Information-Theoretically Secure Hybrid Authentication in the Key Distribution Problem

Valeri Korjik
CINVESTAV-Guadalajara
E-mail: vkorjik@gdl.cinvestav.mx

Maxim Bakin
St.Petersburg University Of Telecommunications
E-mail: maxusb@hotmail.com

Abstract. We consider a cryptographic scenario when legal users share no secret key initially but their goal is to generate a shared *information - theoretically* secure key in the presence of an active eavesdropper. Some center broadcasts random string over noisy channels to legal users and this string can be received by an eavesdropper too. The main assumption is that legal channels are *superior* to illegal one. We focus our attention on the problem of authentication because an eavesdropper is active. Unfortunately, the use of the code authentication [1, 2] occurs ineffective because it requires very long authenticators that results in a very low key rate. We propose the *hybrid authentication* (*HA*) that combines both the code authentication and the authentication based on *Almost Strong Universal*$_2$ (*ASU*$_2$) class of hash functions [3]. The main contribution in the running paper is the theorem that allows to estimate the probability of incorrect authentication based on *ASU*$_2$ hashing and the use of partially secret key obtained after *privacy amplification* procedure. An example is presented to confirm that *HA* results in larger key rate than "pure" code authentication.

1. Introduction

We consider a particular case of broadcasting by friendly party some random chosen binary string X_0 of length k_0 to legal users Alice and Bob over binary symmetric channel (*BSC*) with bit error probabilities ε_A and ε_B, respectively, whereas an eavesdropper (Eve) receives the same string over *BSC* with bit error probability ε_E. We assume that $\varepsilon_A < \varepsilon_E$ and $\varepsilon_B < \varepsilon_E$, because it has been proved in [1] that key sharing problem is solvable if and only if these conditions are true.

To share information-theoretically secure key in a cryptographic scenario presented above it is necessary to *reconcile* the strings received by legal users and to apply privacy amplification procedure. The first procedure is necessary to correct errors in the strings X and Y that are noisy versions received by Alice and Bob, respectively, after broadcasting the string X_0. It can be done by sending by Alice a check string to Bob (or vice versa – sending by Bob a check string to Alice). The second procedure consists in a *hashing* by legal parties their reconciled strings into shorter key string. It is necessary in order to bound the information that Eve can gain about the final key. But before doing it one of the legal parties has to pick up randomly a hash function from *Universal*$_2$ class of hash functions and then send this function to another party. It is easy to prove that the probability to reconcile X and Y approaches 1 as $k_0 \to \infty$ if the number of check symbols r_0 satisfies the condition

$$r_0 \sim k_0\, h(\varepsilon_{AB}) \qquad (1)$$

where $h(..)$ is the entropy function and
$\varepsilon_{AB} = \varepsilon_A + \varepsilon_B\,(1 - 2\varepsilon_A)$.

The average amount of Shannon information I_0 about the final key leaking to an eavesdropper if she knows hash function, check string and eavesdropper's string Z, can be upper bounded as follows [4, 5]:

$$I_0 \le 2^{-(k_0 - l_0 - r_0 - t_0)} / \ln 2 \qquad (2)$$

where l_0 is the length of final key and t_0 is the Renyi information about X obtained be Eve from her string Z. In the case of broadcasting over *BSC*-s, the Renyi information can be found as follows

$$t_0 = k_0\left(1 + \log_2\left(\varepsilon_{AE}^2 + (1 - \varepsilon_{AE})^2\right)\right) \qquad (3)$$

where $\varepsilon_{AE} = \varepsilon_E + \varepsilon_A\,(1 - 2\varepsilon_E)$

Since an eavesdropper is active she can change both the check strings and the hash functions (also presented as some strings) transmitted by Alice. It enables Eve to share false key with Bob. To prevent such Eve's attack Alice should authenticate these messages containing roughly $k_0(1 + h(\varepsilon_{AB}))$ bits. Due to lack of secret keys possession by legal parties, the use of the parts $\tilde{X}$ ($\tilde{Y}$) of initial strings X (Y) is necessary to provide this authentication. The method of *code authentication* based on smaller difference between the strings $\tilde{X}$ and $\tilde{Y}$ rather than $\tilde{X}$ and $\tilde{Z}$ can be used to solve this problem. Then the authenticator δ taken from some bit positions of $\tilde{X}$ is appended to message M in line by a certain rule to each message M. This rule can be considered as some binary block code of length $\tilde{k}_0$ consisting of 2^s code words for each of 2^s possible messages, where $\tilde{k}_0$ is the length of $\tilde{X}$ ($\tilde{Y}$) and s is the length of message being authenticated.

