سیستمهای پرداخت الکترونیکی ایمن

دانشگاه شیراز

دانشکده آموزشهای الکترونیکی

عناوین مطالب

- مقدمه بر امنیت
  - مفهوم رمزنگاری
  - رمزنگاری و رمزگشایی
  - رمزنگاری متقارن
  - هش کردن یا خلاصه سازی پیغام
  - رمزنگاری کلید عمومی یا نامتقارن
  - امضای دیجیتال و پوشش

عناوین مطالب

- پروتکلهای پرداخت الکترونیکی
  - اکتورهای پرداخت کارت اعتباری
  - پروتکلهای iKP
  - پروتکل e-Cash
  - تراکنش الکترونیکی ایمن
  - پروتکل 3D-Secure

رمزنگاری

- رمزنگاری روشی برای محافظت در برابر انواع حملاتی است که ممکن است بر روی ارتباطات بین دو نقطه روی دهد.

رمزنگاری و رمزگشایی

- در حوزه عبارات رمزنگاری به پیغامی که توسط انسان قابل خواندن می باشد ***plaintext*** یا ***cleartext*** گفته می شود.

- فرآیند تغییر یک پیغام بگونه ای که اجزای تشکیل دهنده آن مخفی بمانند را رمزنگاری می گویند و پیغام حاصل از اینکار را ***ciphertext*** می نامند.

- رمزگشایی متن cipher را بعنوان ورودی دریافت می کند و متن plain اصلی را بازیابی می کند.

رمزنگاری و رمزگشایی

P: Plaintext

C: Ciphertext

- تابع رمزنگاری ***E*** بر روی ***P*** عمل می کند تا ***C*** را تولید کند:

$$C = E(P)$$

- در فرآیند معکوس، تابع رمزگشایی ***D*** بر روی ***C*** عمل می کند تا ***P*** را تولید کند.

$$P = D(C)$$

رمزنگاری و رمزگشایی

- ***الگوریتم رمزنویسی (پنهان نگاری):*** یک تابع ریاضی است که برای رمزنگاری و رمزگشایی استفاده می شود.

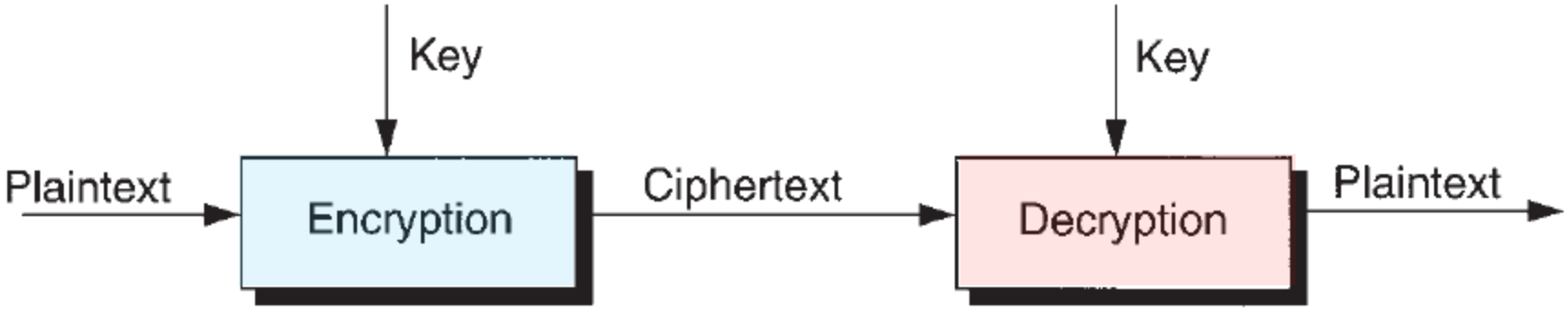
- تمامی الگوریتمهای رمزنگاری جدید از یک کلید استفاده می کنند که با ***K*** نشان داده می شود.

- مقدار این کلید توابع رمزنگاری و رمزگشایی را تحت تأثیر قرار می دهد.

$$E(K, P) = C$$

$$D(K, C) = P$$

رمزنگاری و رمزگشایی



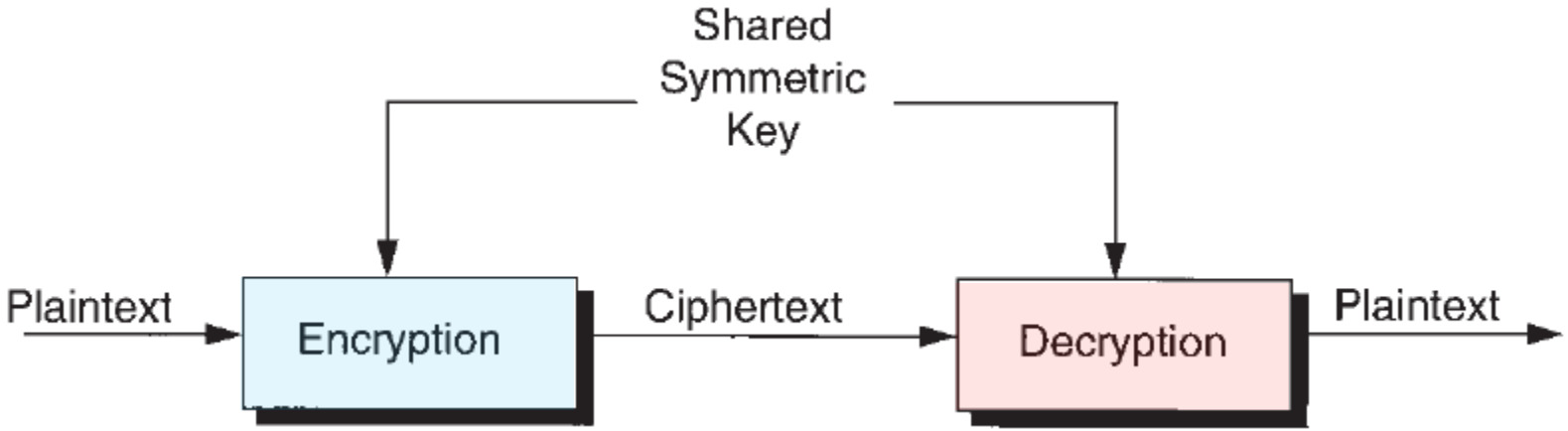
شکل 1: رمزنگاری و رمزگشایی با استفاده از یک کلید

رمزنگاری و رمزگشایی

- هدف اصلی رمز نویسی (پنهان نگاری) این است که متن اصلی از افراد متخاصم پنهان بماند.
- کشف رمز (***Cryptanalysis***) علم بازیابی متن اصلی بدون داشتن هیچ دانشی در مورد کلید می باشد.

رمزنگاری متقارن

- رمزنگاری متقارن نشان می دهد که هر دو طرف شرکت کننده در یک ارتباط بایستی در ابتدا یک کلید محرمانه را در اختیار داشته باشند.



شکل 2: عملکرد یک سیستم رمزنگاری متقارن

خلاصه سازی پیغام یا هش کردن

- در بسیاری از موارد چک کردن درستی پیغام مورد نیاز می باشد.
- یک راه برای تهیه درستی پیغام بدون نیاز به محرمانگی استفاده از تکنیکی است که با عنوان خلاصه سازی پیغام شناخته می شود.

خلاصه سازی پیغام یا هش کردن

- اینکار شامل اعمال کردن یک تابع هش یا خلاصه سازی بر روی یک پیغام (طولانی) و تولید یک پیغام خلاصه شده (کوتاه) می باشد.

- کلید محرمانه را می توان بر روی این هش اعمال کرد و نتیجه را می توان به همراه پیغام بر روی شبکه منتقل کرد.

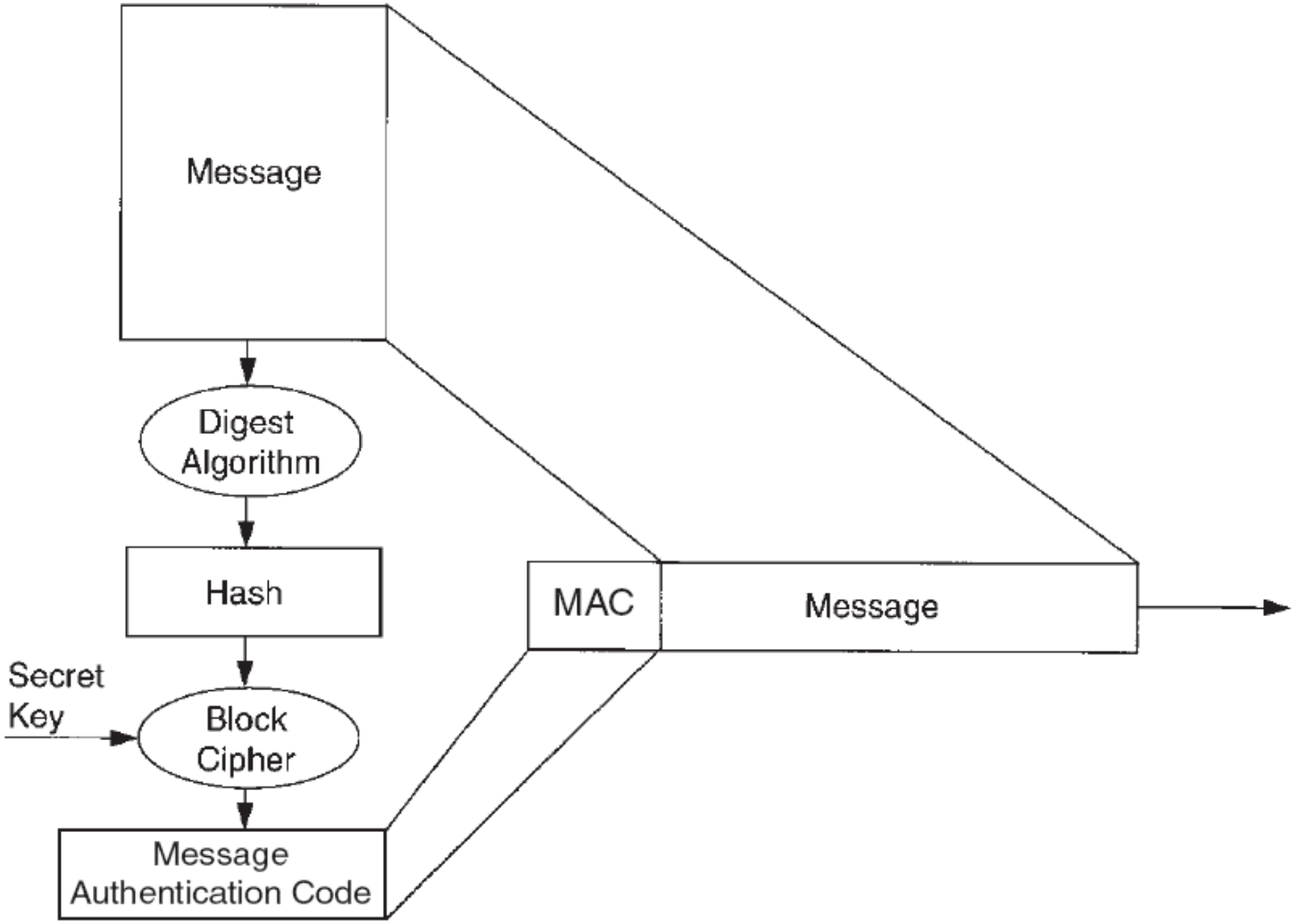
- سپس هش رمزنگاری می شود تا تبدیل به یک کد تصدیق پیغام (MAC) گردد که قبل از انتقال به پیغام ضمیمه می شود.

خلاصه سازی پیغام یا هش کردن

- وقتی پیغام رسید، دریافت کننده هش پیغام را با استفاده از الگوریتم یکسانی محاسبه می کند.

- اگر مقدار بدست آمده با MAC رمزگشایی شده که به همراه پیغام رسیده است انطباق داشته باشد، در اینصورت مطمئن خواهیم شد که پیغام در بین راه تغییر داده نشده است.

خلاصه سازی پیغام یا هش کردن



شکل 3: محاسبه کردن کد تصدیق پیغام

رمزنگاری نامتقارن یا کلید عمومی

- **مشکل سیستمهای رمزنگاری متقارن:** قبل از اینکه هر ارتباطی رخ دهد؛ هر دو عضو شرکت کننده باید یک کلید مشترک را به دست آورند.
- در رمزنگاری کلید عمومی، هر فرد یک جفت کلید را در اختیار دارد که کلید عمومی و کلید خصوصی نامیده می شوند.

