A Note on the Cramer-Damgård Identification Scheme

Yunlei Zhao[1], Shirley H.C. Cheung[2], Binyu Zang[1], and Bin Zhu[3]

[1] Software School, Fudan University, Shanghai 200433, P.R. China
{990314, byzang}@fudan.edu.cn
[2] Department of Computer Science, City University of Hong Kong, Hong Kong
hccheung@cs.cityu.edu.hk
[3] Microsoft Research Asia, Beijing, P.R. China
binzhu@microsoft.com

Abstract. In light of the recent work of Micali and Reyzin on showing the subtleties and complexities of the soundness notions of zero-knowledge (ZK) protocols when the verifier has his public-key, we re-investigate the Cramer-Damgård intended-verifier identification scheme and show two man-in-the-middle attacks in some reasonable settings: one simple replaying attack and one ingenious interleaving attack. Our attacks are independent of the underlying hardness assumptions assumed.

Keywords: Cryptography, identification scheme, Σ_{OR}, man-in-the-middle attacks.

1 Introduction

Identification protocol is one of the major cryptographic applications, especially in E-commerce over the Internet. Feige, Fiat and Shamir introduced a paradigm for identification (ID) schemes based on the notion of zero-knowledge (ZK) proof of knowledge [6, 5]. In essence, a prover identifies himself by convincing the verifier of knowing a given secret. Almost all subsequent ID schemes followed this paradigm. But, all previous Fiat-Shamir-like ZK-based ID schemes suffer from a weakness, as observed by Bengio et al [1]. Specifically, a malicious verifier may simply act as a moderator between the prover and yet another verifier, thus enabling the malicious verifier to pass as the prover. In [2] Cramer and Damgård presented a simple yet efficient ZK-based (specifically, Σ_{OR}-based) solution for preventing aforementioned man-in-the-middle attacks. Essentially, beyond the novel use of Σ_{OR} in the identification setting, in the Cramer-Damgård ID scheme not only the identification prover but also the identification verifier are required to have public-keys. In other words, the Cramer-Damgård scheme is an *intended-verifier* ID scheme. Though the intended-verifier property is necessary to prevent aforementioned man-in-the-middle attacks, it brings other security issues, as we shall observe in this paper, in light of the recent work of Micali and Reyzin [8] on showing the subtleties and complexities of the soundness notions of zero-knowledge (ZK) protocols when the verifier has his public-key.

X. Deng and Y. Ye (Eds.): WINE 2005, LNCS 3828, pp. 385–390, 2005.

2 Description of the Cramer-Damgård Intended-Verifier ID Scheme

In this section, we first present the basic tools used in the Cramer-Damgård ID scheme and then give the protocol description of the Cramer-Damgård ID scheme.

We assume the following form of *3-round protocol* is considered, which is known as Σ-protocols. Suppose P and V are probabilistic polynomial-time (PPT) machines, on common input x to P and V, and a w such that $(x, w) \in R$ is the only advantage of P over V that he knows w. The *conversation* of a 3-round protocol $\langle P, V\rangle$ is defined as a 3-tuple, say (a, e, z), where a is the first message sent from P to V, e is a random string sent from V to P, and z is replied by P to V. After this 3-round conversation, V would decide to accept or reject based on the conversation.

2.1 Σ-Protocol and Σ_{OR}-Protocol

Definition 1 (Σ-protocol). *A 3-round protocol $\langle P, V\rangle$ is said to be a Σ-protocol for a relation R if the following holds:*

- **Completeness.** *If prover P and verifier V follow the protocol, the verifier always accepts.*
- **Special soundness.** *From any common input x of length n and any pair of accepting conversations on input x, (a, e, z) and (a, e', z') where $e \neq e'$, one can efficiently compute w such that $(x, w) \in R$. Here a, e, z stand for the first, the second and the third message respectively, and e is assumed to be a string of length k (that is polynomially related to n) selected uniformly at random from $\{0, 1\}^k$.*
- **Perfect Special honest verifier zero-knowledge (SHVZK).** *There exists a probabilistic polynomial-time (PPT) simulator S, which on input x (where there exists a w such that $(x, w) \in R$) and a random challenge string $\hat{e}$, outputs an accepting conversation of the form $(\hat{a}, \hat{e}, \hat{z})$, with the same probability distribution as that of the real conversation (a, e, z) between the honest $P(w)$, V on input x.*

Σ-protocols have been proved to be a very powerful cryptographic tool and are widely used in numerous important cryptographic applications including digital signatures, efficient electronic payment systems, electronic voting systems, et al. We remark that a very large number of Σ-protocols have been developed in the literature, mainly in the field of applied cryptography and in industry. Below, we give Σ-protocol examples for DLP and RSA.

