18

ZERO KNOWLEDGE BROADCASTING IDENTIFICATION SCHEME

Magdi S. El-Soudani, Heba S. El-Refaey, and Hebat-Allah M. Mourad,
Electronics and Communications Dept., Faculty of Engineering, Cairo University

Abstract: Zero knowledge proofs form an important category in the public key identification protocols, they are depending on number theory. In 1989, Stern announced his protocol which is based on syndrome-decoding problem, he also studied the attacks against this type of problems. In this paper, we propose a broadcasting variant based on the Stern's Identification scheme. Broadcasting is applied when there are one prover and many verifiers. In the proposed broadcasting scheme, the prover is communicating with verifiers through a broadcasting channel so he is running the identification session once, which minimizes the time and the communication complexity. We have developed Stern basic scheme to be adequate for broadcasting applications, but the underlying hard problem that the security of Stern identification scheme depends on, is used as it is.

Keywords: Zero-knowledge, syndrome decoding

1. INTRODUCTION

Zero Knowledge proofs are the proofs used in convincing someone that you have successfully solved a problem without conveying any knowledge whatsoever of what the solution is [1]. This type of proofs can be used in identification as a public key cryptosystem. Identification is a process through which one ascertains the identity of another person or entity.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35586-3_46

M. A. Ghonaimy et al. (eds.), *Security in the Information Society*

Entities on a network may also identify one another using Zero Knowledge proofs. Nearly all the public key cryptosystems are based on hard problems from the number theory except few cases like the one introduced by McEliece based on error correcting codes, which started a new direction in cryptography [2]. There are problems from the coding theory that proved to be suitable for cryptography such as the syndrome-decoding problem and the problem of finding a codeword with a specific weight [3,4]. Using these problems reduces the computing load since the problems from coding theory depend on simple operation compared to those from number theory. On the other hand, zero knowledge proofs used for public key identification protocols were dependent on number theory till 1989 when Stern announced his protocol which is based on syndrome-decoding problem [5]. In the case of identification application with one prover and many verifiers the broadcasting is a good solution as mentioned in [6]. In this work we propose a broadcasting variant of Stern basic scheme. In such a case, the prover broadcasts his messages through the network, so all the verifiers can check the prover's identity at the same time. This type of proofs can be used in any application with group of users dealing with one entity and need to be sure of its identity. In the next section, we explain the basic information that will be needed through the rest of the paper with a brief review of the zero knowledge identification scheme introduced by Stern. In section three, we give the broadcasting variant of Stern scheme. We discuss the complexity of the proposed scheme and its aspects of security in section four. We conclude our paper with comments on the proposed scheme.

2. STERN IDENTIFICATION PROTOCOL

Stern Zero Knowledge identification scheme is based on the hardness of a problem from coding theory which is called the syndrome decoding (SD) [7]. Using this technique in addition to the use of one-way hash functions- to ensure security- makes Stern's scheme practical. This is because it only uses the logical operations induced by coding and the efficient operations involved in hashing. In this scheme, the identification of users in a system is run by a central trusted authority. The scheme is an interactive. In an interactive proof system, there are two participants Prover **P,** and Verifier **V.** **P** knows some facts, and he wishes to prove to **V** that he does. Each of them will perform private computations and has private random number generator. They communicate to each other through a communication channel.

The scheme uses a fixed (m x n) matrix **H,** over GF(2), which is common to all users and is originally built randomly by the authority. **H** is used as parity check matrix; it provides a linear binary code with a good correcting power. The scheme consists of two stages, the first stage is the key generation, and the second one is the interactive protocol steps. Upon registration, each user **U** receives a secret key $\mathbf{S_U}$, chosen at random by the