رمزنگاری نامتقارن یا کلید عمومی

- کلید عمومی بطور گسترده منتشر می شود و همه می توانند آن را در اختیار داشته باشند اما کلید خصوصی محرمانه است و هرگز آشکار نمی شود.
- فرستده از کلید عمومی گیرنده استفاده می کند تا پیغام را رمزنگاری کند.
- سپس گیرنده از کلید خصوصی خود (SKB) استفاده می کند تا پیغام را رمزگشایی کند.
- هر کس که به کلید عمومی گیرنده دسترسی داشته باشد می تواند یک پیغام رمز شده را برای او ارسال کند اما هیچ کسی غیر از دریافت کننده نمی تواند آن را رمزگشایی کند.

ویژگیهای سیستم رمزنگاری کلید عمومی

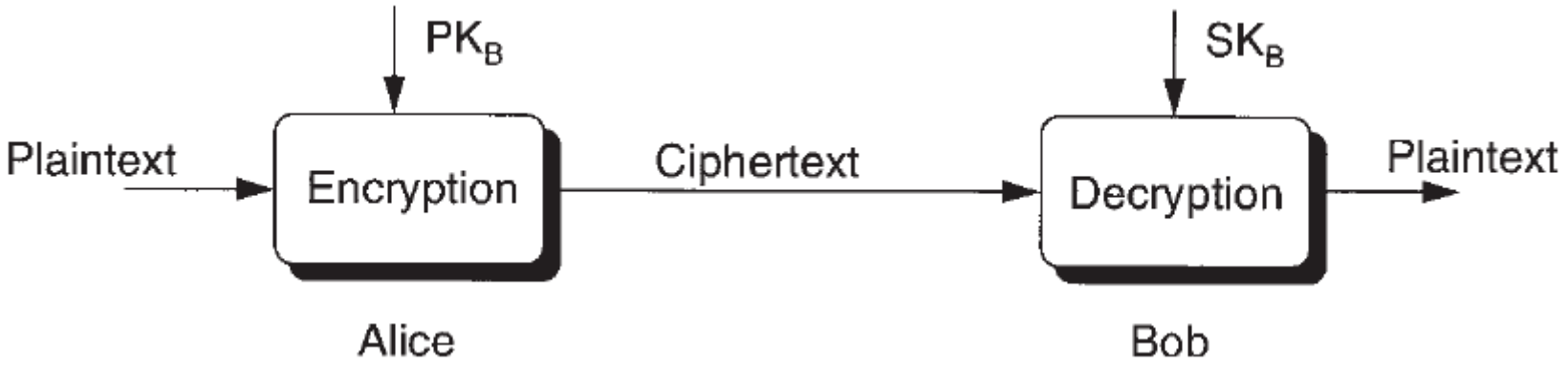
- اگر پس از رمزنگاری پیغام M بر روی آن رمزگشایی انجام شود همان پیغام اولیه بدست خواهد آمد.

$$SK\ (PK\ (M)) = M$$

- با داشتن PK و SK به سهولت می توان رمزنگاری و رمزگشایی را بطور متناظر انجام داد.

- با منتشر کردن عمومی PK، کاربر روش ساده ای را برای محاسبه SK آشکار نخواهد ساخت.

ویژگیهای سیستم رمزنگاری کلید عمومی



شکل 4: سیستم رمزنگاری کلید عمومی

امضاهای دیجیتال و پوشش

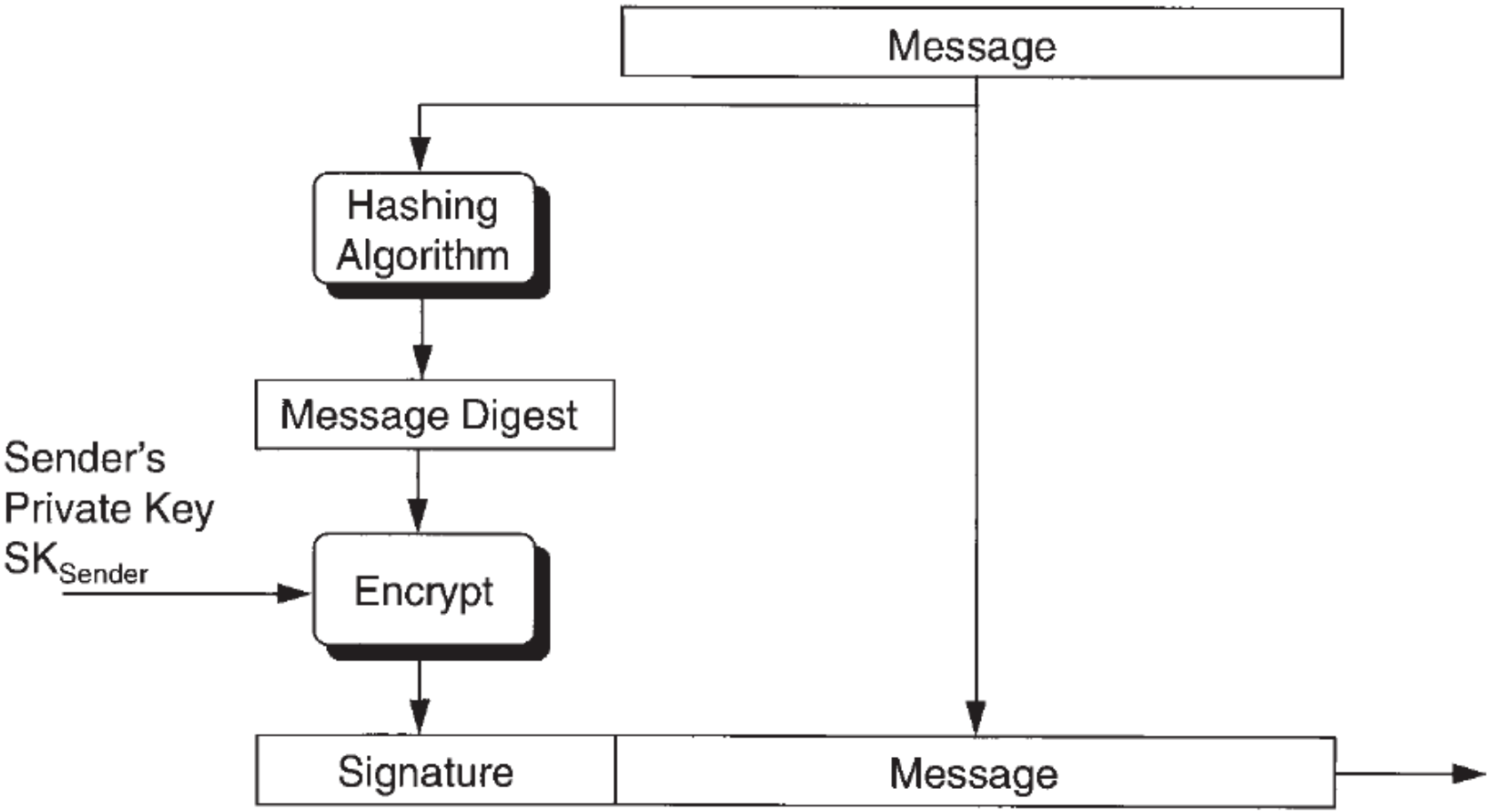
- تصدیق پیغام مورد توجه می باشد.
- **راه حل**: یک خلاصه پیغام را با استفاده از الگوریتمهایی مانند MD5 و SHA محاسبه کنید و سپس کلید خصوصی فرستنده را بر روی آن اعمال کنید.
- مقدار حاصل را می توان بعنوان امضای دیجیتال در نظر گرفت و قبل از انتقال به پیغام ضمیمه می شود.

امضاهای دیجیتال و پوشش

- در مقصد، گیرنده از الگوریتم یکسانی برای تولید خلاصه پیغام استفاده می کند و با استفاده از کلید عمومی فرستنده تأیید می کند که خلاصه پیغام محاسبه شده با امضای رمزگشایی شده انطباق دارد.

- در مورد یک انطباق، گیرنده می تواند مطمئن شود که پیغام از یک فرستنده مورد تأیید ناشی شده است و در طول انتقال تغییر نیافته است.

امضاهای دیجیتال و پوشش



شکل 5: اضافه کردن امضای دیجیتال به پیغام قبل از انتقال

امضاهای دیجیتال و پوشش

- اگر نیاز به محرمانگی پیغام داشته باشیم بایستی پیغام پوشانده شود.

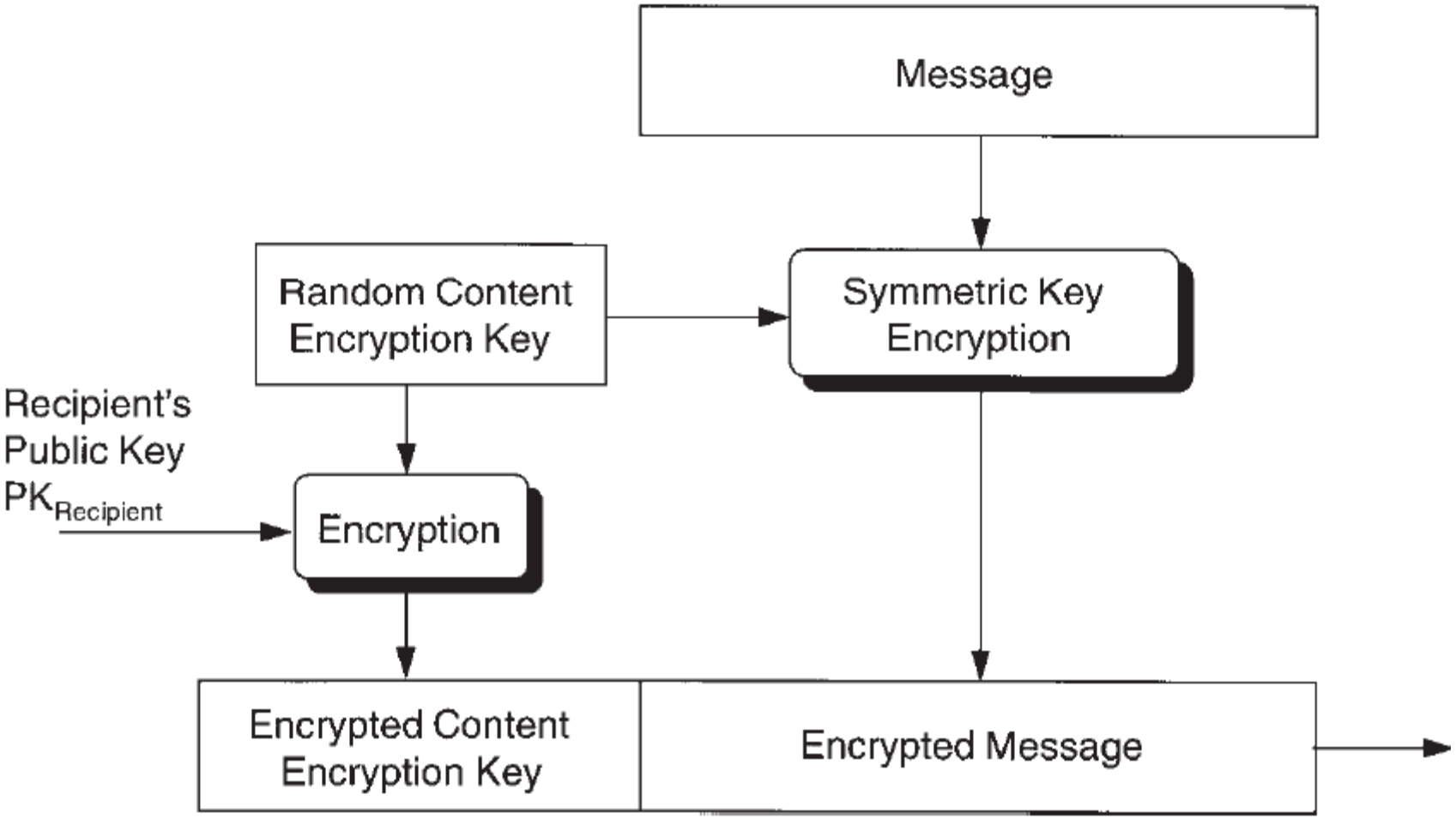
- برای رسیدن به این هدف، فرستنده می تواند یک کلید را بطور تصادفی تولید کند و می تواند از این کلید پیغام به همراه یک الگوریتم رمزنگاری سریع برای رمزنگاری پیغام استفاده کند.