Σ-Protocol for DLP [9]. The following is a Σ-protocol $< P, V >$ proposed by Schnorr [9] for proving the knowledge of discrete logarithm, w, for a common input of the form (p, q, g, h) such that $h = g^w \ mod \ p$, where on a security parameter n, p is a uniformly selected n-bit prime such that $q = (p-1)/2$ is also a prime, g is an element in $\mathbf{Z}_p^*$ of order q. It is also actually the first efficient Σ-protocol proposed in the literature.

- P chooses r at random in $\mathbf{Z}_q$ and sends $a = g^r \ mod \ p$ to V.
- V chooses a challenge e at random in $\mathbf{Z}_{2^k}$ and sends it to P. Here, k is fixed such that $2^k < q$.
- P sends $z = r + ew \ mod \ q$ to V, who checks that $g^z = ah^e \ mod \ p$, that p, q are prime and that g, h have order q, and accepts iff this is the case.

Σ-Protocol for RSA [7]. Let n be an RSA modulus and q be a prime. Assume we are given some element $y \in Z_n^*$, and P knows an element w such that $w^q = y \mod n$. The following protocol is a Σ-protocol for proving the knowledge of q-th roots modulo n.

- P chooses r at random in Z_n^* and sends $a = r^q \mod n$ to V.
- V chooses a challenge e at random in Z_{2^k} and sends it to P. Here, k is fixed such that $2^k < q$.
- P sends $z = rw^e \mod n$ to V, who checks that $z^q = ay^e \mod n$, that q is a prime, that $gcd(a, n) = gcd(y, n) = 1$, and accepts iff this is the case.

The OR-proof of Σ-protocols [3]. One basic construction with Σ-protocols allows a prover to show that given two inputs x_0, x_1, it knows a w such that either $(x_0, w) \in R_0$ or $(x_1, w) \in R_1$, but without revealing which is the case. Specifically, given two Σ-protocols $\langle P_b, V_b \rangle$ for R_b, $b \in \{0, 1\}$, with random challenges of, without loss of generality, the same length k, consider the following protocol $\langle P, V \rangle$, which we call Σ_{OR}. The common input of $\langle P, V \rangle$ is (x_0, x_1) and P has a private input w such that $(x_b, w) \in R_b$.

- P computes the first message a_b in $\langle P_b, V_b \rangle$, using x_b, w as private inputs. P chooses e_{1-b} at random, runs the SHVZK simulator of $\langle P_{1-b}, V_{1-b} \rangle$ on input (x_{1-b}, e_{1-b}), and let $(a_{1-b}, e_{1-b}, z_{1-b})$ be the simulated conversation. P now sends a_0, a_1 to V.
- V chooses a random k-bit string e and sends it to P.
- P sets $e_b = e \oplus e_{1-b}$ and computes the answer z_b to challenge e_b using (x_b, a_b, e_b, w) as input. He sends (e_0, z_0, e_1, z_1) to V.
- V checks that $e = e_0 \oplus e_1$ and that both (a_0, e_0, z_o) and (a_1, e_1, z_1) are accepting conversations with respect to (x_0, R_0) and (x_1, R_1), respectively.

Theorem 1. [4] *The above protocol Σ_{OR} is a Σ-protocol for R_{OR}, where $R_{OR} = \{((x_0, x_1), w) | (x_0, w) \in R_0 \text{ or } (x_1, w) \in R_1\}$. Moreover, for any malicious verifier V^*, the probability distribution of conversations between P and V^*, where w satisfies $(x_b, w) \in R_b$, is independent of b. That is, Σ_{OR} is perfectly witness indistinguishable.*

2.2 Description of Protocol

Let X and Y be two parties, and let f_X and f_Y be two one-way functions that admit Σ-protocols. The following description of protocol is taken from [4, 2], in which X plays the role of identification prover and Y plays the role of identification verifier.