authority among all n-bit words with a prescribed number p of 1's. This prescribed number p is also part of the system. The public identification of the user is computed as: $\mathbf{i_U} = \mathbf{H} \cdot \mathbf{S_U}$. This public identification is the syndrome for the vector $\mathbf{S_U}$. The vector $\mathbf{i_U}$, or its value is made available in some form of directory or certified by means of a digital signature of the authority. Using this public identification, the user proves that he is registered either through the directory or by means of the certificate, so he can execute the session. After the user (prover) proves that he is a registered user using his public key, he needs to go through an interactive session (which forms one round). The interactive proof, which is a challenge-and-response protocol, consists of a specified number of rounds. The prover picks a random n-bit word **y** together with a random permutation σ of the integers {1...n} and sends commitments C_1, C_2, and C_3 to the verifier. A commitment is an electronic way to temporarily hide a sequence of bits that cannot be changed. These commitments are calculated using the following relations [8]:

$$\mathbf{C_1} = < \sigma \,||\, \mathbf{H(y)} > \qquad \text{(1-a)}$$

$$\mathbf{C_2} = < \mathbf{y} \bullet \sigma > \qquad \text{(1-b)}$$

$$\mathbf{C_3} = <(\mathbf{y} \oplus \mathbf{S_U}) \bullet \sigma > \qquad \text{(1-c)}$$

where permutation σ is being considered in this setting as a vector of bits, which encodes it; and $(\mathbf{y} \bullet \sigma)$ refers to the image of **y** under permutation σ. The permutation operation means changing the bits order of the vector according to a function that depends on σ without changing the vector's Hamming weight. The commitment C_1 is a concatenation, referred to by $||$, of two vectors. The verifier sends a random element "b" of {0, 1, and 2}. Upon the value of "b", the prover sends the vectors that will help the verifier to do the necessary check. For example, If "b" is equal to 1, the prover returns $(\mathbf{y} \oplus \mathbf{S_U})$ and σ, and the verifier checks that commitments C_1 and C_3 were correct. If the result of checking the commitments leads to accepting the prover, the verifier continues for next round, otherwise he sends a "reject" answer to the prover and ends the session. This round could be repeated r times. In order to counterfeit a given identity without knowing the secret key, Stern proposed various strategies that could be used to cheat the basic scheme. The probability of success of any cheating strategy is bounded by $(2/3)^r$, where r is the number of rounds [8].

3. PROPOSED BROADCASTING SCHEME

This variant is similar to Stern basic scheme but instead of receiving the challenge from one verifier, the prover receive it from multi verifiers and uses it as an input for a pre-specified function (agreed upon in advance). This function outputs one challenge that used in same way as in the basic

scheme. In this scheme, the verifiers are allowed to communicate freely during the rounds, but we assumes that if any of the verifiers discovered any cheating from the prover by altering his challenge, he must stop his participation in this session. Then at the end of the session each verifier will send his signed decision, and the prover will broadcast all these decisions to prove that he is accepted from all the verifiers present at the start of the session.

3.1 Model Assumptions and Preparation Stage

To run the protocol we need the keys to be ready. The key generation stage is as described in the previous section for the basic scheme. We have the same construction for the secret key which is a randomly selected n-bits vector S_U with a prescribed number of ones p, and the public key which consists of the parity check matrix of the used code H (m x n) and the vector i_U which is the syndrome for the secret key, $i_U = H.S_U$.

Furthermore, we made the following assumptions.

(1)We have one prover P and many verifiers Vi, I=1,2,..v.
(2)P broadcasts to all the verifiers.
(3)Each V_i has a one-directional communication link to P.
(4)The verifiers are honest, independent and no interaction exist between them.
(5)All the protocol parties are members in a digital signature scheme controlled by the Trusted authority. Each one can sign a message and all the others (the prover and the other verifiers) can check the signature.
(6)The number of rounds for running the protocol is predetermined before starting the session. The secret key is calculated in the same way as the basic scheme.

3.2 Protocol Steps

As before, the identification schemes consists of two stages: key generation and interactive protocol steps. In the first stage, the prover sends his public key to the trusted authority to get its session secret key. Each verifier has to send a signed request to join the session, then P has to broadcast the list of the session's verifiers. as shown in Figure 1. In the second stage, the following protocol steps are executed. This stage is repeated for each round.