- برای ارسال این کد پیغام به دریافت کننده، این کلید با استفاده از کلید عمومی دریافت کننده رمز می شود و به پیغام ارسالی ضمیمه می شود.

امضاهای دیجیتال و پوشش

- وقتیکه پیغام رسید، دریافت کننده از کلید خصوصی اش استفاده می کند تا کلید رمزنگاری را بدست آورد، در اینصورت وی این امکان را بدست می آورد تا متن دریافت شده را بصورت خوانا تبدیل کند.

امضاهای دیجیتال و پوشش



شکل 6: پوشش یک پیغام برای دریافت کننده

پروتکلهای پرداخت ایمن

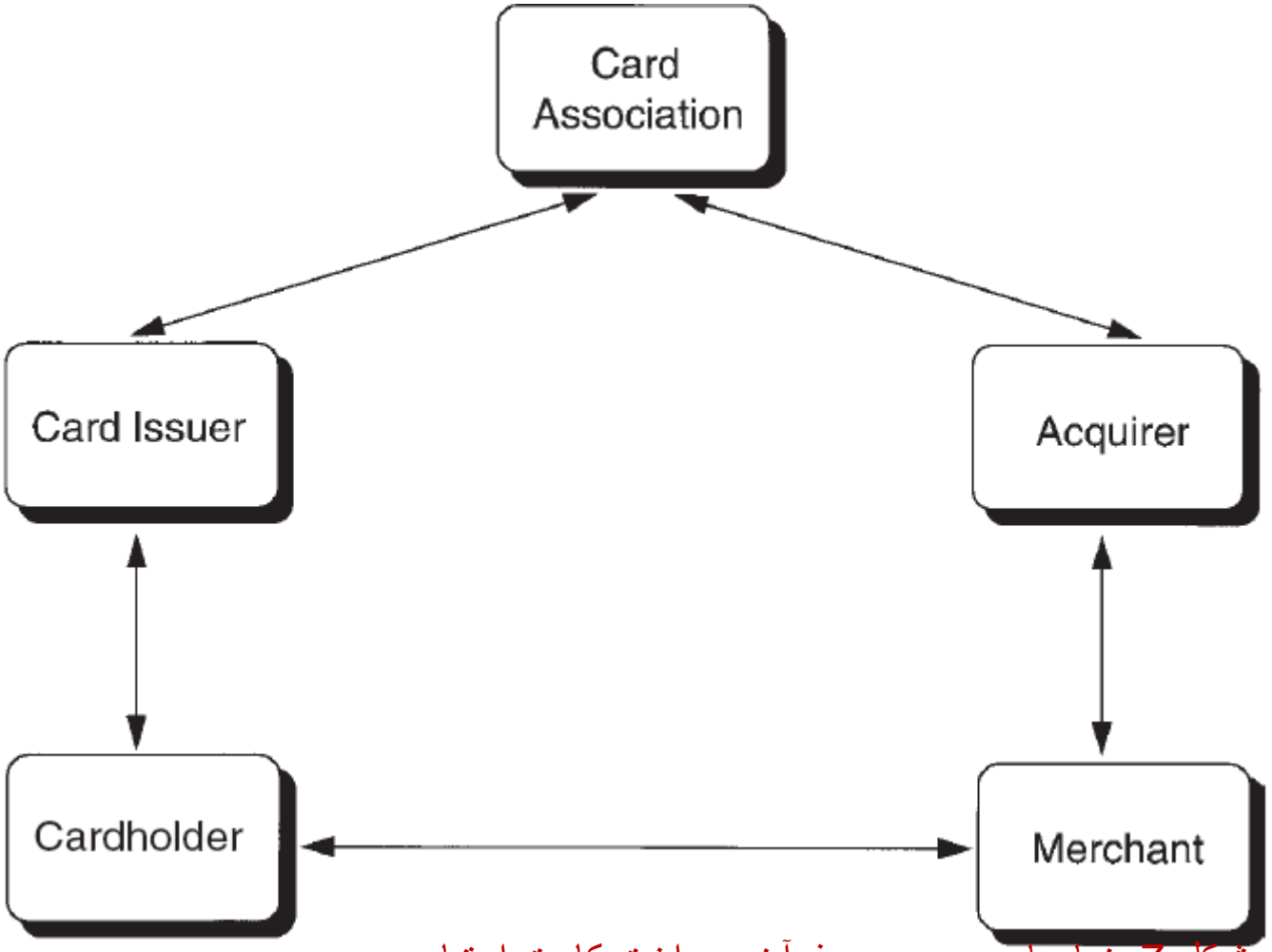
اکتورهای پرداخت کارت اعتباری

- بانکهایی که عضو انجمن کارت ممکن است بعنوان صادر کننده کارت به افراد یا کسب و کارهای مشتریانشان عمل کنند.
  - اینکار شامل ارائه یک کارت و نگهداری حساب کارت اعتباری برای افراد می باشد که به وسیله آن بتوانند تراکنشهای ارسال شده از سوی آنها را مدیریت کنند.
- بانک دیگری یا همان بانک بعنوان دریافت کننده برای مشتریان یا کسانی که می خواهند پرداختهای کارت اعتباری را دریافت کنند عمل می کند.
  - اینکار شامل تهیه تجهیزات و/یا نرم افزاری است تا پرداختهای رخ داده در محل بازرگان را پردازش کند.

اکتورهای پرداخت کارت اعتباری

- آرایش مورد نیاز برای تأیید آنلاین تراکنشها و همچنین سیاست برای نیازهای تأیید آنلاین به وسیله دریافت کننده آماده می شوند.
- خریدار: فردی است که دارنده کارت اعتباری است و خریدهای خود را از بازرگان انجام می دهد.
- بازرگان: همان فروشنده است که محصولات خود را در اختیار مشتری قرار می دهد.

نهادهای موجود در یک تراکنش کارت اعتباری مرسوم



شکل 7: نهادهای موجود در فرآیند پرداخت کارت اعتباری

پروتکلهای پرداخت iKP

- iKP خانواده ای از پروتکلهای پراخت ایمن می باشد که به وسیله IBM توسعه داده شده اند.
- پروتکلهای iKP بر مبنای رمزنگاری کلید عمومی می باشند و تفاوت آنها در تعداد شرکایی است که جفتهای کلید عمومی اشان را در اختیار دارند.
- این تعداد نشان دهنده نامی است که برای یک پروتکل خاص در نظر گرفته می شود:
  - 1KP: پروتکل 1KP بر مبنای زیرساخت امنیتی است که تقریباً امروزه وجود دارد.
  - 2KP، 3KP: نسخه های توسعه یافته 1KP می باشند که در آنها زیرساختهای تصدیق پیچیده تری استفاده می شود.
- هر چه تعداد بخشهایی که جفتهای کلید عمومی را دارا می باشند بیشتر باشد، سطح امنیت تهیه شده افزایش می یابد.

پروتکلهای پرداخت iKP

- در حال حاضر تأکید جاری این پروتکلها بر روی پرداختهای کارت اعتباری می باشد.
- نهادهای شرکت کننده در این سیستم عبارتند از:
  - خریدار
  - بازرگان
  - بانک بازرگان: بعنوان دریافت کننده شناخته می شود زیرا برگه های شارژ کاغذی را از بازرگان دریافت می کند.
  - بانک خریدار: با نام صادر کننده شناخته می شود زیرا کارتهای اعتباری را برای کاربران صادر می کند.

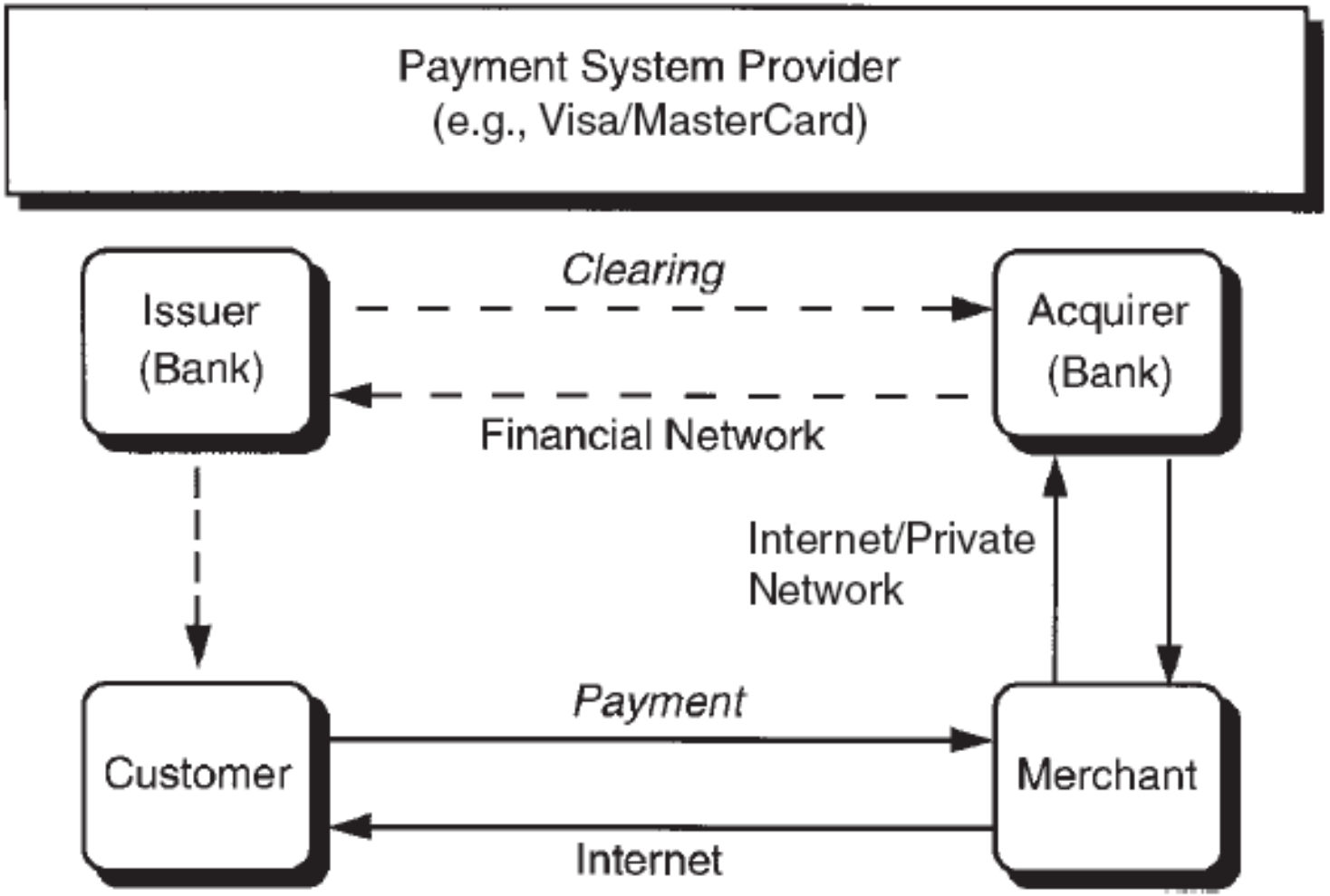
پروتکلهای پرداخت iKP

- مجموعه پروتکلهای iKP پروتکلهای خرید نیستند.
  - پروتکلها، رمزنگاری اطلاعات سفارش را انجام نمی دهند و فرض می کنند که جزئیات سفارش و قیمت بین مشتری و بازرگان توافق شده است.
- عملکرد اصلی آنها اینست که تراکنشهای پرداخت بین بخشهای مختلف را امکانپذیر سازند.
- این مسأله به پروتکلها اجازه می دهد تا با مکانیسمهای مختلف browsing سازگاری یابند.