The simplest way to design such authentication code [1] is to take some binary linear ($\tilde{k}_0$, s) – code with maximum possible Hamming distance d and replace every bit in its code words by pair of bits following the rule: 0 by 01 and 1 by 10. Then the following relations for the probability P_{Re} (to be rejection of the original message by Bob when an intruder has not intervened into transmission at all) and P_{Ch} (to be the acceptance of the false message by Bob if it has been changed by an intruder) are true, respectively [2]:

$$P_{Re} = \sum_{i=\tilde{l}+1}^{\tilde{k}_0} \binom{\tilde{k}_0}{i} \varepsilon_{AB}^{i} \left(1-\varepsilon_{AB}\right)^{\tilde{k}_0-i} \tag{4}$$

$$P_{Ch} = \sum_{i=0}^{\tilde{l}} \binom{d}{i} \varepsilon_{BE}^{i} \left(1-\varepsilon_{BE}\right)^{d-i} \sum_{j=0}^{\tilde{l}-i} \binom{\tilde{k}_0-d}{j} \varepsilon_{AB} \left(1-\varepsilon_{AB}\right)^{\tilde{k}_0-d-j} \tag{5}$$

where $\varepsilon_{BE} = \varepsilon_B + \varepsilon_E (1 - 2\varepsilon_B)$, $\tilde{l}$ is some fixed threshold.

Example 1. Consider binary BCH code (1023, 208, 231). Let us select $\varepsilon_{AB} = 0.0177$ and $\varepsilon_{BE} = 0.2$, then using (4) - (5) we get $P_{Re} \approx 1.1*10^{-4}$, $P_{Ch} \approx 1.0*10^{-4}$ for $\tilde{l}$ =35.

It is wise to perform the optimixation procedure. Given the parameters ε_A, ε_B, ε_E, $\tilde{k}_0$, P_{Re} and P_{Ch}, minimize the length of the authenticator over all ($\tilde{k}_0$, s, d) codes.

2. Hybrid Authentication

Unfortunately, the use of even the best authentication codes results in very long authenticators and requires huge consumption of the X(Y) string material [1, 2]. More effective *hybrid authentication* can be described as a part of the following *key sharing* algorithm secure against active eavesdropper.

1. Alice and Bob divide the received strings X(Y) of length k_0 into three parts X_1(Y_1), X_2(Y_2) and X_3(Y_3), which have the lengths k_1, k_2, k_3, respectively.
2. Alice forms the check string c_2 of length r_2 for her substring X_2 using the check matrix of some binary linear (n_2, k_2) – code V, where $n_2 = k_2 + r_2$. The initially chosen code V should have a constructive error correcting algorithm that is capable to reconcile the strings X_2 and Y_2 with high probability.
3. Alice generates a truly random hash function h_2 taken from Universal$_2$ class for privacy amplification of X_2. Then the length of binary string to represent h_2 has to be k_2.
4. Alice sends to Bob over a public noiseless channel both c_2 and h_2 using the authentication code considered in the Introduction, based on her substring X_3.
5. Bob verifies the authenticity of the received strings c_2 and h_2 using authentication code and his substring Y_3, as it was described in the Introduction.
6. Bob corrects error on Y_2 using the received check string c_2. (We believe that after the completion of error correcting procedure Bob obtains the string $\tilde{Y}_2$ that coincides with X_2 with high probability.)
7. Both Alice and Bob hash their strings X_2 and $\tilde{Y}_2$ respectively (which should be the same with high probability) into shorter string S_0 having such a length l that the information of S_0 leaking to Eve is negligible (see (2), where k_0, r_0 and l_0 should be changed to k_2, r_2 and l, respectively).
8. Alice forms the check string c_1 of length r_1 to her substring X_1 using the check matrix of some binary linear (n_1, k_1) – code $\tilde{V}$, where $n_1 = k_1 + r_1$. The initially chosen code $\tilde{V}$ should have a constructive error correcting algorithm that is capable to reconcile the strings X_1 and Y_1 after the completion of error correcting procedure with high probability.
9. Alice generates a truly random hash function h_1 taken from Universal$_2$ class for privacy amplification of X_1. Then the length of binary string to represent h_1 has to be k_1.
10. Alice sends to Bob over a public noiseless channel both c_1 and h_1 using the authentication based

on keyed hashing in the class ε - ASU_2 hash functions with the key S_0 obtained in the step 7.