(1) P picks randomly y and then forms the commitments C_1, C_2, C_3 and C as follows:

$$\mathbf{C1} = \langle \sigma \,||\, \mathbf{H(y)} \rangle \quad \text{(2-a)}$$

$$\mathbf{C2} = \langle \mathbf{y} \bullet \sigma \rangle \quad \text{(2-b)}$$

$$\mathbf{C3} = \langle (\mathbf{y} \oplus \mathbf{SU}) \bullet \sigma \rangle \quad \text{(2-c)}$$

$$\mathbf{C} = \langle \mathbf{C1} \,||\, \mathbf{C2} \,||\, \mathbf{C3} \rangle \quad \text{(2-d)}$$

The prover P forms the image < u > of u using public key cryptographic Hash function.

(2) Each verifier V_i sends his challenge "b_i" , $b_i \in (0, 1, 2)$, to P.

(3) P receives challenges from all V_i's, and calculates the challenge to be used from the received challenges according to the formula:

$$b = \bigoplus_{i=0}^{v} b_i$$

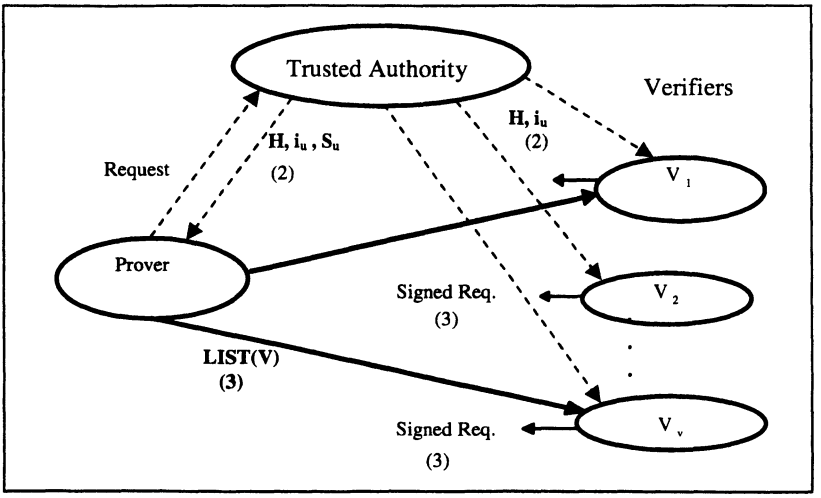

Figure 1. Key generation and preparation stage.

(4) P forms the challenge array as a list of all the received challenges.

$$B = \|_{i=0}^{v} b_i$$

(5) According to b, and similar to the basic scheme, the prover broadcasts the appropriate response:

If b equals 0, the response will be y, σ, C_3,and B.

If b equals 1, the response will be $y \oplus S_U$, σ, C_2, and B.

If b equals 2, the response will be $y \bullet \sigma$, $S_U \bullet \sigma$, C_1, and B.

Each verifier rechecks the B array against b_i, then regenerates the challenge and verifies the commitment C using the response:

If b equals 0, then forms C_1, and C_2.

If b equals 1, then forms C_1, and C_3.
If b equals 2, then forms C_2, and C_3. In addition the verifier checks that the weight of S_U, σ equals the prescribed value of p.

(6) If the verification gives "accept" as a result, the verifier continues to the next round. Otherwise, he leaves the session.

(7) At the last round each verifier forms a signed decision message, and sends it to the prover, which broadcasts these signed messages successively, to prove his identity.

The protocol steps are shown in Figure 2.