پروتکلهای پرداخت iKP

- دریافت کننده بعنوان درگاهی بین اینترنت و شبکه های مالی موجود که تراکنشهای بین بانک ها را پشتیبانی می کنند عمل می کند.
- پروتکلهای iKP تنها با تراکنشهای پرداخت سر و کار دارند.
- بخشهای اصلی شرکت کننده در تراکنش عبارتند از:
  - مشتری ,(C)
  - بازرگان ,(M)
  - درگاه دریافت کننده.(A)

پروتکلهای پرداخت iKP



شکل 8: پروتکل پرداخت

چارچوب پروتکلهای iKP

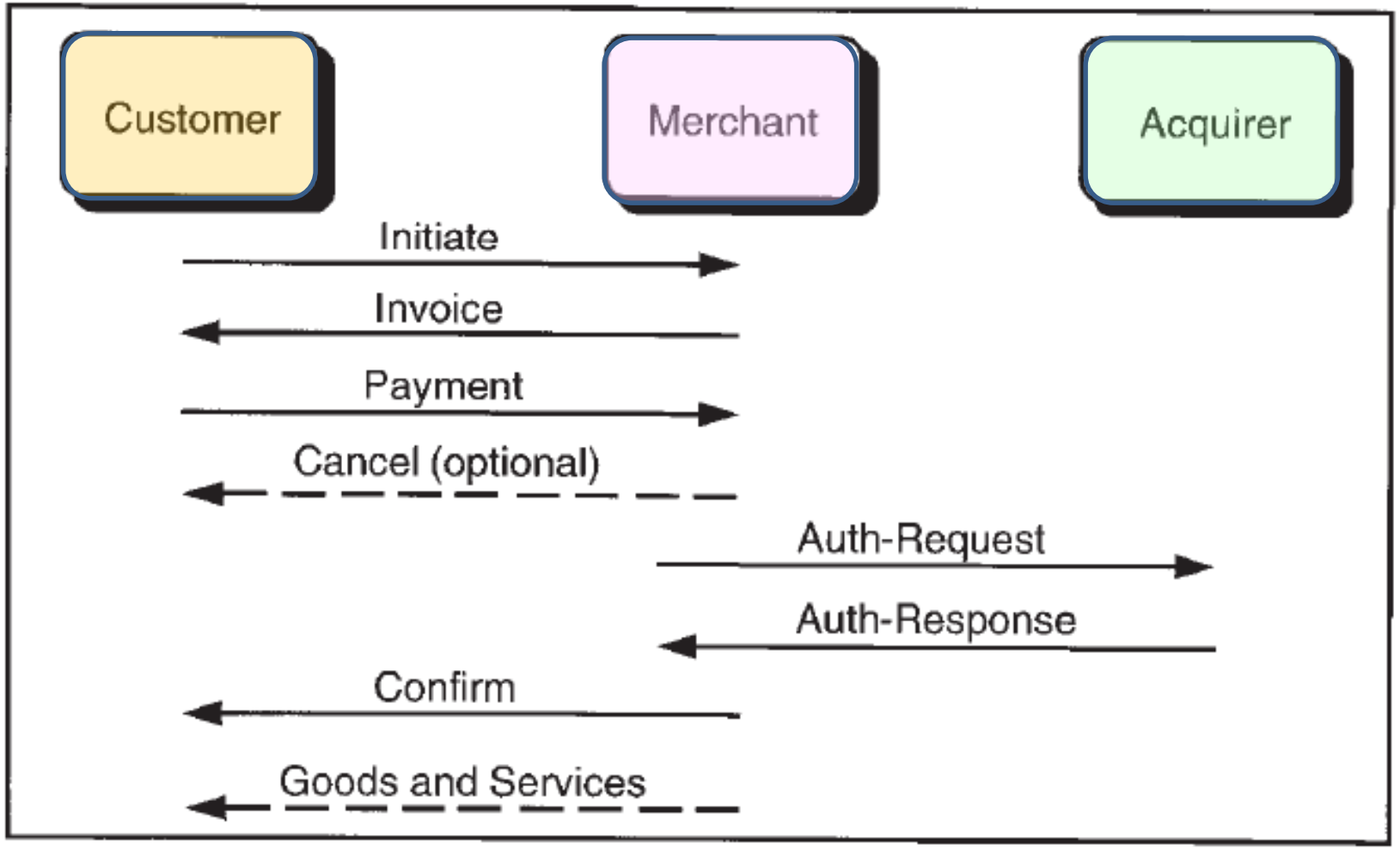
- گامهای پروتکل iKP عبارتند از:

1. ***آغاز:*** مشتری جریان پروتکل را آغاز می کند.
2. ***صورتحساب:*** بازرگان با تهیه یک صورتحساب پاسخ می دهد.
3. ***پرداخت:*** مشتری یک دستور پرداخت را تولید کرده و آن را برای بازرگان می فرستد.
4. ***لغو:*** بازرگان می تواند از ادامه پردازش تراکنش صرفنظر کند.

چارچوب پروتکلهای iKP

5. ***درخواست احراز هویت:*** بازرگان یک درخواست احراز هویت را برای دریافت کننده می فرستد.
6. ***پاسخ احراز هویت:*** دریافت کننده از سیستمهای تسویه حساب شبکه شده موجود استفاده می کند تا احراز هویت را بدست آورد و پاسخ احراز هویت را بر می گرداند.
7. ***تأیید:*** بازرگان پاسخ امضاء شده دریافت کننده و هر پارامتر اضافه دیگری را برای مشتری ارسال می کند.

چارچوب پروتکلهای iKP



شکل 9: مراحل اصلی پروتکل iKP

کمیتهای موجود در پروتکلهای iKP

Item	Description
CAN	Customer's account number (e.g., credit card number)
ID_M	Merchant ID; identifies merchant to acquirer
TID_M	Transaction ID; uniquely identifies the transaction
DESC	Description of the goods; includes payment information such as credit card holder's name and bank identification number
$SALT_C$	Random number generated by C; used to randomize DESC and thus ensure privacy of DESC on the M to A link
$NONCE_M$	Random number generated by a merchant to protect against replay
DATE	Merchant's current date/time
PIN	Customer's PIN which, if present, can be optionally used in 1KP to enhance security

جدول 1: کمیتهای مورد استفاده در iKP

کمیتهای موجود در پروتکلهای iKP

Item	Description
Y/N	Response from card issuer; Yes/No or authorization code
R_C	Random number chosen by C to form CID
CID	A customer pseudo-ID which uniquely identifies C; computed as CID = $H(R_C, CAN)$
V	Random number generated in 2KP and 3KP by merchant; used to bind the Confirm and Invoice message flows

جدول 2: کمیتهای مورد استفاده در iKP

فیلدهای ترکیبی

Item	Description
Common	Information held in common by all parties: PRICE, ID_M, TID_M, DATE, $NONCE_M$, CID, H(DESC, $SALT_C$), [H(V)]
Clear	Information transmitted in the clear: ID_M, TID_M, DATE, $NONCE_M$, H(Common), [H(V)]
SLIP	Payment instructions: PRICE, H(Common), CAN, R_C, [PIN]
EncSlip	Payment instruction encrypted with the public key of the acquirer: PK_A (SLIP)
$CERT_X$	Public-key certificate of X
Sig_A	Acquirer's signature: SK_A[H(Y/N, H(Common))]
Sig_M	Merchant's signature in Auth-Request: SK_M[H(H(Common), [H(V)])]
Sig_C	Cardholder's signature: SK_C[H(EncSlip, H(Common))]

جدول 3: کمیتهای مورد استفاده در iKP

1KP

- تنها دریافت کننده نیاز دارد تا گواهی کلید عمومی (CERTA) را در اختیار داشته باشد و آن را انتشار دهد.
- رمزنگاری کلید عمومی تنها از جانب مشتری مورد نیاز می باشد درحالیکه رمزگشایی تنها از جانب دریافت کننده مورد نیاز می باشد.
- هم مشتری و هم بازرگان برای تأیید امضای تولید شده به وسیله دریافت کننده مورد نیاز می باشند.
- فرض می شود که هر عضو شرکت کننده در پروتکل مقداری اطلاعات شروع را در اختیار دارد.

اطلاعات مورد نیاز برای شروع 1KP

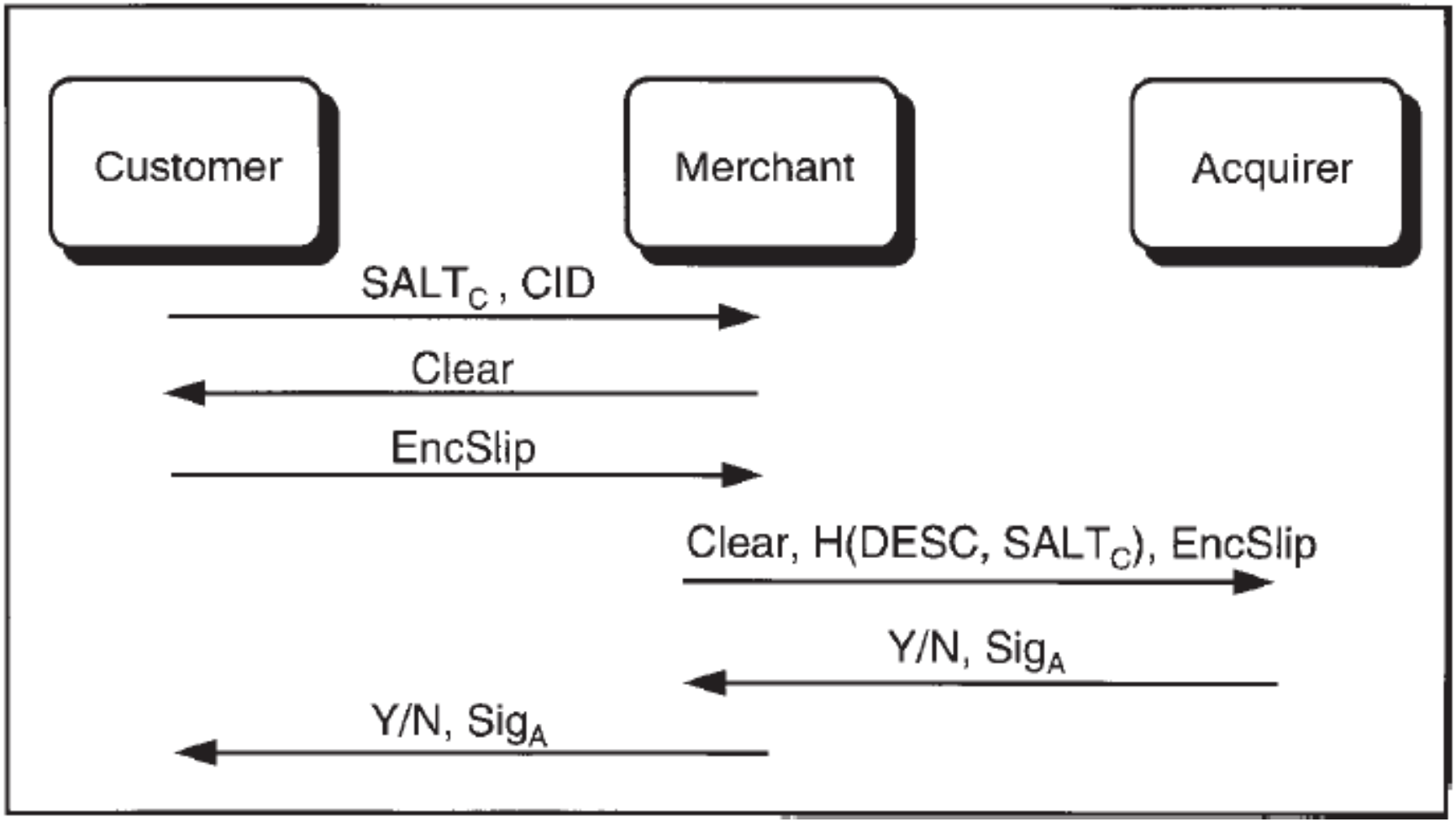
Actor	Information Items
Customer	DESC, CAN, PK_{CA}, [PIN], $CERT_A$
Merchant	DESC, PK_{CA}, $CERT_A$
Acquirer	SK_A, $CERT_A$

جدول 4: اطلاعات مورد نیاز برای شروع 1KP

1KP

- فرض می شود که خریدار و بازرگان بر روی توصیف محصولات (DESC) قبل از آغاز پروتکل به توافق رسیده اند.
- همچنین فرض می شود که هر دوی صاحب کارت و بازرگان کلید عمومی certification authority (CA) و گواهینامه دریافت کننده در اختیار دارند که از طریق آنها می توانند کلید عمومی آن را استخراج کنند.