11. Bob verifies the authenticity of the received strings c_1 and h_1 using ε - ASU_2 based algorithm and the key S_0 obtained in the step 7. (A description of the class ε - ASU_2 functions will be given below).

12. Bob corrects errors on Y_1 using the received check string c_1. (We can believe that after the completion of error correcting procedure Bob obtains the string $\tilde{Y}_1$ that coincides with X_1 with high probability).

13. Both Alice and Bob hash their strings X_1 and $\tilde{Y}_1$, respectively (which should be the same with high probability after error correction) into shorter string K having such a length l_0 that the information of K leaking to Eve is negligible (see (6), where k_0 and r_0 should be changed to k_1 and r_1, respectively and t_0 is the Renyi information about X_1 obtained by Eve from her string Z.

The final key K produced by both of the legal parties can be used as a cryptographic secret key either to encrypt long messages in computational secure cryptosystems or to encrypt messages of the same length in ideal time – pad cryptosystems.

The "trick" of hybrid authentication is that authentication procedure based on ε - ASU_2 class can be used with short enough keys. Hence large string material consumption which is necessary for code authentication using the strings $X_2(Y_2)$ does not practically affect on the key rate because it is commonly to take $k_1 >> k_2$. But the "bottleneck" of hybrid authentication is keyed hashing in the class of ε - ASU_2 with partially secret key S_0 obtained after privacy amplification of the substrings $X_2(Y_2)$. Unconditionally – secure authentication with a partially secret key and the use of *Strong Universal*$_2$ hashing has been considered in [6]. But we need an extension of that theory to the ε - ASU_2 class of hash functions.

Let us remember the authentication technique based on keyed hash functions. Consider a class F of hash functions $\{0, 1\}^a \to \{0, 1\}^b$, $a > b$, each of them can be compared with the key, chosen at random. Then we can take the message (the bit string of the length a) as an argument for a hash function and the output bit string of the length b – as an authenticator that can be appended to the message string. Assuming that a verifier knows the authentication key he (or she) can calculate the hash function of the received message and compare the result with the received authenticator. In the case of their coincidence the message is accepted as a genuine one and rejected otherwise. It is very attractive to select as a class F of hash functions the so called ε - *Almost Strongly Universal*$_2$ (ε - ASU_2 for short) class because it allows to save the length of keys [3].

Denote by $A = \{0, 1\}^a$ and $B = \{0, 1\}^b$ the sets of messages and authenticators respectively. A class F of ε - ASU_2 hash functions satisfies the following two conditions [3]:

$$\# f = \frac{|F|}{|B|}$$

if $f \in F$, $f(\alpha) = \beta$, for every $\alpha \in A$, $\beta \in B$, where $|X|$ denotes the cardinality of the set X.

$$\# f \leq \frac{\varepsilon \ |F|}{|B|}$$

if $f \in F$, $f(\alpha_1) = \beta_1$, $f(\alpha_2) = \beta_2$ for every distinct $\alpha_1, \alpha_2 \in A$ and for every $\beta_1, \beta_2 \in B$. An $(1/|B|)$ - ASU_2 class is called *Strongly Unversal*$_2$ (or SU_2).

We suggest to use the authentication scheme based on ε - ASU_2 hashing in the steps 10-11 of key sharing algorithm. Then the key to hashing procedure occurs only partially secure because it is obtained in the step 7 after privacy amplification of the strings $X_2(Y_2)$ and hence Eve has some information (very small as a rule) about this key.

The following theorem is the main contribution of the running paper:

Theorem. *The expected conditional probability P_f of false authentication based on ε - ASU_2 keyed authentication and partially secret key S_0, averaged over all randomly chosen by Alice hash functions to provide the key S_0 is upper bounded as follows*

$$P_f \leq 2^{-\left(\frac{\tilde{b} - l(1-t)}{2} - 1\right)} \qquad (6)$$

where $\tilde{b} = - \log_2 \varepsilon$, l is the length of the key S_0, $t = - \frac{1}{l} \log_2 P_C$, P_C is the minimal possible value satisfying the inequality

$$l + (1 - P_C)\log_2 \frac{1 - P_C}{2^l - 1} + P_C \log P_C \leq 2^{-(k_2 - r_2 - l - t_2)} \quad (7)$$

$$t_2 = k_2\left(1 + \log_2\left(\varepsilon_{AE}^2 + (1 - \varepsilon_{AE})^2\right)\right)$$

(The proof of this theorem is given in [7]).