Key Generation. On a security parameter n, randomly select x_X and x_Y of length n each in the domains of f_X and f_Y respectively, compute $pk_X = f_X(x_X)$ and $pk_Y = f_Y(x_Y)$. pk_X and pk_Y are the public-keys of X and Y respectively and x_X and x_Y are their corresponding secret-keys.

The ID Protocol. In order to identify himself to the *intended* verifier Y with public-key pk_Y, X proves to Y that he knows either the preimage of pk_X (i.e. x_X) or the preimage of pk_Y (i.e. x_Y), by executing the Σ_{OR}-protocol on common input (pk_X, pk_Y). We denote by a_{XY}, e_{XY}, z_{XY} the first, the second and the third message of the Σ_{OR}-protocol respectively.

3 Two Man-in-the-Middle Attacks

In this section, we show two attacks on the Cramer-Damgård ID scheme in some reasonable settings: one replaying attack and one interleaving attack.

3.1 The Replaying Attack

As shown in [2, 4], the intended-verifier property of the Cramer-Damgård ID scheme prevents a malicious verifier to pass as the prover to another *different* verifier. But, we observe that a simple replaying attack enables an adversary (the man-in-the-middle) to identify himself as the (honest) verifier to the (honest) prover. In other words, the Cramer-Damgård ID scheme suffers from the man-in-the-middle attack when it is used for mutual identification purpose between two players X and Y, in which both X and Y identify themselves to each other concurrently with reversed playing role in the two concurrent protocol executions.

Now, suppose X (with public-key pk_X) is identifying himself to Y (with public-key pk_Y) and an adversary A (i.e. the man-in-the-middle) controls the communication channel between X and Y and wants to identify himself as Y to X. The following is the message schedule of the adversary:

Move-1: After receiving a_{XY} from X, A sets $a_{YX} = a_{XY}$ and sends a_{YX} back to X.

Move-2: After receiving the random challenge e_{YX} from X, A sets $e_{XY} = e_{YX}$ and sends back e_{XY} as the random challenge to X.

Move-3: After receiving z_{XY} from X, A sets $z_{YX} = z_{XY}$ and sends z_{YX} back to X.

Clearly, if X can successfully identify himself to Y (which means (a_{XY}, e_{XY}, z_{XY}) is an accepting conversation on (pk_X, pk_Y) with X playing the role of identification prover and Y playing the role of identification verifier), then (a_{YX}, e_{YX}, z_{YX}) is also an accepting conversation on (pk_Y, pk_X) with X playing the role of identification verifier and the adversary A playing the role of identification prover (which means that A has successfully impersonated himself as Y to X).

3.2 The Interleaving Attack

We consider a scenario in which two parties X (with public-key pk_X) and Y (with public-key pk_Y) identify each other internally, but they *externally* identify themselves as a group with public-key (pk_X, pk_Y) to outside parties (say, a third party T with public-key pk_T). That is, when X (or Y) identifies himself to an outsider party T, X (or Y) just convinces T that he is either X or Y without revealing exactly who he is. Specifically, X (or Y) convinces T that he knows the preimage of either pk_X or pk_Y or pk_T, by executing the Σ_{OR} on (pk_X, pk_Y, pk_T) with pk_X (or pk_Y respectively) as his private witness. We remark that this scenario is meaningful in certain applications. Now, suppose the honest player X is identifying himself to the honest player Y, then we show an interleaving attack that enables an adversary A (i.e. the man-in-the-middle who controls the communication channel between X and Y) to convince T that he is one member of the player group $\{X, Y\}$ (i.e. he is either X or Y). The following is the specification of the interleaving message schedule of A who is the man-in-the-middle between X and Y. We remark the interleaving attack is ingenious in comparison with the above simple replaying attack.

Move-1: After receiving a_{XY} from X, A first generates a simulated conversation that he knows the preimage of pk_T (by running the SHVZK simulator as shown in the description of Σ_{OR}). Denote by $(\hat{a}_T, \hat{e}_T, \hat{z}_T)$ the simulated transcript, where $\hat{e}_T$ is a random string. Then, A sends $(a_{XY}, \hat{a}_T)$ to T.