سناریوی 1KP



شکل 10: سناریوی 1Kp

Initiate composition

1. مشتری یک ID صاحب کارت (CID) را ایجاد می کند که بصورت منحصربفرد مشتری را شناسایی می کند.
   - این ID با هش کردن شماره حساب مشتری (CAN) و یک مقدار تصادفی RC ایجاد می شود:

$$CID = H(R_C, CAN)$$

   - اینکار از فاش شدن CAN برای بازرگان جلوگیری می کند.

2. یک عدد تصادفی دیگر $SALT_C$ را تولید می کند که به وسیله بازرگان استفاده می شود تا اطلاعات توصیف محصول را بصورت تصادفی در آورد تا از آشکار شدن آن برای دریافت کننده جلوگیری کند.
3. مشتری دو کمیت را بعنوان بخشی از پیغام آغازین برای بازرگان ارسال می کند.

Initiate: ($SALT_C$, CID)

آغاز پردازش و ساخت صورتحساب

1. بازرگان $NONCE_M$ و DATE را تولید می کند. اینکار به دریافت کننده اجازه می دهد تا یک سفارش را بطور منحصر بفرد شناسایی کند. همچنین دریافت کننده یک ID تراکنش (TID_M) را انتخاب می کند تا زمینه را شناسایی کند.

2. $H(DESC, SALT_C)$ محاسبه می شود و حال در وضعیتی هستیم تا Common را ایجاد کنیم. سپس بازرگان H(Common) را محاسبه می کند.

3. Clear به همراه جریان Invoice برای مشتری فرستاده می شود. (فیلدهای موجود در Clear بعنوان cleartext ارسال می شوند).

Invoice: (ID_M, TID_M, DATE, $NONCE_M$, H (Common))

پردازش صورتحساب

1. مشتری DESC و $SALT_C$ را بعنوان بخشی از اطلاعات اولیه در اختیار دارد و $H(DESC, SALT_C)$ را محاسبه می کند. این کمیت توسط بازرگان در Common قرار داده می شود و توسط مشتری استفاده می شود تا در گام بعدی Common را ایجاد کند.

2. H(Common) محاسبه می شود و انطباق آن با مقدار H(Common) موجود در Clear که توسط بازرگان تولید می شود تأیید می شود. این عمل تأیید می کند که مشتری و بازرگان بر روی محتوای Clear با یکدیگر توافق دارند.

پردازش صورتحساب

3. دستور العمل پرداخت SLIP تولید می شود. سپس مشتری SLIP را با استفاده از کلید عمومی دریافت کننده رمز می کند.

4. SLIP رمز شده (EncSlip) بعنوان بخشی از جریان پرداخت ارسال می شود.

Payment: (PK_A (SLIP))

پردازش پرداخت

1. اگر به هر دلیلی بازرگان نخواهد فرآیند را ادامه دهد، بازرگان پیغام ***Cancel*** را برای مشتری ارسال می کند.
2. حال بازرگان احراز هویت را انجام می دهد:
   - ✓ بازرگان یک پیغام Auth-Request را ایجاد می کند.
   - ✓ بازرگان Clear و $H(DESC, SALT_C)$ را به همراه *EncSlip* که بعنوان بخشی از دستورالعمل پرداخت دریافت کرده است در پیغام قرار می دهد.
   - ✓ اینکار به دریافت کننده اجازه می دهد تا Common را ایجاد کند و H(Common) تولید شده به وسیله بازرگان و مشتری را تأیید کند.

Auth-Request: (EncSlip, Clear, $H(DESC, SALT_C)$)

پردازش Auth-Request

1. دریافت کننده مقدار H(Common) را از Clear همچنانکه به وسیله بازرگان محاسبه شده است استخراج می کند. این مقدار h1 نامیده می شود. همچنین ارسال دوباره با استفاده از مقادیر ID_M، TID_M، DATE، و $NONCE_M$ چک می شود.
2. دریافت کننده، EncSlip را رمزگشایی می کند و H(Common) را از Slip همچنانکه که به وسیله مشتری محاسبه شده است استخراج می کند. این مقدار h2 نامیده می شود.
3. دریافت کننده چک می کند آیا h1 = h2 . اینکار به او اطمینان می دهد که مشتری و بازرگان بر روی اطلاعات سفارش با یکدیگر به توافق رسیده اند.

پردازش Auth-Request

4. دریافت کننده، Common را با استفاده از فیلدهای مختلفی که در Auth-Request دریافت می کند ایجاد کرده و اطمینان می یابد که H(Common) = h1 = h2.
5. سپس دریافت کننده با صادر کننده کارت تماس می گیرد و گواهینامه (clearance) را برای تراکنش دریافت می کند.

Auth-Request

- به محض دریافت یک پاسخ از صادر کننده کارت، دریافت کننده یک امضای دیجیتال را بر روی پاسخ (Y/N) و H(Common) محاسبه می کند و Auth-Response را برای بازرگان می فرستد.

Auth-Response: (Y/N, Sig_A)

پردازش Auth-Response

1. بازرگان امضای (Sig_A) را تأیید می کند.

2. بازرگان پاسخ و امضای دریافت کننده را بعنوان بخشی از جریان پیغام Confirm برای مشتری می فرستد.

3. در مقابل مشتری، امضای دریافت کننده را تأیید می کند و تراکنش تکمیل می شود.

نقایص اصلی 1KP

- یک مشتری بجای استفاده از امضای دیجیتال، برای احراز هویت خودش در نزد بازرگان تنها از یک شماره کارت اعتباری و PIN استفاده می کند.
- بازرگان احراز هویت خود را نزد مشتری یا دریافت کننده انجام نمی دهد.
- نه بازرگان و نه مشتری دریافتهای غیر قابل انکار را برای تراکنش تهیه نمی کنند.

2KP

- در 2KP، علاوه بر دریافت کننده، بازرگان نیاز دارد تا یک جفت کلید عمومی را در اختیار داشته باشد تا کلید عمومی موجود در گواهینامه CERTM را در اختیار مشتری و دریافت کننده قرار دهد.

- اینکار مشتری و دریافت کننده را قادر می سازد تا هویت بازرگان را تأیید کنند.

المانهای جدید در صورتحساب

1. بازرگان یک عدد تصادفی V را تولید می کند و یک خلاصه پیغام H(V) را ایجاد می کند و سپس H(V) را به Clear اضافه می کند.

2. بازرگان از کلید محرمانه خود (SK_M) برای امضای H(Common) و H(V) استفاده می کند تا Sig_M را تولید کند.

3. همچنین بازرگان گواهینامه کلید عمومی $CERT_M$ خود را در ارائه می دهد بنابراین مشتری می تواند امضای Sig_M را تأیید کند.

اطلاعات مورد نیاز برای شروع 2KP

Actor	Information Items
Customer	DESC, CAN, PK_{CA}, $CERT_A$
Merchant	DESC, PK_{CA}, $CERT_A$, SK_M, $CERT_M$
Acquirer	PK_{CA}, SK_A, $CERT_A$

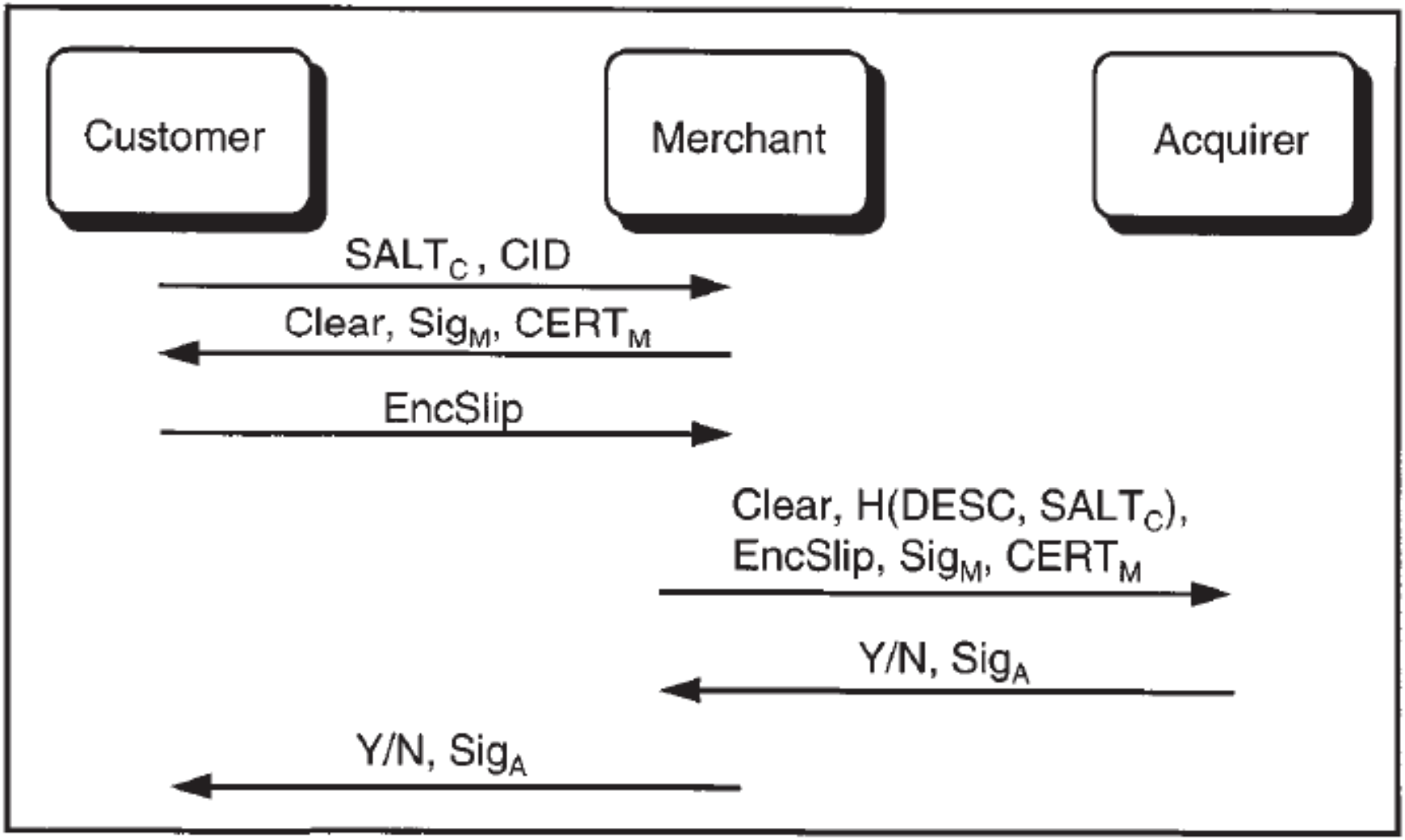
جدول 5: اطلاعات مورد نیاز برای شروع 2KP

2KP

- به محض دریافت صورتحساب، مشتری امضای بازرگان Sig_M را چک کرده و جریان پیغام پرداخت را مشابه با قبل تولید می کند.
- بازرگان همان امضای Sig_M که برای مشتری ارسال کرده بود را به همراه گواهینامه کلید عمومی اش ($CERT_M$) به Auth-Request ضمیمه می کند.
- acquirer قبل از تأیید اعتبار تراکنش امضای بازرگان را چک می کند.

- در نهایت، مقدار V در برای مشتری ارسال می شود تا وی در پاسخ H(V) را محاسبه کند و تأیید کند که این مقدار با مقدار موجود در صورتحساب (Invoice) انطباق دارد.

جریانهای پروتکل 2KP



شکل 11: جریانهای پروتکل 2KP

3KP

- در 3KP، تمامی اعضای شرکت کننده زوج کلید عمومی و گواهینامه های متناظر را در اختیار دارند.
- این امر قابلیت عدم انکار را بر روی تمامی تبادلات پروتکل فراهم می آورد.
- پروتکل بگونه ای تغییر داده شده است تا مشتری یک گواهینامه را برای بازرگان ارسال کند و سپس بازرگان آن را برای acquirer ارسال می کند.

تغییرات نسبت به 2KP

- حال مشتری گواهینامه کلید عمومی اش را بعنوان بخشی از پیغام Initiate برای بازرگان ارسال می کند.