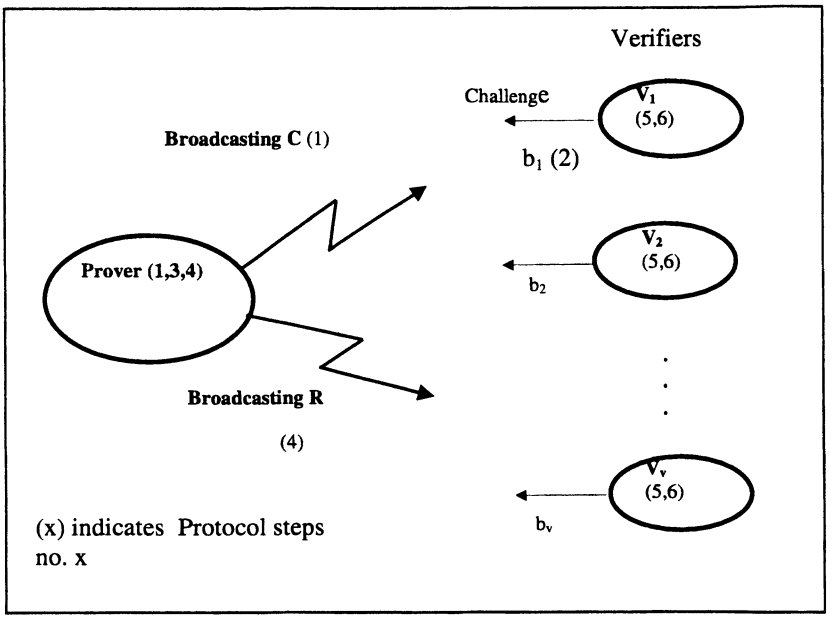

Figure 2 Steps of proposed broadcasting scheme

4. COMPLEXITY AND SECURITY OF PROPOSED SCHEME

4.1 Communication Complexity

The overall number of bits is a measure of the communication complexity (CC). This is given by the following equation.

$$CC = v * LDS_{ID} + r * LC + v * LDS_F \qquad (3)$$

where v is the number of verifiers, LC is the number of bits for one round; LDS_{ID} is the length of the signed ID's in the preparation stage and LDS_F is the length of the final decision message. For each one-way channel, the required number of bits is:

$$N_b = LDS_{ID} + r * (L_b)$$

where L_b is the length of the challenge messages. Selecting appropriate digital signature scheme will control the communication complexity of the protocol. For the basic protocol the communication complexity is (v * LC * r) assuming that verification is done for each verifier separately. Stern proposed various strategies that could be used to cheat the basic scheme and showed that the probability of success of any cheating strategy is bounded above by $(2/3)^r$, where r is the number of rounds [8]. Therefor, the security level can be given by:

$SL = (2/3)^r$, or $$r = \frac{\log SL}{\log 2/3} = -5.679 \log SL$$

For example, in order to achieve a level of security of 10^{-6} r must be 35. We can rewrite equation (3) as follows:

$$CC = v * LDS_{ID} + (-5.679 \log SL) * LC + v * LDS_F$$

It is clear that the weight produced by the digitally signed messages is independent on the number of rounds. It depends only on the number of verifiers in the session.

Figure 3 shows the communication complexity of the proposed broadcasting variant compared with the basic scheme. We assume permutation of 120 bit long, hashing value of 128 bits, code length of 512 bits, and DSA is used as a digital signature scheme. Thus for a length of the digital signature 320, assuming r = 35, and number of verifiers = 10, the communication complexity is calculated for the broadcasting channel. We assume that the basic scheme is run for each user separately. It is clear that the broadcasting variant has lower communication complexity.

4.2 Computing Load

The computing load for the scheme is the time needed for the prover and for the verifier to calculate the required values for the protocol. The Computing Load for the prover (CLP) and verifier (CLV) can be found using the following relations:

$$CLP = r * \{CL(H(y)) + 4* CL(Hash) + 3* CL(Permutation) + (3 + v) * CL(Concatenation) + (1 + v)CL(XOR)\} + v * CL(DS). \quad (4)$$

$$CLV = r * \{CL(H(y)) + 3* CL(Hash) + CL(Permutation) + 3 * CL(Concatenation) + (1 + v) CL(XOR)\} + v * CL(DS). \quad (5)$$

where CL(x) is the computing load for performing process (x).