Move-2: After receiving the random challenge e_T from T, A sets $e_{XY} = e_T \oplus \hat{e}_T$, and sends e_{XY} to X as the random challenge in the protocol execution between X and Y.

Move-3: After receiving z_{XY} from X, A sends $(z_{XY}, \hat{z}_T)$ to T.

Note that from the point view of T: $(\hat{a}_T, \hat{e}_T, \hat{z}_T)$ is an accepting conversation on pk_T, (a_{XY}, e_{XY}, z_{XY}) is an accepting conversation on (pk_X, pk_Y) for proving the knowledge of the preimage of either pk_X or pk_Y, and furthermore $e_{XY} \oplus \hat{e}_T = e_T$. This means A has successfully identified himself to T as one member of the player group $\{X, Y\}$.

4 Concluding Remarks

Identification protocol is one of the major cryptographic applications, especially in E-commerce over the Internet, and the Cramer-Damgård intended-verifier ID scheme is a famous one (due to its conceptual simpleness and highly practical efficiency) that may have been employed in practice. Though the intended-verifier property is necessary to prevent man-in-the-middle attacks of certain types, but as shown in this work, the intended-verifier property (i.e. letting the verifier also have his public-key) brings other security issues. Note that the two attacks shown in this work are all related to the intended-verifier property. In particular, if the identification verifier (e.g. Y) has no public-key (say, pk_Y), but, rather *freshly* generates and sends the "public-key message" (i.e. pk_Y) to the identification prover in each invocation, then our attacks will not work. But a verifier

with a public-key suffers from other security vulnerabilities, as we mentioned in Section 1. We note that the security vulnerabilities we reported in this paper are not an incidental phenomenon. Actually, the underlying reason behind the above two attacks is just the subtleties and complexities of soundness notions of ZK protocols in public-key models when the verifier has his public-key. Specifically, Micali and Reyzin showed in [8] that for ZK protocols although an adversary cannot get more advantages by concurrent interactions than by sequential interactions in the standard model, but, the soundness notion in the public-key model (when the verifier has his public-key) turns out to be much subtler and more complex than that in the standard model [8]. In particular, they showed that in the public-key setting concurrent interactions are strictly more powerful to an adversary than only sequential interactions.

Acknowledgements. The first author is indebted to Moti Yung for his many valuable discussions and comments.

References

1. S. Bengio, G. Brassard, Y. Desmedt, C. Goutier and J. J. Quisquater. Secure Implementation of Identification Systems. *Journal of Cryptology*, 1991(4): 175-183, 1991.
2. R. Cramer and I. Damgård. Fast and Secure Immunization Against Adaptive Man-in-the-Middle Impersonation. In *Wa. Fumy (Ed.): Advances in Cryptology-Proceedings of EUROCRYPT 1997, LNCS 1233*, pages 75-87. Springer-Verlag, 1997.
3. R. Cramer, I. Damgård and B. Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *Y. Desmedt (Ed.): Advances in Cryptology-Proceedings of CRYPTO 1994, LNCS 839*, pages 174-187. Springer-Verlag, 1994.
4. I. Damgård. On Σ-protocols. A lecture note for the course of Cryptographic Protocol Theory at Aarhus University, 2003. Available from: http://www.daimi.au.dk/∼ivan/CPT.html
5. U. Feige, A. Fiat and A. Shamir. Zero-knowledge Proof of Identity. *Journal of Cryptology*, 1(2): 77-94, 1988.
6. A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *A. Odlyzko (Ed.): Advances in Cryptology-Proceedings of CRYPTO'86, LNCS 263*, pages 186-194. Springer-Verlag, 1986.
7. L. Guillou and J. J. Quisquater. A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing both Transmission and Memory. In *C. G. Gnther (Ed.): Advances in Cryptology-Proceedings of EUROCRYPT 1988, LNCS 330* , pages 123-128, Springer-Verlag, 1988.
8. S. Micali and L. Reyzin. Soundness in the Public-Key Model. In *J. Kilian (Ed.): Advances in Cryptology-Proceedings of CRYPTO 2001, LNCS 2139*, pages 542–565. Springer-Verlag, 2001.
9. C. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3): 24, 1991.