- بعنوان بخشی از پرداخت، مشتری امضای خود (Sig_C) را برای بازرگان می فرستد. Sig_C با رمزنگاری EncSlip و H(Common) محاسبه می شود.

- بازرگان قادر خواهد بود امضای مشتری را تأیید کند زیرا در حال حاضر بازرگان گواهینامه مشتری ($CERT_C$) را در اختیار دارد.

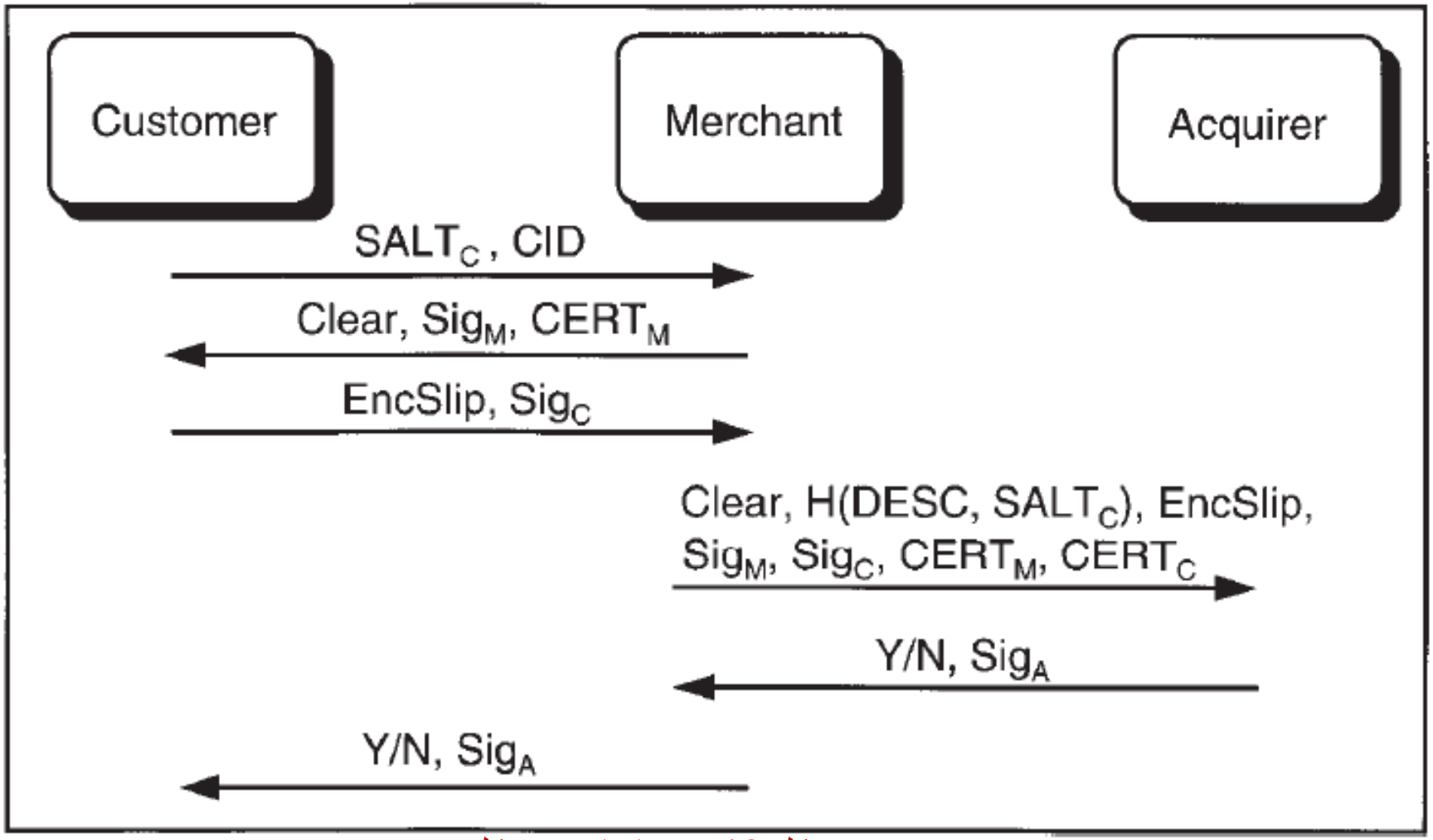
- بازرگان Sig_C را بعنوان بخشی از Auth-Request برای acquirer می فرستد و acquirer در پاسخ امضا را تأیید می کند.

اطلاعات مورد نیاز برای شروع 3KP

Actor	Information Items
Customer	DESC, CAN, PK_{CA}, SK_C, $CERT_A$
Merchant	DESC, PK_{CA}, $CERT_A$, SK_M, $CERT_M$
Acquirer	PK_{CA}, SK_A, $CERT_A$

جدول 6: اطلاعات مورد نیاز برای شروع 3KP

جریان پروتکل 3KP



شکل 12: جریانهای پروتکل 3KP

تراکنش الکترونیکی ایمن

- SET از قدرتمندترين پروتكلهاي پرداخت اينترنتي كارت اعتباري است كه در سال 1996 توسط شركتهاي بزرگ رايانه اي طراحي و در سال 1998 بوسيله دو شركت Visa و مستركارت عملياتي شد.

- قدرت امنيتي پروتكل SET بر مبناي دو عامل است:
  - استفاده از امضاي ديجيتال كه انكارناپذيري را به دنبال دارد.
  - امضاي دوگانه

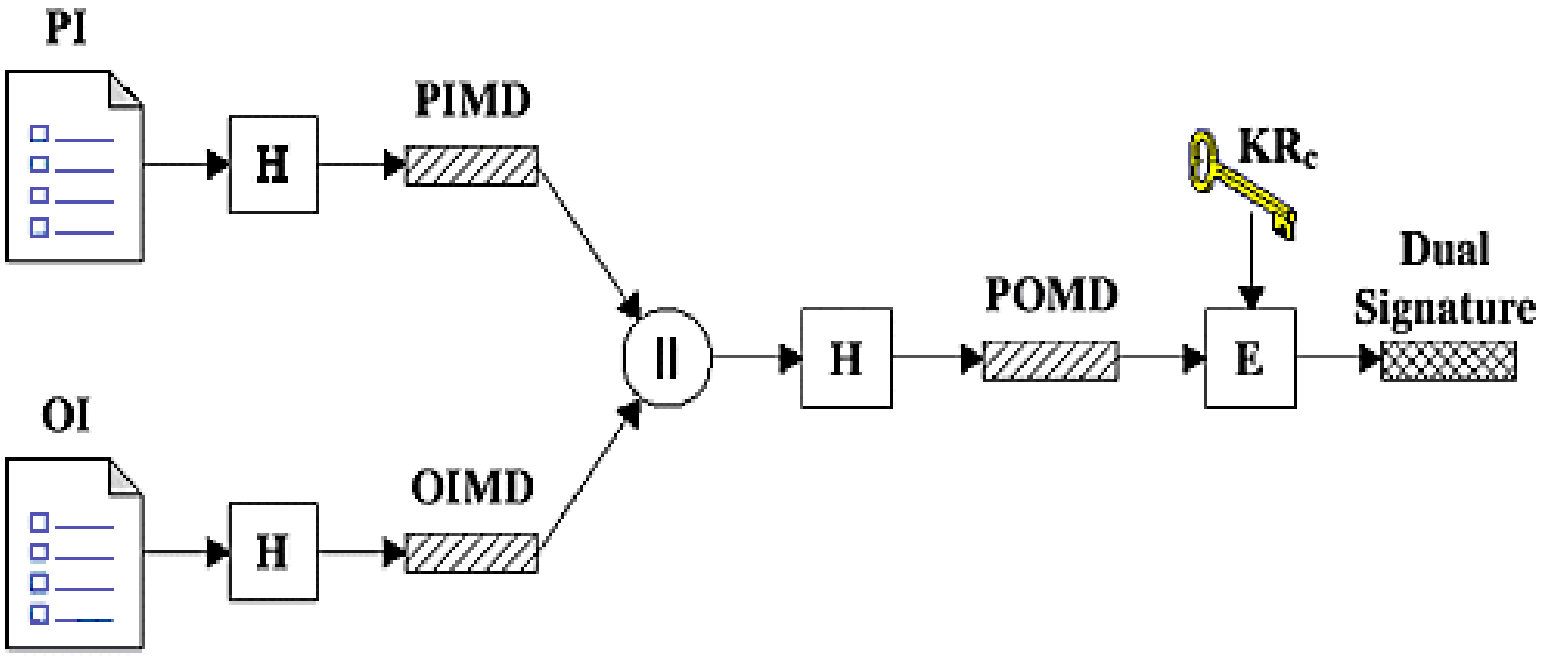
نکات مثبت در SET

- **محرمانگی اطلاعات:** یک ویژگی مهم و جالب SET اینست که از شناسایی شماره کارت اعتباری مشتری توسط بازرگان جلوگیری می کند و تنها بانک صادر کننده کارت می تواند شماره کارت اعتباری را در اختیار بگیرد.
- **یکپارچگی داده:** امضاهای دیجیتال RSA با استفاده از کدهای هش SHA-1 یکپارچگی پیغام را تهیه می کند.
- **تصدیق حساب دارنده کارت:** SET بازرگان را قادر می سازد تا تأیید کند که یک دارنده کارت یک کاربر قانونی یک شماره حساب کارت معتبر می باشد.
- **تصدیق بازرگان:** SET دارنده کارت را قادر می سازد تا تأیید کند که یک بازرگان دارای یک رابطه با یک مؤسسه مالی است که به وی اجازه می دهد تا پرداختهای کارت را بپذیرد.

موجودیتها در پروتکل SET

- **صاحب کارت:** یک صاحب کارت یک دارنده قانونی یک کارت پرداخت (مانند MasterCard، Visa و ...) می باشد که به وسیله یک بانک صادر کننده صادر شده است.
- **بازرگان:** بازرگان، فرد یا سازمانی است که محصولات و سرویس هایی را در اختیار دارد که آنها را به صاحب کارت می فروشد.
- **صادر کننده:** یک مؤسسه مالی می باشد (مانند یک بانک) که یک کارت را برای صاحب کارت تهیه می کند.
- **دریافت کننده:** یک مؤسسه مالی می باشد که حسابی را برای بازرگان ایجاد می کند و احراز هویتهای کارت پرداخت و پرداختها را پردازش می کند.
- **دروازه پرداخت:** عملکردی است که به وسیله دریافت کننده یا شخص ثالث طراحی شده ای است که پیغامهای پرداخت بازرگان را پردازش می کند.
- **Certification Authority (CA):** نهادی است مطمئن که گواهینامه های کلید عمومی را برای صاحبان کارتها، بازرگانان و دروازه های پرداخت تهیه می کند.

ویژگیهاي پروتکل SET-امضاي دوگانه (Dual Signature)



PI = Payment Information
OI = Order Information
H = Hash function (SHA-1)
|| = Concatenation

PIMD = PI message digest
OIMD = OI message digest
POMD = Payment Order message digest
E = Encryption (RSA)
KR_c = Customer's private signature key

شکل 13: امضای دوگانه

ويژگيهاي پروتكل SET-بررسي صحت امضا توسط بازرگان

- مشتري براي بازرگان امضاي دوگانه ، PIMD و گواهينامه خود را مي فرستد.
- بازرگان دو عبارت زير را محاسبه كرده و در صورت برابري، از صحت امضا اطمينان حاصل مي كند:

$$H\ (PIMD||H(OI))$$

و

$$D_{KUc}\ [DS]$$

- در اينجا مشتري دو پيام را به هم چسبانده و متصل بودن هر دو قابل اثبات است.