Using this theorem we can maximize the key rate given parameters ε_A, ε_B, ε_E.

Example 2. Let us take $\varepsilon_{AB} = 0.01$, $\varepsilon_{AE} = \varepsilon_{BE} = 0.2$, P_R (the probability that the amount of Shannon information I_0 about the final key K leaking to an eavesdropper can be exceeded) $\le 10^{-4}$, P_{CS} (the probability of correct key sharing between legal parties when an eavesdropper did not intervened at all) ≥ 0.99, $I_0 \le 10^{-10}$ bit, l_0 (the final key length) = 2048.

Using the formulas (2) and (3) we get the following parameters for the first substring $X_1(Y_1)$: $k_1 = 5672$, r_1 (the length of check string to X_1) = 1071. Next we use a class of ε - ASU_2 functions ($A \rightarrow B$) [3] with parameters: $\varepsilon = (i + 1) / q$, $|A| = q^{2^i}$, $|B| = q$, where i, q are arbitrary integers. If we take $i = 7$, $q = 2^{23}$, this method allows to authenticate strings of length 6784 that is enough to authenticate hash function of the length 5672 and 1071 check bits. The key length should be then 477. To produce the key S_0 of the length 477 we can select $k_2 = 2686$ and $r_2 = 563$. Code authentication can be done by the use of binary linear code (4832, 3246) with minimal code distance 292. Eventually, it results in the full length of the string $X(Y)$ as $k_1 + k_2 + k_3 = 18034$ bits to produce information – theoretically secure and reliable final key of the length 2048. It can be easy shown that the use of "pure" code authentication for the same length of final key requires the length of *X(Y)* equals to 23406 bits. This moderate difference in efficiency between HA and code authentication can be increased for larger key lengths l_0.

3.Conclusion

The main goal of this paper was to elaborate hybrid authentication. But it cannot be considered isolated from a key sharing because eventually we want to provide the maximal possible key rate and nothing else.

Our contribution to this theme is to extend the bound proved before for SU_2 – class hashing for the probability of undetected modification of a message to the case of ε - ASU_2 hashing. We represent also an algorithm to select the parameters of hybrid authentication (and as a matter of fact, the parameters of key sharing protocol based on noisy channels). We are going to consider an example of such optimization in the future and we hope that hybrid authentication provides an advantage in comparison with pure code authentication. The difficulty consists in finding effective classes of ε - ASU_2 hashing. It seems likely that interactive authentication codes [8] can be adopted specially for this purpose as well. It is not inconceivable that hybrid authentication becomes very effective for large enough key length (beginning from tens of hundred thousand bits). Then they can be changed to such bounds which are especially effective for large block lengths. Anyway, we hope that this paper is some contribution to move perfect secret key agreement based on noisy channels closer to being practical.

References.

1. U.Maurer, "Information – theoretically secure secret – key agreement by NOT authenticated public discussion", *Proc. of Eurocrypt'97*.
2. V.Korjik, M.Bakin, "Information – theoretically secure keyless authenticaton", *Proc. 2000 Int. Symp. on IT*, p. 20.
3. D.R.Stinson, "Universal hashing and authentication codes", *Advances in Cryptology, Proc. of Crypto'91, Lecture Notes in Computer Science*, vol. 576, 1992, pp. 74 – 85.
4. C.Bennet, G.Brassard, U.Maurer, "Generalized Privacy Amplification", *IEEE Trans. on IT*, vol. 41, no 6, 1995, pp. 1915 – 1923.
5. V.Korjik, G.Morales – Luna, V.Balakirsky, "Enhanced Privacy Amplification Theorem for Noisy Main Channel", (*Submitted for Crypto'2001*).
6. U.Maurer, S.Wolf, "Privacy amplification secure against active adversaries", *Proc. Crypto'97*.
7. V.Korjk, M.Bakin, "Hybrid Authentication Based on Noisy Channels" (*Submitted for the "Information Security" Journal*)
8. P.Gemmel, M.Naor, "Codes for interactive authentication", *Advances in Cryptology, Proc. of Crypto'93, Lecture Notes in Computer Science*, vol. 773, 1993, p. 355-367.