.symantec.com/connect/articles/security-11-part-2-trojans-and-other-threats Cited: September 24, 2017.

[6] I. Semaev, "Experimental Study of DIGIPASS GO3 and the Security of Authentication", Cryptology ePrint Archive, 2015/609.

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731