ويژگيهاي پروتكل SET-بررسي صحت امضا توسط بانك

- مشتري براي بانك امضاي دوگانه ، OIMD و گواهينامه خود را مي فرستد.
- بانك دو عبارت زير را محاسبه كرده و در صورت برابري، از صحت امضا اطمينان حاصل مي كند:

$$H\ (H(PI)||OIMD)$$

و

$$D_{KUc}\ [DS]$$

روند معامله در پروتکل SET

- معامله در چهار فاز انجام مي پذيرد:
  1. درخواست شروع
  2. پاسخ شروع
  3. درخواست خرید
  4. پاسخ خرید

درخواست شروع

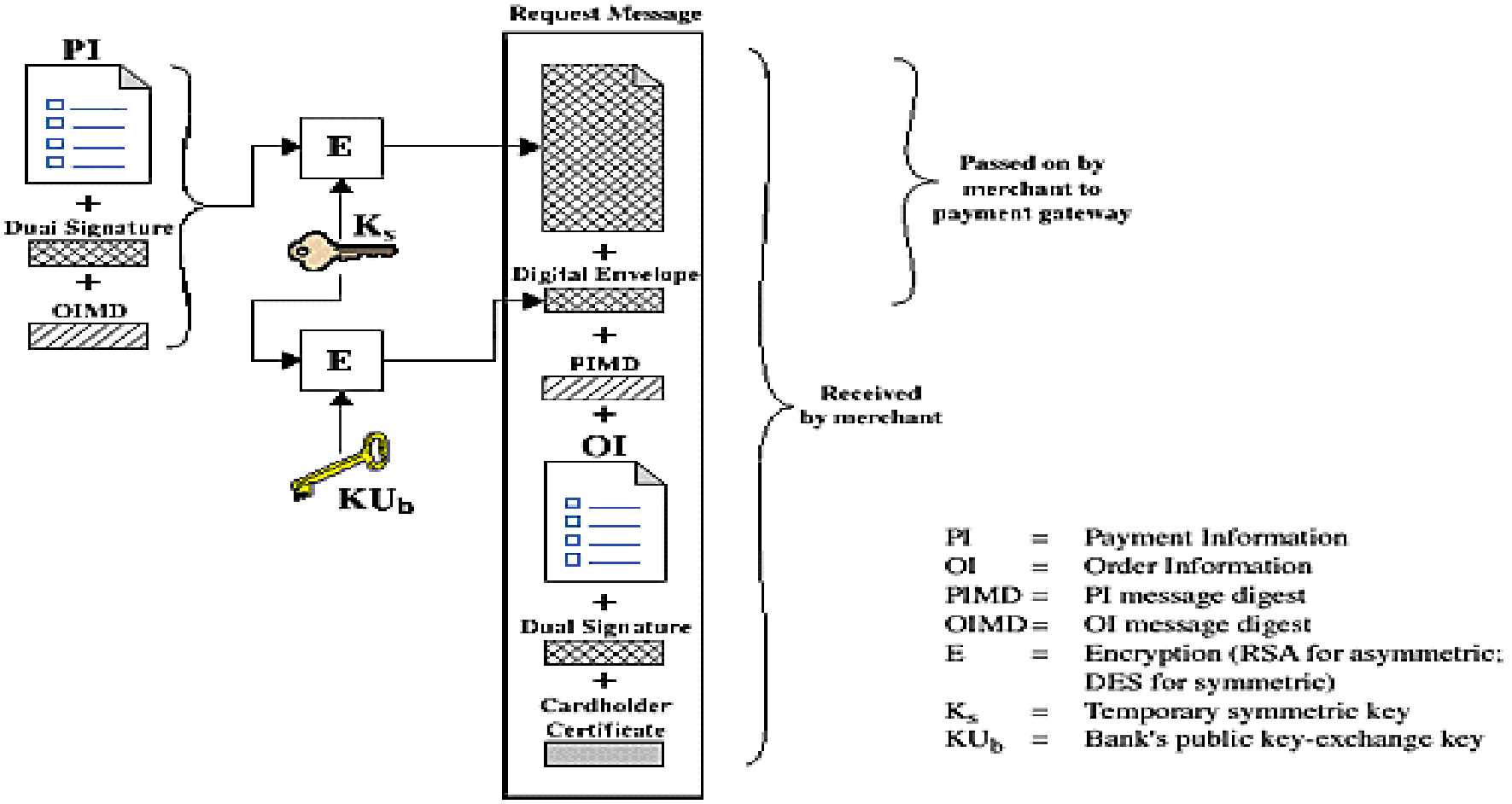
- در پيام درخواست شروع ،مشتري از بازرگان درخواست گواهينامه وي و درگاه پرداخت را مي كند.

- پیام شامل نوع کارت استفاده شده و ID مربوط به این خرید است.

پاسخ شروع

- بازرگان پيام پاسخ شروع را كه شامل ID معامله است را امضا كرده و به همراه هر دو گواهينامه براي مشتري مي فرستد.
- سپس مشتري OI و PI را تشكيل مي دهد و سپس پيام درخواست خريد را آماده مي كند.

روند معامله در پروتکل SET-پیام درخواست خرید



شکل 14: پیغام درخواست خرید

روند معامله در پروتکل SET-پیام پاسخ خرید

بازرگان پس از دريافت پيام مراحل زير را انجام مي دهد:

1. گواهینامه های صاحب کارت را با استفاده از امضاهای CA تأیید می کند.
2. امضای دوگانه را با استفاده کلید امضای عمومی مشتری تأیید می کند. اینکار این اطمینان را بوجود می آورد که سفارش در طول انتقال تغییر داده نشده است و همچنین با استفاده از کلید امضای خصوصی صاحب کارت امضاء شده است.
3. سفارش را پردازش می کند و اطلاعات پرداخت را برای احراز هویت به دروازه پرداخت ارسال می کند.
4. یک پاسخ خرید را برای صاحب کارت ارسال می کند.
5. مشتري امضاي بازرگان روي پيام را بررسي كرده و در صورت صحت يا عدم صحت، پيام متناظر را به مشتري نمايش مي دهد.

پروتكل SET چه مشكلاتي دارد؟

- نياز مشتري به يك نرم افزار حجيم كه گواهينامه اش روي آن نصب شده باشد.
- عدم امكان معامله مشتري در مكانهايي ديگر به جز رايانه خودش
- هزينه سنگين پياده سازي

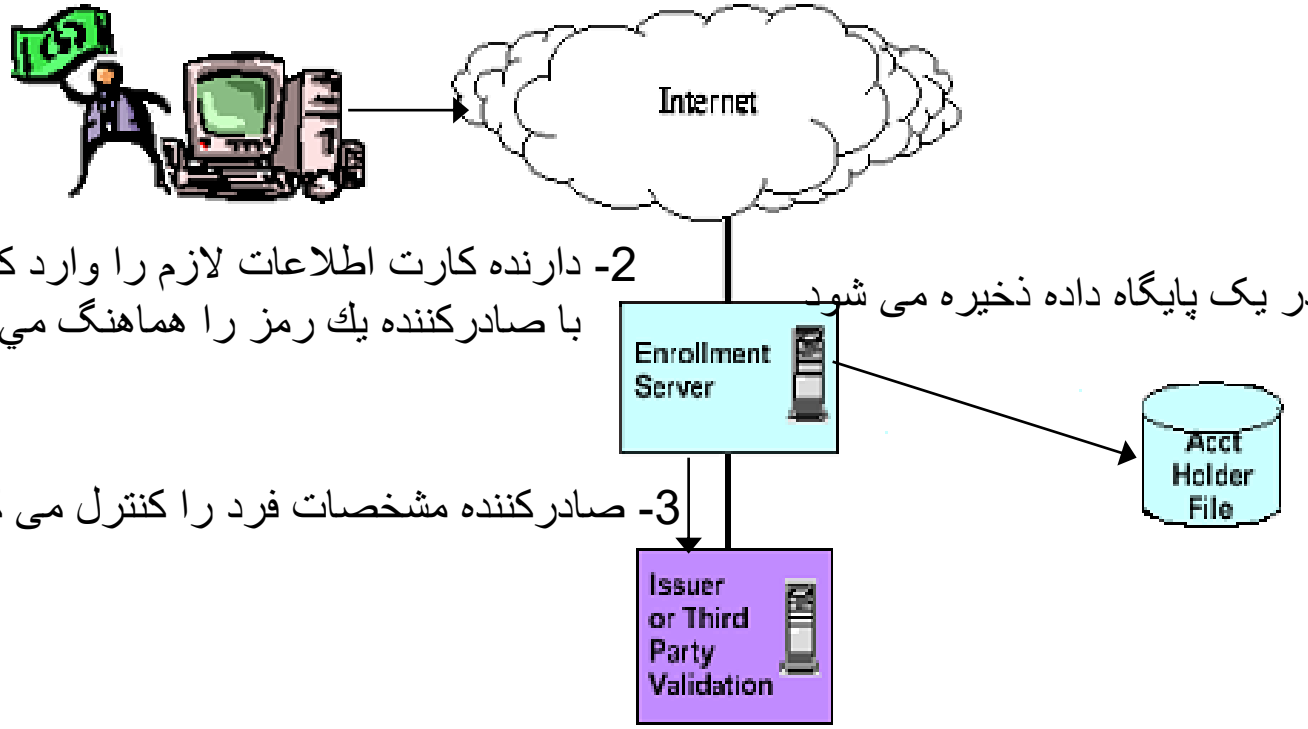
پروتکل 3D Secure

- پروتکل 3D Secure در سال 2001 طراحي شد و هم اكنون به عنوان پروتكل معتبر معاملات كارت اعتباري از طرف شركت ويزا و مستركارت ارائه شده است.

- اين پروتكل دو مرحله اساسي دارد كه عبارتند از:
  - مرحله ثبت نام
  - مرحله انجام معامله

پروتکل 3D Secure-عملیات ثبت نام

1- مشتري وارد سايت صادركننده مي شود



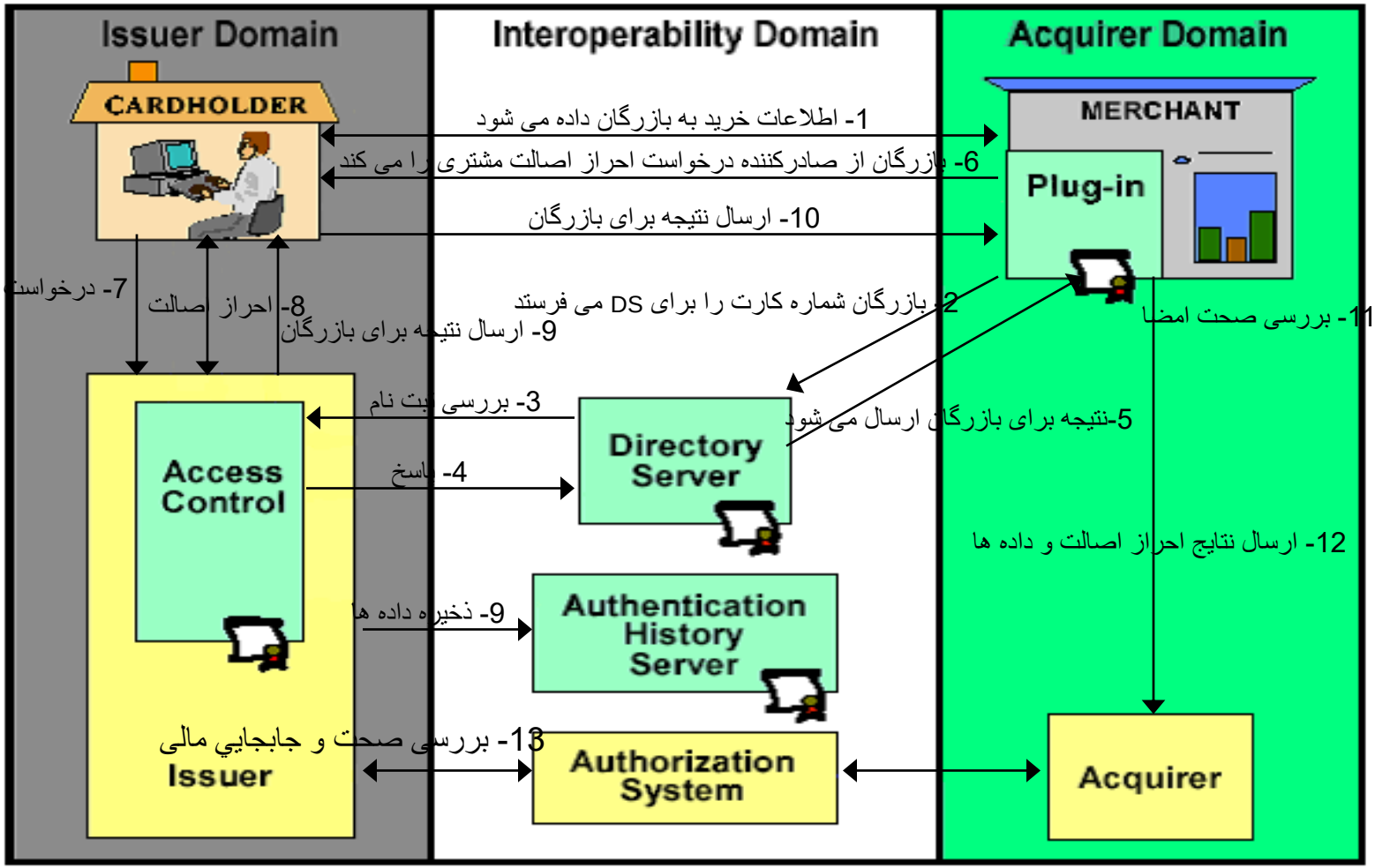
2- دارنده كارت اطلاعات لازم را وارد كرده و با صادركننده يك رمز را هماهنگ مي كند

3- صادرکننده مشخصات فرد را کنترل می کند

4- اطلاعات در یک پایگاه داده ذخیره می شود

شکل 15: عملیات ثبت نام

پروتکل 3D Secure



شکل 16: انجام معامله

نیازمندیهای امنیتی در سیستم 3D Secure

- کلیه مسیرها باید توسط SSL از شنود محافظت شوند. بجز مشتری، کلیه موجودیتها باید دارای گواهینامه های Client و Server برای SSL باشند.
- صادرکننده کارت باید پِیام نتیجه احراز اصالت مشتری را به همراه داده های دیگر توسط کلید خصوصی خود امضا کند. این امضا توسط بازرگان بررسی می گردد.
- احراز اصالت مشتری نه تنها از شماره کارت وی، بلکه از یک کلمه عبور که وارد صفحه صادرکننده می شود انجام می پذیرد. بنابراین بازرگان با داشتن شماره کارت وی قادر به انجام معامله نمی باشد.