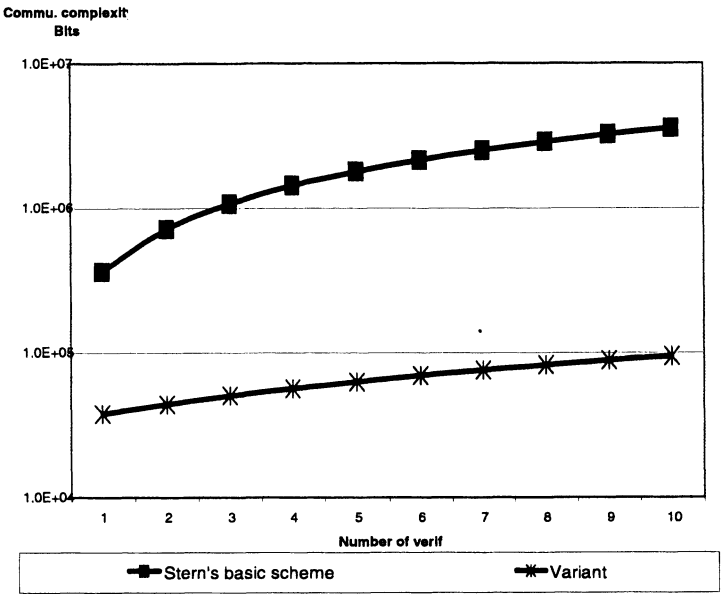

Figure 3 Relation between communication complexity and number of verifiers for basic scheme and proposed scheme (variant).

The major computing load is to find H (y). This is because the protocol is used for large codes, otherwise the hashing computing load will be comparable to H (y) computing load, so this variant has the same computing load as the basic scheme in addition to the load due to the digital signature.

4.3 Security of the Scheme

The security of any scheme depends on three parameters the completeness, the soundness, and the zero knowledge, therefore, we discuss these parameters in our case.

4.3.1 Completeness:

As defined in [9], the honest prover will answer correctly -with overwhelming probability- the verifier's challenges. Similar to the basic scheme we have input I = ($\mathbf{H}$,$\mathbf{i_u}$) and the polynomial time predicate consists of the two relations $\mathbf{H}(\mathbf{S_U}) = \mathbf{i_u}$; and $w_H(\mathbf{S_U}) = p$. ACC(P, V, I) is the case that the verifiers will accept the honest prover. In this variant we have Pr {ACC (P, V, I)} = 1 always [8].

4.3.2 Soundness:

Similarly, soundness is the probability that a dishonest prover answers correctly the verifiers' questions is negligible. In our scheme Pr {ACC (P', V, I)} = $(2/3)^r$. So the soundness in this case will depend on the number of rounds r. To make the probability of success of the dishonest prover negligible we need large r, i. e. for order 10^{-6} we need r = 35. Stern used another definition from [9] to prove the soundness of the basic scheme.

4.3.3 Zero Knowledge:

The definition of Zero Knowledge in [1] states that an interactive proof system of knowledge is Zero Knowledge if for all polynomial time machine $\mathbf{V'}$ its view of communication ($\mathbf{P}$, $\mathbf{V'}$) can be recreated with indistinguishable probability distribution under no additional assumption by a polynomial time probabilistic machine (simulator) S. The multiple verifiers can be simulated by a separate machine for each verifier or by one machine running multiple clones of the verifier program. Since the commitments are built from a related data, it is difficult to claim that they will be indistinguishable from those created by the simulator. To solve this problem Stern offered to use the random hashing technique [8], which involves changing <x> into $< \rho \,||\, \rho \oplus x >$ where ρ is a randomly chosen string with the same length as x. To apply this to our case we need to prove that the commitment C follows the same distribution when they come from a legitimate user P and when it is produced by any of the three cheating strategies given in [8]. For the first strategy $\mathbf{S_U}$ is replaced by some arbitrary vector t of the same weight p; and $\mathbf{C}$ will be in the form:

$$\mathbf{C} = < \rho \,||\, \rho \oplus \{<\rho_1 \,||\, \rho_1 \oplus (\sigma \,||\, \mathbf{H(y)})> \,||\, <\rho_2 \,||\, \rho_2 \oplus (\mathbf{y} \cdot \sigma) > \,||\, <\rho_3 \,||\, \rho_3 \oplus ((\mathbf{y} \oplus \mathbf{t}) \cdot \sigma)>\}> \qquad (6)$$