ویژگیهای مثبت 3D Secure

- امکان انجام معامله توسط مشتری در هر مکان
- عدم نیاز مشتری به نرم افزار خاص برای انجام معامله
- کاهش بار ترافیکی مزاحم روی سرورهای صادرکننده های کارت
- امکان احراز اصالت مشتری مستقیما از طریق صادرکننده
- وجود یک حافظه مرکزی برای ذخیره اطلاعات در زمان مشاجره
- پیچیدگی کم

نقاط ضعف 3D Secure

- ارسال شماره کارت اعتباری برای بازرگان، دو مجهول معامله (که شماره کارت اعتباری و کلمه عبور است) را به یک مجهول کاهش داده است.
- امکان ایجاد یک صفحه تقلبی از طرف صادرکننده برای دریافت کلمه عبور مشتری وجود دارد.
- بازرگان نیاز به یک نرم افزار پر حجم دارد (Plug in)
- تعداد پیامها در این پروتکل زیاد و زمان انجام معامله زیاد است.
- مشتری از اطلاعات خریدی که برای صادرکننده ارسال می شود خبر ندارد.
- عدم گمنامی مشتری در معامله

پروتکل e-Cash

- دلايل متعددي وجود دارد كه ممكن است يك مصرف كننده به سمت استفاده از پولهاي تحت Web برود. برخي از اين دلايل در زير آورده شده است:
  - امنيت بيشتر مبادلات نسبت به كارتهاي اعتباري
  - توانايي خريدهاي اينترنتي با حجم پايين
  - گمنامي مبادلات و عدم امكان ايجاد ارتباط بين پول و شخص خريدار
  - هزينه كم مبادلات
- در پروتكل e-Cash، گمنامي خريدار مهمترين عامل در طراحي پروتكل است.

موجوديتهاي دخيل درپروتكل e-Cash

- بانکهایی که سکه ها را ضرب می کنند، سکه های موجود را تأیید می کنند و پول واقعی را برای e-cash مبادله می کنند.

- خریدارانی که در یک بانک حساب دارند و می توانند سکه های e-cash را از حسابهایشان برداشت کنند یا به آن واریز کنند.

- بازرگانانی که می توانند سکه های e-cash را برای پرداخت هزینه اطلاعات یا محصولات فیزیکی دریافت کنند. همچنین بازرگانان این امکان را خواهند داشت تا سرویس pay-out را برای شرایطی که می خواهند به یک مشتری سکه های e-cash را پرداخت کنند اجرا کنند.

امضاي كور، مبناي الگوريتم e-Cash

- مبنای پروتکل e-cash امضای کور می باشد.
- امضای کور نوعی از امضاء است که در آن امضاء کننده نمی تواند ببیند چه چیزی را امضاء می کند.

معرفي پروتكل e-Cash-چگونه مشتري پول را از حساب خود بيرون مي كشد؟

1- بانك ابتدا دو عدد اول بزرگ p و q را انتخاب كرده و حاصلضرب آنها (n = p*q) را بعنوان كليد عمومي در نظر مي گيرد.

2- آليس توسط نرم افزار CyberWallet خود يك عدد سريال X و يك ضريب كوركننده r را مي سازد.X يك عدد 100 رقمي است.

3- آليس X را از تابع يك طرفه f رد كرده و f(X) را مي سازد. سپس وي عدد زير را حساب مي كند:

$$B = ((f(x)* r) \mod n$$

فرض كنيد كه‌توان 5 در اينجا نشاندهنده 2 دلار باشد. (توانهاي مختلف نشانگر مقادير مختلف پول مي باشد. بطور مثال 1 دلار با عدد 3 ، 2 دلار با عدد 5 و ... مشخص مي شود) سپس آليس B را به بانك مي فرستد.

معرفي پروتكل e-Cash-چگونه مشتري پول را از حساب خود بيرون مي كشد؟

4-بانك ريشه پنجم B را مي گيرد يا بعبارتي D را بگونه اي محاسبه مي كند كه:

$$(D^5) \bmod n = B$$

براي اين منظور، ابتدا inv1 بگونه اي محاسبه مي شود كه:

$$(5*inv1) \bmod (p-1)(q-1) = 1$$

سپس:

$$D = (B^{inv1}) \bmod n$$

فرمول بالا مي تواند به صورت زير نوشته شود:

$$D = ((f(x)\ ^{(1/5)})*r \bmod n$$

بنابراين گرفتن ريشه پنجم B همانند اين است كه بانك پول را امضا كرده است. اين امضا مي تواند توسط دريافت كننده پول كنترل شود. نكته ديگر اين است كه بانك x و f(x) را ندارد . سپس بانك 2 دلار از حساب آليس بيرون آورده و D را براي آليس مي فرستد.

معرفي پروتكل e-Cash-چگونه مشتري پول را از حساب خود بيرون مي كشد؟

5- آليس با تقسيم D بر C، r را بدست مي آورد يا بعبارتي C را چنان پيدا مي كند كه:

$$D = (C*r) \bmod n$$

براي اين منظور، Inv2 بگونه اي محاسبه مي شود كه:

$$(r*inv2) \bmod n = 1$$

سپس:

$$C = (D*inv2) \bmod n$$

$$C= (f(x)^{(1/5)}) \bmod n$$

اكنون آليس X و C را دارد. زوج (X, C) با همديگر يك پول الكترونيكي را تشكيل مي دهند.بصورت عملي داده هاي يك پول الكترونيكي شامل نام بانك، مقدار پول ، X و C مي باشد.

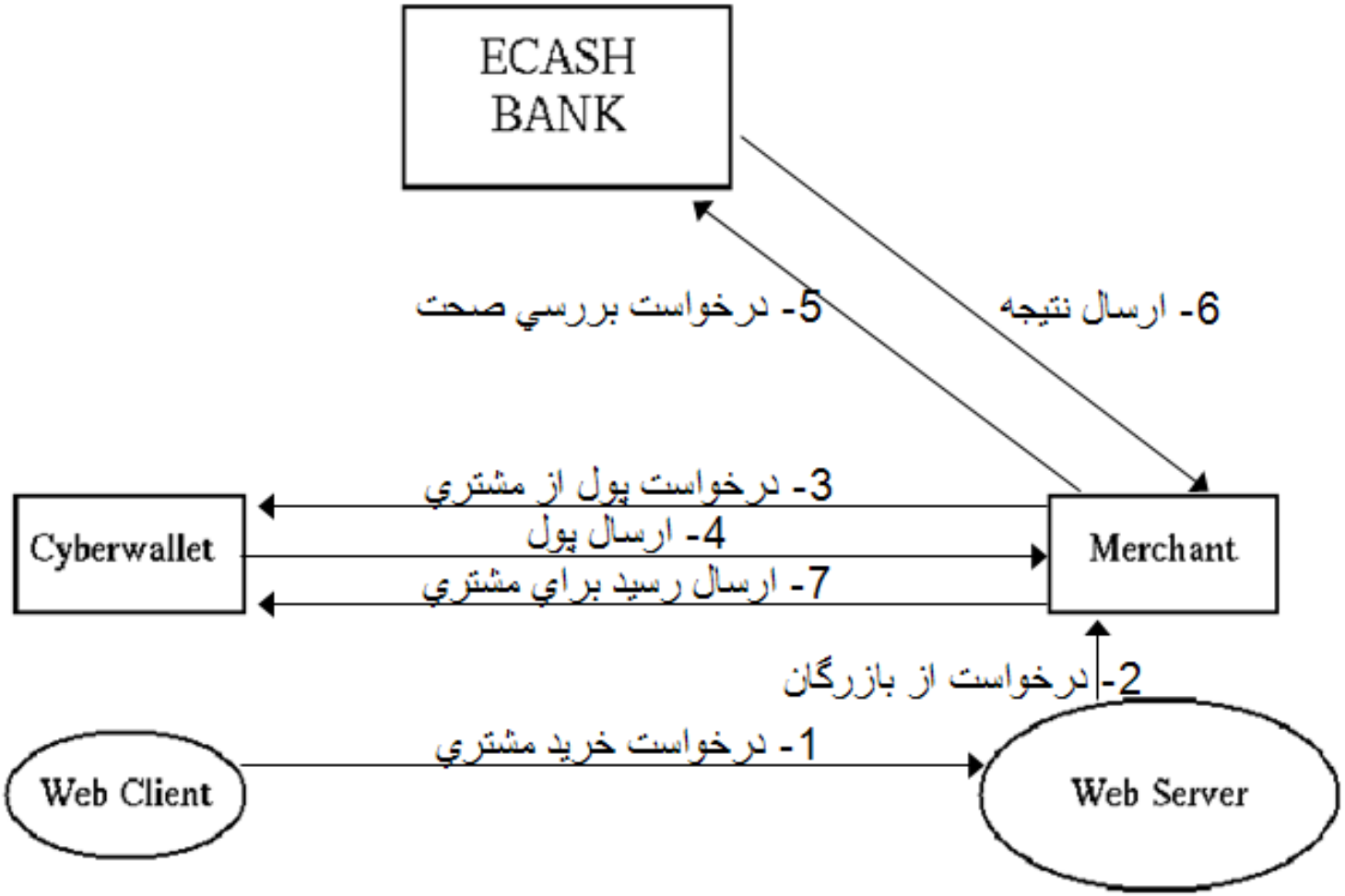
معرفي پروتكل e-Cash- مشتري چگونه پول را خرج مي كند؟

1- براي پرداخت 2 دلار به باب، آليس به او (x, C) را مي دهد.

2- باب براي اينكه مطمئن شود كه پول توسط بانك صادر شده است، C را به توان 5 مي رساند و آنرا با f(x) مقايسه مي كند. . حال باب براي اينكه مطمئن شود كه اين پول قبلا خرج نشده است (x, C) را به بانك مي فرستد.

3- بانك x را چك مي كندكه آيا قبلا استفاده شده است يا خير.در صورت استفاده نشدن به باب اطلاع داده و 2 دلار به حساب وي واريز مي كند. در واقع بانك ليست كليه x ها را نگه مي دارد.

معرفي پروتكل e-Cash- مراحل انجام معامله



شکل 17: گامهای پروتکل e-cash

خلاصه و نتیجه گیری

- مفهوم رمزنگاری
- رمزنگاری و رمزگشایی
- رمزنگاری متقارن
- هش کردن یا خلاصه سازی پیغام
- رمزنگاری کلید عمومی یا نامتقارن
- امضای دیجیتال و پوشش

خلاصه و نتیجه گیری

- پروتکلهای iKP
- پروتکل e-Cash
- تراکنش الکترونیکی ایمن
- پروتکل 3D-Secure