So we have

$$(\rho, \rho_1, \rho_2, \rho_3, \sigma, y) \rightarrow (\rho, \rho_1, \rho_2, \rho_3 \oplus \{(S_U \oplus t) \cdot \sigma\}, \sigma, y)$$

As a permutation of the underlying probability space, which transforms the commitment C into the corresponding commitment generated by the legitimate user. So both distributions are alike and the variant is Zero Knowledge.

5. COMMENTS

We proposed a broadcasting variant of Stern basic protocol. The proposed scheme is an interactive version that uses digital signature to ensure the final decision. This makes it secure and the communication complexity is much lower. We have developed Stern basic scheme, to be adequate for broadcasting applications, but the underlying hard problem that the security of Stern identification scheme depends on, is used as it is. So our variants' security still depends on the hardness of the syndrome-decoding problem. This variant can be used in case one user has to prove his identity to multi-verifiers; which can be used in point to multi-point (P-MP) broadcasting schemes. However, the basic scheme can be used to prove something to many people by running it *sequentially* i.e., by running the basic scheme with each verifier separately and after finishing it with each verifier the prover starts a new session with the next verifier till finishing with all the verifiers, but this solution could be very time consuming. The other alternative is to apply the basic scheme *in parallel*, by running the scheme with all the verifiers at the same time, but with different messages for each verifier. However, in both cases the number of bits communicated will be very large. In our study we mainly use linear codes since they are well-defined and easy to deal with. Stern basic scheme uses random codes since the use of random codes of large length is really a hard problem either in generating the code or in performing the necessary protocol steps. There is more work to do verify the properties, applicability and reliability of the proposed scheme in the real situations; so it is still an open point that can be covered in a future work.

REFERENCES

[1] Simmons G., " Contemporary Cryptography, The Science of Information Integrity", IEEE Press, 1991.

[2] McEliece R., " A Public Key Crypto-System Based on Algebraic Coding Theory", Jet Propulsion Lab. DSN Progress Report., 1978, pp. 114-116.

[3] Stern J., " A New Identification Scheme based on Syndrome Decoding", In Proc. Crypto '93, Lecture Notes in Computer Science, Berlin, Springer Verlag, 1993, Vol-773, pp.13-21.
[4] Chabaud F., " On the Security of Some Cryptosystems Based on Error Correcting Codes", Advances in Cryptology, Eurocrypt'94, Proc. Of Workshop on The Theory and Applications of Cryptographic Techniques, Perugia, Italy, May 1994, pp. 131-139.
[5] Stern J., " An Alternate to the Fiat-Shamir Protocol", In Proc. Eurocrypt 89, Lecture Notes in Computer Science, Berlin, Springer Verlag, Vol. 434, pp.106-113.
[6] Burmester M., Desmedt Y., "Broadcast Interactive Proofs", extended abstract, Eurocrypt 91, pp. 81-95.
[7] E.R. Berlkamp, R. J. MacEliece, and H. C. A. Van Tilborg, " On the Inherent Interactability of Certain Coding Problems", IEEE Trans. Information Theory, Vol- IT-24, 1987 pp. 384-386.
[8] J. Stern, " A New Paradigm for Public Key Identification", IEEE Trans. Information Theory, Vol-42, 1996, pp. 1757-1768.
[9] U. Feige, A. Fiat, and A. Shamir, " Zero Knowledge Proofs of Identity", in Proc. Of 19th ACM Symposium on Theory of Computing, 1987.